Proxmox VE

- ZFS over ISCSI
- Quorum with an External QDevice
- Debian Cloud-Init
    - Proxmox · Debian-VM-Strategie (Cloud-Init)
- Knowledge Base
    - NTP auf dem Proxmox-Cluster
    - UEFI Secure Boot Zertifikate 2023 (Proxmox EFI-Disk)

ZFS over ISCSI

Quorum with an External QDevice

Having only two nodes in my Proxmox cluster, I wanted to add a third external device to keep quorum during reboots or other outages. To do so I added an external qdevice as a node which works as a voting only member of the cluster. The qdevice has to be debian based, so I set mine up on a raspberry pi.

First you'll need to install the following packages on your external server.

```
sudo apt install corosync-qnetd corosync-qdevice
```

Once the install is complete, use this command to install a package on each of your existing Proxmox nodes.

```
apt install corosync-qdevice
```

Now you can run the following on one of your Proxmox nodes.

```
pvecm qdevice setup <QDEVICE-IP>
```

You will need to have a root password available for the setup. In my case I temporarily allowed root login with password and then reverted it back to `prohibit-password` after. That configuration is found in:

```
sudo nano /etc/ssh/sshd_config
```

Once you've run the setup command, you'll be able to run the following to view the status and nodes.

```
pvecm status
pvecm nodes
```

For additional information, check out the documentation here.

Debian Cloud-Init

Proxmox · Debian-VM-Strategie (Cloud-Init)

Proxmox · Debian-VM-Strategie (Cloud-Init)

> **abstract:** Zweck Debian-13-VMs auf Proxmox reproduzierbar aus einem Cloud-Init-Template bereitstellen. Ausführbarer Source of Truth ist das zentrale interne Git-Repo; dieses Dokument beschreibt den Standard und die Betriebsregeln.

Gültig für: Debian 13 (trixie), Proxmox VE, UEFI/OVMF, qcow2-Images

1 · Standard und Entscheidung

Modus	Image	Einsatz
`cloudinit`	`debian-13-generic-amd64.qcow2`	Standard; User, Key, Hostname und IP pro VM per Cloud-Init
`bake`	`debian-13-nocloud-amd64.qcow2`	Sonderfall für bewusst fest gebackene Images

> **tip:** Empfehlung Für neue Staging- und Infrastruktur-VMs immer `cloudinit` mit `--img genericcloud` verwenden.

> **warning:** Image-Falle `nocloud` enthält kein Cloud-Init. Der Skriptparameter `--img genericcloud` verwendet die Datei `debian-13-generic-amd64.qcow2`.

2 · Skripte installieren

Auf jedem neuen Proxmox-Host als `root`:

```
curl -fsSL https://git.corestream.ch/corestream/proxmox-debian-cloudinit-scripts/~raw/master/install-on-proxmox.sh | bash
```

- Installationspfad: `/opt/corestream/proxmox-debian-cloudinit/` (`build-debian-template.sh`, `clone-vm.sh`)
- Kommandos über Symlinks: `/usr/local/sbin/build-debian-template`, `/usr/local/sbin/clone-vm`
- Der OneDev-Raw-Zugriff ist anonym als `Code Reader` freigegeben; der Installer braucht kein Token.

3 · Voraussetzungen

3.1 Pakete auf Proxmox

```
apt update
apt install -y libguestfs-tools wget curl
```

`qemu-guest-agent` wird nicht für den Proxmox-Host benötigt — das Build-Skript installiert und verifiziert es im Debian-Image.

3.2 Storage und Debian-Image

```
pvesm status --enabled
mkdir -p /mnt/pve/<STORE>/import
cd /mnt/pve/<STORE>/import
wget -O debian-13-generic-amd64.qcow2 https://cloud.debian.org/images/cloud/trixie/latest/debian-13-generic-amd64.qcow2
wget -O SHA512SUMS https://cloud.debian.org/images/cloud/trixie/latest/SHA512SUMS
sha512sum -c --ignore-missing SHA512SUMS
```

Rollen im Build-Skript:

- **Quellstorage:** NFS mit unverändertem qcow2 unter `/mnt/pve/<SOURCE_STORE>/import/`; bei mehreren NFS-Storages erscheint ein Auswahlmenü, alternativ `--source-store <STORE>`.
- **Zielstorage:** lokaler, aktivierter Proxmox-Storage mit Content-Typ `images`; Auswahlmenü bzw. `--target-store <STORE>`.

