Proxmox

Engineering Handbook

Contents

Overview

Proxmox VE is an open-source virtualization platform that integrates KVM hypervisor and LXC containers under a unified, web-based management interface. It supports clustering, high availability, and software-defined storage to streamline datacenter operations.

Architecture

Proxmox VE's core components:

- Proxmox VE Host
- QEMU/KVM for full virtualization
- LXC for lightweight containers
- Corosync for cluster messaging
- pmxcfs – a database-driven cluster filesystem that stores all VM/CT configs in RAM and on disk, replicated in real time across nodes
- Storage backends: ZFS, Ceph, NFS, iSCSI

Installation

Hardware Requirements

- CPU: 64-bit, hardware virtualization support (Intel VT-x or AMD-V)

- Memory: ≥ 8 GB RAM (16 GB+ for Ceph or large clusters)

- Storage: SSD for OS; additional disks for VMs/containers

- Network: Dual NICs recommended for redundancy

Infrastructure Planning

Plan the environment around workload, hardware compatibility, and future growth.

- Right-size CPU, memory, and storage based on expected VM/container profiles.
- Choose enterprise-grade servers with ECC RAM and official Proxmox compatibility.
- Allocate separate NICs for management, VM traffic, and storage replication to avoid contention.

Installation Steps

1. Download the latest Proxmox VE ISO.

2. Boot target server from USB or virtual media.

3. Follow installer prompts to configure root password, management interface, and storage.

4. Reboot into the Proxmox VE environment and access the web GUI at https://your-host:8006.

Verify Hardware & Prepare Disk

```
# Check VT-x or AMD-V support
        egrep -c "(vmx|svm)" /proc/cpuinfo

# Example:
        partition disk /dev/sda
        parted /dev/sda --script mklabel gpt \
          mkpart primary 1MiB 512MiB name 1 boot \
          mkpart primary 512MiB 100% name 2 data
```

```
# Format and mount
    mkfs.ext4 /dev/sda1
    mount /dev/sda1 /mnt

# Set hostname
    echo proxmox-node1 > /mnt/etc/hostname
```

Add Proxmox Repository & Install

```
# Add enterprise repo file
    cat <<EOF > /etc/apt/sources.list.d/pve-enterprise.list
    deb https://enterprise.proxmox.com/debian/pve bookworm pve-enterprise
    EOF

# Add Proxmox key
    wget -qO - http://download.proxmox.com/debian/proxmox-ve-release-7.x.gpg | apt-key
    add -

# Update and install Proxmox VE metapackage
    apt update && apt full-upgrade -y
    apt install proxmox-ve postfix open-iscsi -y
```

Post-Install Access

```
# Reboot into Proxmox VE
    reboot

# On another machine, access web GUI
# https://proxmox-node1:8006
```

Regular Maintenance

Keep the environment resilient and up to date through disciplined housekeeping.

- Schedule quarterly firmware and BIOS upgrades on all nodes.
- Review Proxmox and Linux security advisories monthly and apply patches promptly.
- Audit user accounts, SSH keys, and role assignments to detect orphaned credentials.
- Clean up unused templates, snapshots, and stale backups to reclaim storage.

Central Management

All tasks can be performed from any cluster node via:

- Web GUI (ExtJS-based, multi-master, no external DB needed)
- CLI tools (`pvecm`, `qm`, `pct`, `vzdump`, `ha-manager`) with tab completion and man pages
- REST API (JSON-Schema defined) for seamless third-party and IaC integration

Networking

Bridge-based networking is native in Proxmox:

- Create a Linux bridge (e.g., vmbr0) mapped to a physical NIC.

- Attach VMs and containers to vmbr0 for external connectivity.

- VLAN tagging: define VLAN interfaces on the bridge (e.g., vmbr0.100).

- Bonding: aggregate multiple NICs for throughput and failover.

Best practices:

- Define distinct Linux bridges for management, VM, and storage traffic.
- Implement VLAN tagging on both switches and hosts to separate tenant or service networks.
- Configure NIC bonding (LACP or active-backup) to increase throughput and provide failover.
- Enable jumbo frames (MTU 9000) on storage and VM networks if the infrastructure supports it.

