

Proxmox best Practice Cluster mit ZFS, CepH, SAN - Themenübersicher von KI analysiert

Dieser Proxmox-Workshop behandelt den praktischen Aufbau und Betrieb moderner Virtualisierungsumgebungen mit Proxmox VE. Im Mittelpunkt stehen nicht nur Installation und Bedienung der Weboberfläche, sondern vor allem die technischen Zusammenhänge hinter Storage, ZFS, Cluster, High Availability, Netzwerk, Migration, Backup und Monitoring.

Die Inhalte stammen aus einem zweitägigen praktischen Workshop und wurden aus gesprochenem Kursmaterial sowie den dazugehörigen Bildschirmaufzeichnungen zusammengeführt.

Inhaltsverzeichnis

- Proxmox Grundlagen und Architektur
- Storage und ZFS
- ZFS Snapshots und Recovery
- Proxmox Installation
- Post-Installation und Optimierung
- VM Best Practices
- Linux- und Windows-Gastsysteme
- Templates und Cloning
- Netzwerk, Bridges und VLANs
- Cluster und Quorum
- VM-Migration
- ZFS-Replikation
- High Availability

- [SAN und iSCSI](#)
- [Multipath](#)
- [Ceph](#)
- [Monitoring](#)
- [Backup und Disaster Recovery](#)
- [Updates und Versionswechsel](#)
- [Sicherheit und Hardening](#)
- [Troubleshooting](#)

1. Proxmox Grundlagen und Architektur

Proxmox VE ist eine Virtualisierungsplattform auf Basis von Debian Linux. Für virtuelle Maschinen kommt KVM zum Einsatz, zusätzlich können Linux-Container betrieben werden.

Im Workshop wurde zunächst geklärt, wie sich Proxmox von VMware ESXi und Hyper-V unterscheidet und welche Architektur für unterschiedliche Anforderungen sinnvoll ist.

- Standalone-Proxmox
- zwei Hosts mit ZFS-Replikation
- Proxmox-Cluster
- High Availability
- Ceph-Cluster
- SAN-basiertes Shared Storage

Entscheidend ist nicht, möglichst viele Funktionen einzusetzen, sondern eine Architektur zu wählen, die zu Verfügbarkeit, Budget, Hardware und gewünschter Wiederanlaufzeit passt.

2. Storage und ZFS

Ein zentraler Bestandteil des Workshops ist ZFS. Im Gegensatz zu klassischen Hardware-RAID-Systemen übernimmt ZFS mehrere Aufgaben gleichzeitig: RAID-Verwaltung, Datenintegrität, Volume-Management, Snapshots, Kompression und Caching.

Behandelte ZFS-Themen

- ZFS Mirror
- RAID-Z1, RAID-Z2 und RAID-Z3
- RAID10-ähnliche Mirror-Konfigurationen
- Checksummen und Datenintegrität
- Kompression
- ARC als RAM-Cache
- L2ARC als zusätzlicher SSD-Cache
- SLOG für synchrone Schreibvorgänge
- TRIM und AutoTrim
- Resilvering

Zusätzlich wurde gezeigt, wie der Zustand von Pools und Datenträgern mit `zpool status`, `zfs list` und SMART-Daten überprüft wird.

ZFS ermöglicht einen direkten Blick auf den Zustand der Datenträger und des Storage-Systems. Dadurch lassen sich Fehler häufig erkennen, bevor daraus ein tatsächlicher Ausfall entsteht.

3. ZFS Snapshots und Recovery

Snapshots sind einer der größten praktischen Vorteile von ZFS. Sie können sehr schnell erstellt werden und benötigen zunächst nur wenig zusätzlichen Speicher.

Im Workshop wurde `zfs-auto-snapshot` eingesetzt, um automatisch Wiederherstellungspunkte zu erzeugen.