VM-Disk, EFI-Disk und Cloud-Init-Disk liegen immer auf dem Zielstorage; NFS wird nie als VM-Ziel angeboten.

3.3 Admin-Public-Key

Standardpfad prüfen:

```
test -s /root/.ssh/id_csgadmin.pub
```

Fehlt die Datei, explizit einen vorhandenen Public-Key übergeben:

```
build-debian-template -t 9000 -n debian13-init -m cloudinit --img genericcloud --user-key
/pfad/admin.pub
```

> **warning:** > `--user-key /root/.ssh/authorized_keys` übernimmt **alle** dort enthaltenen Keys. Für genau einen Admin-Key eine dedizierte Public-Key-Datei verwenden.

4 · Template bauen

```
build-debian-template -t 9000 -n debian13-init -m cloudinit --img genericcloud
## optional: --user-key /pfad/admin.pub
## optional (opt-in): --hermes --hermes-key /pfad/hermes.pub
```

Das Skript:

- kopiert das qcow2 pro Build nach `/var/tmp` und verändert **niemals** das zentrale NFS-Basisimage
- installiert und verifiziert `qemu-guest-agent`, `sudo` und `openssh-server` in der Arbeitskopie
- aktiviert QEMU Guest Agent und serielle Konsole
- deaktiviert Root-SSH und Passwort-SSH
- erzwingt die NoCloud-Datasource
- deaktiviert IPv6 per Sysctl und Kernelparameter `ipv6.disable=1`
- importiert VM-, EFI- und Cloud-Init-Disk auf den lokalen Zielstorage
- erstellt eine OVMF/Q35-VM mit SCSI-Disk und Cloud-Init-Drive
- setzt `ciuser=csgadmin`, SSH-Key und DHCP als Ausgangswert
- wandelt die VM in ein Template um

Prüfen:

```
qm config 9000 | grep -E 'agent|scsi1|cloudinit|ciuser|boot'
```

Erwartet: `agent: 1` (bzw. `enabled=1`), `scsi1` enthält `cloudinit`, `ciuser: csgadmin`, Boot-Reihenfolge enthält `scsi0;scsi1`.

> **important:** > Template **niemals booten** — immer einen Clone zum Testen verwenden. Ein gebootetes Template hat Cloud-Init bereits ausgeführt.

5 · VM klonen

```
## DHCP, gewünschte Endgrösse
clone-vm 9000 901 test-debian dhcp --disk-size 32G

## statische IPv4, 64-GB-Disk (Werte im Zielnetz geltend setzen)
clone-vm 9000 121 app-staging <IPv4>/<CIDR> <Gateway-IPv4> --disk-size 64G

## zusätzliche NIC (wiederholbar)
clone-vm 9000 130 dualnic dhcp --add-nic vmbr1 dhcp
```

- `--disk-size` ist optional; vergrössert `scsi0` **vor** dem ersten Boot auf die absolute Grösse. Verkleinern nicht möglich; mindestens Template-Grösse. Debian Cloud-Init/`growpart` erweitert Partition und Dateisystem automatisch.
- `--add-nic` erzeugt `net1`/`ipconfig1`, dann `net2`/`ipconfig2` usw. Sekundäre NICs erhalten bewusst kein zweites Default-Gateway.
- Der Clone-Helper erstellt Full Clones, setzt `agent enabled=1`, die IP-Konfiguration, regeneriert den Cloud-Init-Seed nach allen Änderungen und startet die VM.
- Linked Clones nur für Throwaway-Tests; Staging-/Prod-/Infra-VMs bleiben Full Clones.

Netzwerkdienst im Debian-Gast

Das Debian-13-`generic-amd64`-Image (laut offiziellem Paketmanifest, Stand 2026-10) enthält **Netplan** (`netplan.io`) und `systemd-resolved`; NetworkManager und klassisches ifupdown sind nicht enthalten. Netplan rendert per Standard-Renderer `systemd-networkd` (Teil von systemd).

```
systemctl is-active systemd-networkd
systemctl is-active systemd-resolved
networkctl
ls -l /etc/netplan/
```

Bestehende VM manuell um eine NIC erweitern

Auf Proxmox (VM sauber herunterfahren):

```
qm shutdown <VMID>
qm set <VMID> --net1 virtio,bridge=vmbr1
qm start <VMID>
```

Im Gast Interfacename ermitteln (`ip -br link`; typisch `net0` → `ens18`, `net1` → `ens19` — immer tatsächlich prüfen). Netplan-Datei ergänzen:

```
## DHCP
printf '%s\n' 'network:' '  version: 2' '  ethernets:' '    ens19:' '      dhcp4: true'
'      dhcp6: false' \
  | sudo tee /etc/netplan/60-secondary-nic.yaml >/dev/null
## statisch, ohne zweites Default-Gateway
printf '%s\n' 'network:' '  version: 2' '  ethernets:' '    ens19:' '      dhcp4: false'
'      dhcp6: false' '      addresses: [<IPv4>/<CIDR>]' \
  | sudo tee /etc/netplan/60-secondary-nic.yaml >/dev/null

sudo netplan generate && sudo netplan apply
```

Prüfen: `ip -br addr`, `networkctl status ens19`.