Bridge & VLAN Configuration

Edit /etc/network/interfaces:

```
auto vmbr0
iface vmbr0 inet static
  address 10.0.1.10/24
  gateway 10.0.1.1
  bridge-ports enp3s0
  bridge-stp off
  bridge-fd 0

auto vmbr0.100
iface vmbr0.100 inet manual
  vlan-raw-device enp3s0
```

Apply any changes:

```
ifreload -a
ip link show vmbr0.100
```

Add an additional NIC with VLAN 100

```
# Add a second NIC with VLAN 100
        qm set 200 \
         --net1 virtio,bridge=vmbr0.100

# Show current network config
        qm config 200 | grep net
```

Network Interface Card (NIC) Bonding

```
auto bond0
iface bond0 inet manual
bond-slaves enp3s0 enp4s0
  bond-miimon 100
  bond-mode active-backup

auto vmbr1
iface vmbr1 inet static
  address 192.168.100.10/24
  bridge-ports bond0
```

activate:

```
ifup bond0 vmbr1
```

Show bridges and ports

```
brctl show
ip link show
```

Check connectivity inside cluster

```
ping -I vmbr0.10 10.0.10.3
```

Capture VLAN traffic

```
tcpdump -i enp3s0 -e vlan
```

Storage

Supported backends:

- Local LVM/ZFS: for standalone hosts
- NFS/CIFS: shared filesystems
- Ceph RBD: distributed, resilient block storage
- iSCSI: SAN integration

Best practices:

Select and configure storage backends to match performance and resiliency goals.

- For local hosts, use ZFS with compression (lz4) and regular scrubbing for data integrity.
- For shared storage, deploy a Ceph cluster with at least three nodes and dedicate a private cluster network.
- Use thin provisioning (LVM-thin or Ceph RBD) to optimize disk utilization and simplify snapshots.
- Schedule automated *zfs scrub* and *ceph health* checks during off-peak windows.

Adding Storage

GUI Path:

Datacenter → Storage → Add → Directory

1. Directory Storage

Use a filesystem directory for ISOs, templates, backups, and container rootdirs.

```
# Create mountpoint
    mkdir -p /mnt/data

# Mount disk (or NFS share) in /etc/fstab
    /dev/sdf1  /mnt/data  ext4  defaults  0 2

# Register as Proxmox storage
pvesm add dir local-data \
     --path /mnt/data \
     --content iso,backup,vztmpl,rootdir

# Verify storage
    pvesm status

Name             Type     Status           Total            Used       Available        %
ProxMox          cifs     inactive             0               0               0    0.00%
QNAP             cifs     inactive             0               0               0    0.00%
QNAP-ISO         cifs     active        10403071552      4144345636      6258725916   39.84%
VM-Directory     dir      active          491134172         2813380       463299080    0.57%
VMs              lvmthin  active         3873329152       121622535      3751706616    3.14%
```

```
VMs-ADD          lvmthin  active    365760512          0    365760512  0.00%
VMs-Add          lvm      active   3907014656          0   3907014656  0.00%
local            dir      active     98497780   36016076     57432156 36.57%
```

pvesm list local

```
Volid                                                Format  Type         Size VMID
local:backup/vzdump-qemu-100-2024_11_06-21_00_04.vma.zst vma.zst backup
27007593091 100
local:backup/vzdump-qemu-101-2024_10_27-21_04_34.vma.zst vma.zst backup
1595717382 101
local:backup/vzdump-qemu-102-2024_11_05-21_05_01.vma.zst vma.zst backup
512734632 102
local:backup/vzdump-qemu-103-2024_11_05-21_05_14.vma.zst vma.zst backup
558619883 103
local:backup/vzdump-qemu-104-2024_11_06-21_05_33.vma.zst vma.zst backup
480915474 104
local:iso/X860923OE.ISO                              iso     iso      1584857088
local:vztmpl/ubuntu-23.10-standard_23.10-1_amd64.tar.zst tzst    vztmpl     137846089
```

ZFS Management and Scrubbing

```
# Create dataset with quotas and dedupe
    zfs create rpool/data/vmstore \
     -o quota=500G \
     -o dedup=on \
     -o compression=lz4 \
     -o recordsize=16K

# Schedule weekly scrub via cron
    echo "0 3 * * 0 root zpool scrub rpool" > /etc/cron.d/zfs-scrub

# Monitor scrub progress
    zpool status rpool | grep scrub
```