- stündliche Snapshots
- tägliche Snapshots
- wöchentliche Snapshots
- unterschiedliche Aufbewahrungszeiten

Damit können beschädigte Systeme, Fehlkonfigurationen oder problematische Updates schnell zurückgesetzt werden.

Zusätzlich wurde der Unterschied zwischen einem destruktiven Rollback und einer nicht-destruktiven Wiederherstellung über einen Clone erläutert.

4. Proxmox Installation

Die Teilnehmer installierten Proxmox selbst. Dabei wurden sowohl die grundlegende Installation als auch die Storage-Auswahl behandelt.

- grafischer Proxmox-Installer
- Textinstaller
- ext4 und LVM
- ZFS RAID1
- EFI und Bootpartitionen
- Hostname
- Management-IP
- Gateway
- DNS

Besonders wichtig war die Entscheidung, ob ein System klassisch mit LVM oder direkt auf einem ZFS-Mirror installiert wird.

5. Proxmox Post-Installation und Optimierung

Nach der Installation wurde das System für den produktiven Einsatz vorbereitet.

- pve-no-subscription Repository
- zusätzliche Paketquellen
- VirtIO-Windows-ISO
- ZFS-Werkzeuge
- SSH-Hardening
- Locales
- Swappiness
- ZFS ARC-Konfiguration
- ZFS Blocksize
- AutoTrim
- AutoExpand

Ziel war eine reproduzierbare Grundkonfiguration, die anschließend für weitere Proxmox-Hosts verwendet werden kann.

6. Virtuelle Maschinen: Proxmox Best Practices

Eine virtuelle Maschine sollte nicht einfach mit den Standardwerten erstellt werden. Die virtuelle Hardware hat erheblichen Einfluss auf Leistung und spätere Migrationsmöglichkeiten.

- Q35 Machine Type
- UEFI statt Legacy BIOS
- EFI-Disk
- VirtIO SCSI Single
- VirtIO-Netzwerkkarten
- Discard / TRIM
- CPU-Typ und CPU-Features
- AES-Unterstützung
- Nested Virtualization
- RAM Ballooning
- Kernel Samepage Merging

7. Linux- und Windows-Gastsysteme

Linux unter Proxmox

Am Beispiel einer Linux-VM wurden Installation, EFI-Partitionierung, zusätzliche Datenträger und der QEMU Guest Agent behandelt.

Windows unter Proxmox

Bei Windows spielen die VirtIO-Treiber eine wichtige Rolle. Während der Installation müssen unter Umständen Storage- und Netzwerktreiber eingebunden werden.

- VirtIO Storage Driver

- Red Hat VirtIO Ethernet Adapter
- VirtIO Driver Installer
- QEMU Guest Agent

8. Templates und VM-Cloning

Wiederkehrende Systeme lassen sich als Templates vorbereiten und anschließend sehr schnell klonen.

- VM in Template umwandeln
- Full Clone
- Linked Clone
- VM-ID-Verwaltung

Gerade für Schulungen, Tests oder wiederkehrende Serverinstallationen können Templates viel Zeit sparen.

9. Proxmox Netzwerk, Bridges und VLANs

Proxmox verwendet Linux Bridges, um virtuelle Maschinen mit physischen Netzwerkkarten zu verbinden.

- physische Netzwerkinterfaces
- `vmbr0`
- Management-Netz
- VM-Netz
- Storage-Netz
- VLANs
- Bonding

Zusätzlich wurde gezeigt, wie eine beschädigte Netzwerkkonfiguration direkt über `/etc/network/interfaces` analysiert und repariert werden kann.

10. Proxmox Cluster und Quorum

Mehrere Proxmox-Hosts können zu einem Cluster verbunden werden. Damit entsteht zunächst eine gemeinsame Verwaltungs- und Konfigurationsumgebung.