> **important:** > Bei einer bereits initialisierten VM im Gast über Netplan ändern. Nicht darauf vertrauen, dass nachträglich gesetztes Proxmox-`ipconfig1` Cloud-Init vollständig erneut ausführt.

6 · Verifikation

6.1 Im Gast

```
cloud-init status --long
id csgadmin && hostname && ip -4 a
ls -l /dev/disk/by-label/cidata
dpkg-query -W qemu-guest-agent && systemctl is-active qemu-guest-agent
sysctl net.ipv6.conf.all.disable_ipv6 && cat /proc/cmdline
```

Erwartet: Cloud-Init `status: done`, `csgadmin` existiert, Hostname/IPv4 stimmen, `cidata` vorhanden, Guest-Agent aktiv, `disable_ipv6 = 1`, `ipv6.disable=1` in der Kernel-Commandline, keine IPv6-Adressen.

6.2 Auf Proxmox

```
qm agent <VMID> ping
ssh csgadmin@<IPv4>
```

7 · User und SSH-Zugriff

User	Standard	Key	Rechte
`csgadmin`	immer im Cloud-Init-Modus	`--user-key`, Standard `/root/.ssh/id_csgadmin.pub`	Admin-Zugang gemäss Cloud-Image/Cloud-Init
`hermes`	nur mit `--hermes`	`--hermes-key`, Standard `/root/.ssh/id_ed25519.pub`	`NOPASSWD:ALL`, Passwort gesperrt
`root`	kein SSH-Login	keiner	`PermitRootLogin no`

Zusätzlich gilt `PasswordAuthentication no`; SSH-Zugriff nur per Key. Hermes ist opt-in — der User wird nie automatisch angelegt.

8 · Goldene Regeln

1. Für Cloud-Init `debian-13-generic-amd64.qcow2` verwenden.
2. Template nie booten.
3. Cloud-Init-Drive muss vorhanden sein; bei OVMF auf `scsi1`.
4. Nach Änderungen an `ci*` immer `qm cloudinit update <VMID>`.
5. Staging-/Prod-/Infra-VMs als Full Clones erstellen.
6. Admin-Key vor dem Build verifizieren; der Build bricht bei fehlender/leerer Key-Datei ab.
7. Nach dem ersten Clone Cloud-Init, Guest Agent, IPv4, SSH und deaktiviertes IPv6 prüfen.
8. Skripte nicht aus der KB kopieren; immer den aktuellen Stand aus dem Repo installieren.

9 · Troubleshooting

Symptom	Ursache	Fix
`user pubkey missing`	Default-Key-Datei fehlt	dedizierte Datei oder `--user-key <datei>`
`storage ... does not support vm images`	Quell-NFS als VM-Ziel oder alter Skriptstand	Installer erneut ausführen; lokalen Zielstorage wählen bzw. `--target-store`
VM-ID existiert nach fehlgeschlagenem Import	Teilbau vor Storage-Fehler angelegt	`qm config <VMID>` prüfen, Teilbau löschen oder neue ID
Basis-qcow2 stimmt nicht mit SHA512	alter Skriptstand hat NFS-Image direkt verändert	Original neu laden; aktueller Stand arbeitet mit `/var/tmp`-Kopie

Symptom	Ursache	Fix
Kein SSH-Login	Key nicht im Cloud-Init-Seed	`qm config` (`sshkeys`), Key-Datei, `qm cloudinit update` prüfen
`cloud-init: command not found`	`nocloud`-Image verwendet	Template mit `generic-amd64` neu bauen
`cidata` fehlt	Cloud-Init-Drive fehlt/falscher Bus	`scsi1` und `qm cloudinit update` prüfen
Cloud-Init übersprungen	Template wurde gebootet	Template neu bauen; künftig nur Clones booten
`qemu-guest-agent` fehlt	altes/fehlerhaftes Template	aktuelles Build-Skript, Template neu bauen
`qm agent <VMID> ping` scheitert	Paket/Dienst/Agent-Flag fehlt	Gastpaket, Dienst, `/dev/virtio-ports/`, `qm config` prüfen
IPv6 noch aktiv	alter Template-Stand oder kein Neustart	neues Template; Sysctl, Kernelparameter, Cold Boot prüfen
DHCP wartet	kein DHCP im Zielnetz	statische IPv4 beim Clone übergeben

10 · Source of Truth

> **important:** > Ausführbarer Source of Truth ist das interne Git-Repo, nicht ein eingebetteter Skriptblock in der KB.