ZFS Create a Mirrored Pool

GUI Path:

Datacenter → Storage → Add → ZFS

```
# Create a mirrored pool
    zpool create rpool mirror /dev/sdb /dev/sdc \
       -o ashift=12 -O atime=off -O compression=lz4

# Verify
    zpool status rpool
    zfs set mountpoint=/rpool rpool


# Create dataset for VM disks
    zfs create zfspool/vmdata

# Add to Proxmox
```

```
pvesm add zfs data-zfs \
 --pool zfspool \
 --content images,rootdir
```

LVM Thin Pool

GUI Path:

Datacenter → Storage → Add → LVM
Datacenter → Storage → Add → LVM-Thin

```
# Create thinpool
    lvcreate --type thin-pool \
     --poolmetadata vg-data/tmeta \
     --size 1T \
     --name thinpool vg-data

# Register with Proxmox
    pvesm add lvmthin data-thin \
     --vgname vg-data \
     --thinpool vg-data/thinpool \
     --content images,rootdir
```

LVM Thin Snapshots and Clones

```
# Create a thin snapshot of LV 100
    lvcreate --snapshot --name vm100-snap --size 10G /dev/pve-vg/vm-100-disk-0

# Clone into a new LV
    lvcreate --name vm100-clone --virtualprivate pve-vg/vm-100-disk-0-snap

# Remove snapshot when done
    lvremove /dev/pve-vg/vm100-snap
```

Ceph Cluster Setup

GUI Path:

Datacenter → Storage → Add → RBD

```
# Install Ceph on each node
    apt install ceph ceph-common -y

# On node1: create monitor and manager
    pveceph init --cluster-network 10.0.2.0/24

# Create OSD on /dev/sde
    pveceph osd create /dev/sde

# Mark OSD out for maintenance
    ceph osd out 2

# Create RBD pool
    ceph osd pool create vmdata 128
```

```
# Assuming Ceph monitors configured via pveceph init, see above
# Register RBD pool
        pvesm add rbd ceph-vmdata \
         --monhost 10.0.2.11,10.0.2.12 \
         --pool vmdata \
         --username admin \
         --content images \
         --secretring /etc/ceph/ceph.client.admin.keyring

# Verify
        rbd ls -p vmdata
        pvesm status
```

NFS Share

Mount a remote filesystem for ISOs and backups.

GUI Path:

Datacenter → Storage → Add → NFS

```
# Install NFS client
        apt install nfs-common -y

# Add to Proxmox
        pvesm add nfs nfs-backup \
         --server 10.0.2.20 \
         --export /export/backups \
         --path /mnt/nfs-backup \
         --content backup,iso
```

iSCSI and LVM over iSCS

Connect SAN LUNs as logical volumes.

GUI Path:

Datacenter → Storage → Add → LVM

```
# Discover target
        iscsiadm -m discovery -t st -p 10.0.2.30

# Login
        iscsiadm -m node --login

# Create VG on LUN
        pvcreate /dev/sdj
        vgcreate vg-iscsi /dev/sdj

# Add to Proxmox
        pvesm add lvm iscsi-lvm \
         --vgname vg-iscsi \
         --content images
```

Clustering

1. Initialize the first node: `pvecm create CLUSTERNAME`.
2. On each additional node: `pvecm add <IP-of-master>`.
3. Corosync handles messaging and quorum.
4. Configure fencing (e.g., IPMI) to avoid split-brain.

Initialize & Join Nodes

```
# On first node
    pvecm create mycluster --force 1

# On additional node
    pvecm add 10.0.1.10

# Check quorum
    pvecm nodes
    pvecm status
```

Corosync Debug

```
# View real-time cluster map
    corosync-cmapctl | head

# Tail logs
    journalctl -u corosync -f
```

Virtualization

KVM Virtual Machine Lifecycle

GUI Path:

Datacenter → Node → Create VM

- Create with `qm create`, or via GUI.

- Use cloud-init templates for rapid VM provisioning.