- Cluster-Nodes
- Corosync
- Quorum
- `/etc/pve`
- gemeinsame Cluster-Konfiguration

Ein Cluster bedeutet dabei nicht automatisch Hochverfügbarkeit. HA, Storage und Replikation sind zusätzliche Funktionen, die getrennt geplant werden müssen.

11. VM-Migration und Live Migration

Im Workshop wurden virtuelle Maschinen zwischen unterschiedlichen Proxmox-Hosts verschoben.

Dabei hängt der Ablauf stark vom verwendeten Storage ab.

- lokaler Storage
- ZFS-Replikation
- SAN
- Ceph
- Online-Migration
- Migration des RAM-Zustands
- Contrack State

Bei Shared Storage liegen die VM-Daten bereits für mehrere Hosts erreichbar vor. Dadurch muss bei einer Live Migration im Wesentlichen der laufende Zustand der VM übertragen werden.

12. ZFS-Replikation zwischen Proxmox-Hosts

Mit ZFS können virtuelle Festplatten regelmäßig auf einen zweiten Proxmox-Host repliziert werden.

Nach der ersten vollständigen Übertragung werden nur noch die Änderungen zwischen zwei Snapshots übertragen.

Damit lässt sich auch ohne komplexes Shared Storage eine kostengünstige Wiederanlaufstrategie aufbauen.

Die Replikation ist nicht synchron. Der Datenstand des zweiten Hosts kann daher je nach Replikationsintervall einige Minuten hinter dem produktiven System liegen.

13. High Availability mit Proxmox

Mit Proxmox HA können virtuelle Maschinen nach einem Ausfall automatisch auf einem anderen Cluster-Node gestartet werden.

- HA-Ressourcen
- Failback
- Auto-Rebalance
- Node Affinity
- Prioritäten
- Fencing
- Watchdog

Im Workshop wurden Hosts auch bewusst hart abgeschaltet, um das tatsächliche Verhalten bei einem Ausfall zu demonstrieren.

Dabei wurde deutlich: High Availability verbessert die Wiederanlaufzeit, ersetzt aber weder Datensicherung noch Replikation.

14. SAN und iSCSI mit Proxmox

Als klassisches Shared-Storage-Modell wurde ein SAN auf Basis von TrueNAS und iSCSI verwendet.

- ZVOLS
- iSCSI Targets
- Shared Storage
- Proxmox Storage-Anbindung
- LVM auf gemeinsamem Storage

Der Vorteil liegt in gemeinsam nutzbarem Storage und einfacher Live Migration. Gleichzeitig entsteht aber eine zusätzliche zentrale Infrastruktur, die selbst redundant ausgelegt werden muss.

15. Multipath für redundante SAN-Verbindungen

Ein produktives SAN sollte nicht nur über einen einzigen Netzwerkpfad erreichbar sein. Mit Multipath werden mehrere Storage-Verbindungen zu einem logischen Gerät zusammengeführt.

Behandelt wurden unter anderem:

- WWIDs
- `multipath -a`
- `multipath -r`
- `multipath -ll`
- `/dev/mapper`
- `lsblk`

Auch der Ausfall eines einzelnen SAN-Pfades wurde praktisch demonstriert.

16. Ceph Storage mit Proxmox

Ceph verteilt Daten über mehrere Proxmox-Nodes und stellt dadurch gemeinsam nutzbaren Storage ohne klassisches externes SAN bereit.

Dadurch werden Live Migration und Hochverfügbarkeit möglich, allerdings steigen Hardware-, Netzwerk- und Speicherbedarf.

- mehrere Cluster-Nodes
- schnelle SSDs
- schnelle Netzwerkverbindungen
- replizierte Daten
- Shared Storage

Im Workshop wurde Ceph bewusst mit SAN und lokaler ZFS-Replikation verglichen.

17. Proxmox Monitoring mit Checkmk

Die Proxmox-Oberfläche zeigt nicht jeden Hardware- oder Storagefehler ausreichend deutlich an. Deshalb ist externes Monitoring für produktive Installationen wichtig.