- **Repo:** `corestream/proxmox-debian-cloudinit-scripts` (internes OneDev)
- **Installer:** `https://git.corestream.ch/corestream/proxmox-debian-cloudinit-scripts/~raw/master/install-on-proxmox.sh`
- **Installationspfad:** `/opt/corestream/proxmox-debian-cloudinit`

Die früher eingebetteten Vollkopien der Skripte wurden bewusst entfernt, damit Doku und Code nicht auseinanderlaufen.

Verwandt: Staging- und Produktionskonzept

Quellen

- Debian Cloud-Images (trixie, `latest`): https://cloud.debian.org/images/cloud/trix/latest/
- Debian Cloud image Paketmanifest (`debian-13-generic-amd64.json`): https://cloud.debian.org/images/cloud/trix/latest/debian-13-generic-amd64.json

- Proxmox VE Administration Guide – QEMU/Cloud-Init: https://pve.proxmox.com/pve-docs/chapter-qm.html
- Proxmox VE Wiki – Cloud-Init Support: https://pve.proxmox.com/wiki/Cloud-Init_Support

---

Quelle: Obsidian `Knowledge Base/Proxmox/Proxmox Debian Cloud-Init Strategie.md` · Stand: 2026-10-12 · Automatisiert publiziert durch Hermes.

Knowledge Base

Aktive technische Knowledge-Base-Einträge aus Obsidian.

NTP auf dem Proxmox-Cluster

NTP auf dem Proxmox-Cluster

> **abstract:** Zweck Auf allen Cluster-Nodes eine **deterministische Zeitsynchronisation** mit chrony einrichten. Eine stabil laufende Zeit ist im Proxmox-Cluster zwingend: abweichende Uhrzeiten können zu Corosync-Problemen, Zertifikatsfehlern (VNC-Konsole, API) und missverständlichen Log-Zeiten führen.

Grundlagen

- Proxmox VE installiert chrony als Standard-NTP-Daemon und sieht eine laufende Zeitsynchronisation als wichtig an.
- Cluster-Knoten werden nicht zwingend selbst als NTP-Server voneinander genutzt; alle Nodes synchronisieren gegen dieselben, verlässlichen Quellen.
- Empfehlung: eine interne, autoritative Quelle (z. B. Firewall/Router mit NTP-Dienst) mit `prefer`, plus öffentliche Fallback-Server (Swiss NTP Pool) mit `iburst`.
- Debian-Defaults (`pool 2.debian.pool.ntp.org`) und DHCP-abhängige Quellen (`sourcedir /run/chrony-dhcp`) bewusst deaktivieren, damit die Zeitquelle deterministisch ist.

Umstellung Schritt für Schritt

1. Konfiguration anpassen:

```
nano /etc/chrony/chrony.conf
```

2. Dienst neu starten:

```
systemctl restart chrony
```

3. Synchronisation prüfen:

```
chronyc sources -v
```

Erwartung: mindestens eine Quelle mit Status `^*` (aktiv) bzw. `^+` (Kandidat), z. B.:

```
^* 192.0.2.1
^+ ch.pool.ntp.org
```

Konfigurationsvorlage

Auf **jedem** Proxmox-Node identisch setzen (primäre interne Quelle durch die echte Management-Adresse ersetzen):

```
## Welcome to the chrony configuration file. See chrony.conf(5) for more
## information about usable directives.

## Include configuration files found in /etc/chrony/conf.d.
confdir /etc/chrony/conf.d

## === Proxmox Cluster NTP Configuration ===

## Primary NTP - Firewall (LAN / MGMT), interne Adresse eintragen
server <NTP-SERVER-LAN-IP> iburst prefer

## Fallback - Swiss NTP Pool
server ch.pool.ntp.org iburst

## --- Disabled defaults for deterministic cluster time ---
## pool 2.debian.pool.ntp.org iburst
## sourcedir /run/chrony-dhcp
## sourcedir /etc/chrony/sources.d
```

> **warning:** > Vor der Umsetzung prüfen, dass die interne NTP-Quelle (Firewall/Router) tatsächlich NTP-Anfragen beantwortet (`chronyc sources -v` muss `^*` zeigen) und dass NTP auf dem MGMT-Netz nicht per Firewall geblockt ist.