- PCI passthrough: configure IOMMU, add host PCI device.

```
# Create VM 100 with 2 CPUs, 4G RAM, 20G disk
    qm create 100 \
     --name ubuntu2004 \
     --memory 4096 \
     --cores 2 \
     --net0 virtio,bridge=vmbr0 \
     --scsihw virtio-scsi-pci \
     --ide2 local:iso/ubuntu-20.04.iso,media=cdrom

# Adding and resizing disks

# Attach a 40 GB disk on LVM-Thin
    qm set 200 --scsi0 data-thin:40

# Attach a 10 GB Ceph RBD
    qm set 200 --scsi1 ceph-vmdata:10

# Resize scsi0 by +20 GB
    qm resize 200 scsi0 +20G

# Start and console
    qm start 100 && qm monitor 100

# Snapshot VM
    qm snapshot 100 snap-preupdate
```

LXC Container Management

- Create with `pct create` using download templates.

- Leverage resource quotas: CPU shares, memory limits, disk quotas.

- Network: assign to bridges or use veth interfaces.

```
# Download Debian template
    pct create 200 local:vztmpl/debian-12-standard_12.0-1_amd64.tar.gz \
     --hostname ct-deb12 \
     --storage local-lvm \
     --rootfs local-lvm:4 \
```

```
    --memory 1024 \
    --swap 512 \
    --net0 name=eth0,bridge=vmbr0,ip=10.0.1.200/24

# Start and enter
    pct start 200
    pct enter 200

# Limit CPU and memory dynamically
    pct set 200 --cpulimit 2 --memory 2048
```

Start, Stop and Reboot a VM

```
# Start, stop, and reboot
    qm start 200
    qm shutdown 200
    qm reboot 200

# Live migrate to another node
    qm migrate 200 proxmox-node2 --online
```

List VM's

```
# List Virtual machines
    qm list
```

List Volumes

```
# List Volumes
    pvesmpv status
```

Backup and Restore

Implement consistent, automated backups and validate restores regularly.

- Use vzdump in snapshot mode with compression, storing archives on remote NFS or CIFS shares.
- Define retention policies (daily/weekly/monthly) and rotate backups to off-site locations if possible.
- Automate backup schedules via the Proxmox GUI or cron jobs.
- Perform periodic test restores on a non-production host to verify backup integrity.

```
# Live backup all VMs daily
        vzdump --all --mode snapshot \
          --storage backup-nfs \
          --compress lzo \
          --mailto admin@example.com \
          --mailto-retry 3

# Restore VM 300 from backup file
        vzdump --restore /backup/dump/vzdump-qemu-100-2025_09_28-02_00_00.vma.lzo 300
```

Proxmox Backup Server Integration

```
# Register PBS storage
        pvesm add pbs pbs-storage --server pbs.local --username backup@pbs \
          --password 'P@ssw0rd' --fingerprint ABCD1234

# Backup job via CLI
        pbs backup --repository pbs-storage vm/901 --mode snapshot

# Restore to new VM ID
        pbs restore pbs-storage:2025-09-30:vm/901 vm/902
```

Snapshots and Rollback

```
# Create a snapshot
        qm snapshot 200 before-patch

# List snapshots
        qm listsnapshot 200

# Roll back to snapshot
        qm rollback 200 before-patch
```

High Availability

- Enable HA manager in the cluster configuration.

- Define HA groups for critical VMs/containers.

- Ensure fencing is operational to trigger automatic failover.

- Regularly simulate failures to validate HA behavior.

Best Practice:

Ensure critical workloads stay online through proper cluster configuration.

- Build clusters with an odd number of nodes (minimum three) to maintain quorum.
- Configure fencing devices (IPMI, DRAC, iLO) to handle split-brain scenarios.
- Place critical VMs and containers into HA groups and set appropriate restart priorities.
- Monitor quorum status with pvecm status and test failovers annually.

```
# Enable HA on VM 100
    ha-manager add vm:100

# Check HA status
    ha-manager status

# Create HA group
    ha-manager group add critical-group 100 101

# Remove from HA
    ha-manager remove vm:100
```

Disaster Recovery Checklist

- Verify cluster health: pvecm status
- Confirm storage access on DR site
- Restore latest backup:
    vzdump --restore /mnt/dr/vzdump-qemu-901-latest.vma.lzo 950
- Validate network configs, cloud-init, and DNS entries.