Überwacht werden sollten beispielsweise:

- ZFS Pool Status
- SMART-Werte
- SSD Wearout
- Speicherplatz
- Multipath-Verbindungen
- Cluster-Zustand
- Backup-Jobs
- Replikationen

Im Workshop wurde dafür unter anderem Checkmk verwendet und der Agent sowohl auf Linux als auch unter Windows installiert.

18. Backup und Disaster Recovery

High Availability, Snapshots und Replikation sind keine vollständige Datensicherungsstrategie.

Deshalb wurden mehrere Sicherungsansätze behandelt:

- klassische Proxmox Backups
- Proxmox Backup Server
- ZFS Send und Receive
- inkrementelle ZFS-Replikation
- Offsite-Backup
- Offline-Sicherung

Besonders wichtig ist die räumliche und technische Trennung der Backups vom Produktivsystem. Dadurch kann verhindert werden, dass ein Angriff oder Hardwarefehler gleichzeitig Produktion und Sicherung betrifft.

19. Proxmox Updates und Versionswechsel

Da Proxmox auf Debian basiert, erfolgt die Systempflege über die Linux-Paketverwaltung.

- APT Updates
- Kernel Updates
- Repository-Verwaltung
- `apt autoremove`
- `dpkg --audit`
- Proxmox Versionswechsel

Ein weiterer Punkt waren ZFS Feature Upgrades. Diese sollten nicht unmittelbar nach Veröffentlichung aktiviert werden, damit im Notfall weiterhin kompatible Recovery-Medien verfügbar sind.

20. Proxmox Sicherheit und Hardening

Eine produktive Virtualisierungsplattform sollte nicht ungeschützt im normalen Anwendernetz betrieben werden.

Behandelte Maßnahmen waren unter anderem:

- separates Management-Netz
- SSH Hardening
- SSH Keys
- Passwort-Login für root deaktivieren
- Zugriff auf das Webinterface begrenzen
- Proxmox Firewall
- Zwei-Faktor-Authentifizierung

21. Troubleshooting und Proxmox Kommandozeile

Ein wesentlicher Teil des Workshops bestand darin, nicht nur die Proxmox-GUI zu verwenden, sondern den tatsächlichen Zustand des Systems über Linux-Werkzeuge analysieren zu können.

Typische Werkzeuge waren:

- `zpool`
- `zfs`
- `smartctl`
- `multipath`
- `lsblk`
- `ip`
- `qm`
- `apt`
- `dpkg`

Damit können Netzwerkprobleme, Storagefehler, beschädigte Pools, Replikationsprobleme und viele andere Störungen auch dann untersucht werden, wenn die grafische Oberfläche keine eindeutige Ursache zeigt.

Zusammenfassung des Proxmox Workshops

Der Workshop behandelt Proxmox nicht nur als Ersatz für VMware oder Hyper-V, sondern als komplette Virtualisierungsplattform mit unterschiedlichen Architekturvarianten.

Ein besonderer Schwerpunkt liegt auf der Kombination aus Proxmox und ZFS. Damit lassen sich mit vergleichsweise einfacher Hardware leistungsfähige, gut überwachte und schnell wiederherstellbare Virtualisierungsumgebungen aufbauen.

Gleichzeitig werden auch größere Konzepte wie High Availability, Ceph, SAN, iSCSI und Multipath praktisch demonstriert und miteinander verglichen.

Ziel ist es, nicht nur einzelne Funktionen bedienen zu können, sondern die technischen Zusammenhänge zu verstehen und daraus für unterschiedliche Kunden und Anforderungen eine passende Proxmox-Infrastruktur aufzubauen.

Version #2

Erstellt: 2026-10-02 09:56:36 UTC von Christian Zengel (sysops GmbH)

Zuletzt aktualisiert: 2026-10-02 09:59:32 UTC von Christian Zengel (sysops GmbH)