Verifikation Cluster-weit

```
pvecm status                # Cluster gesund?
for h in <node1> <node2>; do ssh $h chronyc tracking; done
```

`chronyc tracking` muss auf allen Nodes dieselbe Referenz/Stratum zeigen.

Quellen

- Proxmox VE Administration Guide – Time Synchronization: https://pve.proxmox.com/pve-docs/pve-admin-guide.html#_time_synchronization
- chrony-Dokumentation (`chrony.conf(5)`): https://chrony-project.org/doc/latest/chrony.conf.html
- Swiss NTP Pool: https://www.pool.ntp.org/zone/ch

Quelle: Obsidian `Knowledge Base/Proxmox/NTP Proxmox Cluster.md` · Stand: 2026-10-12 · Automatisiert publiziert durch Hermes.

UEFI Secure Boot Zertifikate 2023 (Proxmox EFI-Disk)

UEFI Secure Boot Zertifikate 2023 (Proxmox EFI-Disk)

> **abstract:** Problem EFI-Disk ohne `ms-cert=2023k`-Marker: die UEFI-2023-Zertifikate von Microsoft sind nicht eingespielt. Die originalen UEFI-2011-Zertifikate laufen **Juni 2026** ab. Die 2023-Zertifikate werden für Secure-Boot-Updates von Windows und gängigen Linux-Distributionen benötigt.

Hintergrund

- Die Firmware bootet nur Bootloader, deren Signaturzertifikate auf der EFI-Disk hinterlegt sind. Eine EFI-Disk mit nur den 2011-Zertifikaten **verwirft** mit den 2023-Zertifikaten signierte Bootloader.
- Seit `pve-edk2-firmware >= 4.2025.05-1` enthalten neu erstellte EFI-Disk **beide** Zertifikatsätze (2011 + 2023) und tragen den Marker `ms-cert=2023k` in der VM-Konfiguration.
- Für EFI-Disks, die **vor** dieser Version erstellt wurden, müssen die Zertifikate nachgerüstet werden.
- Marker `ms-cert=2023` / `ms-cert=2023w` bedeuten **unvollständige** Enrollment — im Start-Task-Log wird gewarnt; auch hier die Enrollment-Prozedur anwenden.

Lösung

In der Proxmox-Web-UI (VM kann laufen):

1. EFI-Disk in der Hardware-Ansicht auswählen → `Disk Action > Enroll Updated Certificates`.

2. Alternativ denselben Effekt über den API-Endpoint `/nodes/{node}/qemu/{vmid}/config` erreichen; die Enrollment greift beim nächsten Start der VM.

CLI (VM muss heruntergefahren sein):

```
qm enroll-efi-keys <VMID>
```

Windows mit BitLocker

Vor der Enrollment bei jedem BitLocker-Laufwerk in PowerShell (als Administrator):

```
manage-bde -protectors -disable C:
```

Wichtig für **jedes** BitLocker-Laufwerk; sonst verlangt der nächste Boot den BitLocker-Wiederherstellungsschlüssel. Für das Windows-Update und das Signieren des Bootloaders mit den 2023-Zertifikaten die Microsoft-Support-Artikel zu CVE-2023-24932 (Enterprise-Deployment-Guidance) beachten.

Verifikation

```
qm config <VMID> | grep ms-cert
## erwartet: ms-cert: 2023k
```

Quellen

- Proxmox VE Administration Guide – Secure Boot Certificate Expiration: https://pve.proxmox.com/pve-docs/chapter-qm.html
- Proxmox VE `qm(1)` Manpage – `qm enroll-efi-keys`: https://pve.proxmox.com/pve-docs/qm.1.html
- Microsoft – Windows Secure Boot certificate expiration and CA updates: https://support.microsoft.com/en-us/topic/windows-secure-boot-certificate-expiration-and-ca-updates-7ff40d33-95dc-4c3c-8725-a9b95457578e

---

Quelle: Obsidian `Knowledge Base/Proxmox/UEFI Secure Boot Zertifikate 2023.md` · Stand: 2026-10-12 · Automatisiert publiziert durch Hermes.