Monitoring and Alerting

Maintain visibility into host health, resource usage, and service availability.

- Deploy Prometheus with the Proxmox exporter to collect metrics on CPU, memory, disk, and network.
- Create Grafana dashboards to visualize cluster trends, storage latency, and HA failovers.
- Define alert rules for node unreachability, Ceph health degradations, and backup failures.
- Integrate notifications via email, Slack, or PagerDuty for rapid incident response.

```
# Install Prometheus exporter
        apt install prometheus-node-exporter pve-exporter -y

# prometheus.yml snippet
        - job_name: 'proxmox'
          static_configs:
            - targets: ['10.0.1.10:9221','10.0.1.11:9221']

# Alertrule example
        groups:
        - name: ProxmoxAlerts
          rules:
          - alert: ProxmoxNodeDown
            expr: up{job="proxmox"} == 0
            for: 5m
            labels:
              severity: critical
            annotations:
              summary: "Node {{ $labels.instance }} is down"
```

Security

Best Practice:

Lock down access and enforce least-privilege controls to reduce attack surface.

- Disable direct root login over SSH; require key-based authentication and non-default ports.
- Enable two-factor authentication for the Proxmox web interface.
- Define granular roles and permissions via RBAC; avoid granting blanket privileges.
- Keep the OS and Proxmox packages up to date with apt update && apt dist-upgrade.
- Configure a host-based firewall (e.g., ufw or iptables) to restrict management ports.

```
# Harden SSH
        sed -i 's/PermitRootLogin yes/PermitRootLogin no/' /etc/ssh/sshd_config
        systemctl restart sshd

# Add 2FA for web GUI
        pveam update
        pveum user modify root@pam --otp google-authenticator

# Apply security patches
        apt update && apt upgrade --with-new-pkgs -y
```

Performance Tuning

Tune host and VM settings to maximize resource efficiency and predictability.

- Set VM CPU type to host for access to all instruction sets and best performance.
- Allocate hugepages for database-heavy workloads to reduce TLB misses.
- Adjust ZFS recordsize per workload (e.g., 16K for databases, 128K for file servers).
- Pin latency-sensitive VMs to specific NUMA nodes and cores when necessary.
- Use VirtIO drivers in guests for network and storage to minimize overhead.

```
# Enable HugePages
        echo "vm.nr_hugepages=1024" >> /etc/sysctl.conf && sysctl -p

# Tune ZFS recordsize for random I/O
        zfs set recordsize=16K rpool/data

# Pin VM 100 CPUs to host cores 0–1
        qm set 100 --cpuunits 2000 --socket 1 --cores 2 --cpulimit 2
```

Troubleshooting

Common issues and diagnostics:

- Networking: check `ip a`, `brctl show`.
- Cluster: `pvecm status`, `corosync-cmapctl`.
- Ceph: `ceph health`, `ceph osd tree`.
- Logs: `journalctl -u pveproxy`, `/var/log/pve/tasks/`.

```
# Proxmox version and patch level
    pveversion -v

# Cluster quorum issues
    pvecm expected_votes
    pvecm nodes

# ZFS integrity
    zpool status -v

# Ceph health check
    ceph health detail

# GUI/API proxy logs
    journalctl -u pveproxy -f

# Backup failures
    grep ERROR /var/log/vzdump/*
```

Appendix A - Useful Commands

- Cluster: `pvecm status`
- VMs: `qm list`, `qm migrate`, `qm monitor`
- Containers: `pct list`, `pct enter`, `pct migrate`
- Backups: `vzdump`, `pve-backup-client`

Appendix B – Configuring a Solaris VM

This appendix covers the end-to-end process for deploying a Solaris 11 VM on Proxmox VE, from ISO prep through post-install tuning.

1. Upload and Register the Solaris ISO

a) Secure-copy the ISO into Proxmox's ISO storage:

```
scp Solaris-11.4-Live-Server.iso root@proxmox-node:/var/lib/vz/template/iso/
```

b) Verify that it is registered:

```
pvesm list iso
```

2. Create the Solaris VM

```
# VM ID 300, 4 GiB RAM, 2 cores, e1000 NIC, IDE boot
qm create 300 \
 --name solaris11 \
 --memory 4096 \
 --cores 2 \
 --cpu host \
 --net0 e1000,bridge=vmbr0 \
 --ide2 local:iso/Solaris-11.4-Live-Server.iso,media=cdrom \
 --boot order=ide2 \
 --serial0 socket \
 --vga serial0
```

- net0 e1000 ensures compatibility with Solaris's built-in NIC driver.
- serial0 socket + --vga serial0 enable text-mode console access.

3. Start and Install

```
qm start 300
# Open serial console from CLI or web GUI
qm terminal 300
```

- Follow the live-server installer prompts over the serial console.
- Partition with Solarish defaults (ZFS root recommended).

4. Post Installation Configuration

```
# Enable SSH on the first boot if not already
    svcadm enable ssh

# Sync time with the host
    pkg install system/ntp
```

```
svcadm enable ntp

# (Optional) Convert disk bus to VirtIO if Solaris drivers installed
qm shutdown 300
qm set 300 --ide2 none
qm set 300 --scsi0 local-lvm:32 \
 --scsihw virtio-scsi-pci
qm start 300
```

5. Ongoing Maintenance Tips

- Use ZFS on Solaris as both root and data pool for snapshots and checksums.
- If VirtIO drivers were added, monitor /var/adm/messages for disk I/O errors.
- To take online ZFS snapshots from the Proxmox host:

```
ssh root@solaris11 "zfs snapshot rpool/ROOT/solaris11@pve-snap-$(date +%F)"
```

- Keep the Solaris ISO and VM templates up to date for patch-level consistency.

Appendix C – Configuring a OpenVMS VM

This appendix covers the end-to-end process for installing and tuning OpenVMS 8.4 on Proxmox VE—covering ISO upload, VM creation, installation via VNC, networking setup, and maintenance tips.

1. Upload the OpenVMS ISO

```
scp OVMSE84_CD.iso root@proxmox-node:/var/lib/vz/template/iso/
ls /var/lib/vz/template/iso | grep OVMSE84_CD.iso
```

2. Create the OpenVMS VM

```
qm create 400 \
 --name openvms84 \
 --memory 4096 \
 --cores 2 \
 --cpu host \
 --net0 e1000,bridge=vmbr0 \
 --ide2 local:iso/OVMSE84_CD.iso,media=cdrom \
 --scsi0 local-lvm:32 \
 --scsihw virtio-scsi-pci \
 --boot order=ide2 \
 --vga std \
 --serial0 socket
```

- Uses e1000 NIC for native OpenVMS driver support.
- virtio-scsi-pci for performant disk I/O (ensure OpenVMS SCSI drivers loaded).
- --vga std + VNC console for graphical installer.

3. Start and Install

```
        qm start 400
# Note the VNC port printed; or run:
        qm vncproxy 400
```

4. Post Installation Configuration

After boot:

```
# In the console, run
        SET PROC/PRCL_DATETIME
        TCPIP$CONFIGURATION
```

a) Configure TCP/IP:

- Define ADDRESS, SUBNET, GATEWAY under TCPIP$CONFIGURATION.
- Enable IFCONFIG STARTUP.

b) For DECnet: edit SYS$MANAGER:MODPARAMS.DAT and add

NETX$NODE_TYPE = END_SYSTEM
NETX$ADDRESS = 10.0.4.40
NETX$GATEWAY_ADDR = 10.0.4.1

c) Reboot: `SHUTDOWN REBOOT`

5. Ongoing Maintenance Tips

- Host-side ZFS snapshots:
  zfs snapshot zfspool/vmdata/openvms84@pve-$(date +%F)

- Monitor OpenVMS console logs:
  qm monitor 400

- Regularly back up using OpenVMS BACKUP utility to a shared drive mounted via CIFS/NFS on the guest.
- If you install VirtIO SCSI drivers on OpenVMS, watch /var/adm/messages (guest) for I/O errors.

References

- Official docs: https://pve.proxmox.com/wiki/
- Community forum: https://forum.proxmox.com/
- Ceph integration guide