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

> Dieser Artikel ist KI generiert. Als Basis diente die Sprache des Referenten und die ca 800 Screenshots!

Zusammenfassung des Proxmox VE Workshops

Der zweitägige Workshop behandelt Proxmox VE nicht nur als Oberfläche zum Starten virtueller Maschinen, sondern als vollständige Virtualisierungsplattform. Im Mittelpunkt stehen Architektur, Storage, ZFS, Netzwerk, Cluster, High Availability, Backup, Monitoring und die praktische Fehlersuche.

Dabei werden verschiedene Betriebsmodelle miteinander verglichen: vom einzelnen Proxmox-Host über zwei Systeme mit ZFS-Replikation bis hin zu Clustern mit Ceph oder klassischem SAN-Storage.

1. Proxmox Grundlagen und Architektur

Proxmox VE basiert auf Debian Linux und verwendet KVM für virtuelle Maschinen sowie Linux-Container für containerisierte Systeme. Dadurch unterscheidet sich Proxmox in einigen Punkten deutlich von klassischen VMware- oder Hyper-V-Umgebungen.

- **Standalone-System:** Ein einzelner Proxmox-Host ist einfach und kostengünstig, bietet aber bei einem Hardwareausfall zunächst keinen zweiten Host für den Wiederanlauf.
- **ZFS-Replikation:** Zwei oder mehr Hosts können lokale VM-Daten regelmäßig untereinander replizieren. Dadurch lässt sich ohne zentrales SAN bereits eine relativ

einfache Wiederanlaufstrategie realisieren.
- **Cluster:** Mehrere Proxmox-Systeme werden gemeinsam verwaltet und teilen wichtige Konfigurationsinformationen. Ein Cluster allein bedeutet jedoch noch keine automatische Hochverfügbarkeit.
- **Ceph oder SAN:** Shared Storage ermöglicht mehreren Hosts den Zugriff auf dieselben VM-Daten und vereinfacht dadurch unter anderem Live-Migration und HA.

2. Storage und ZFS

ZFS ist einer der Schwerpunkte des Workshops. Es verbindet RAID-Funktionen, Volume-Management, Datenintegrität, Snapshots, Kompression und Caching in einem Storage-System.

- **Mirror und RAID-Z:** Je nach gewünschter Leistung, Kapazität und Ausfallsicherheit können unterschiedliche ZFS-VDEV-Strukturen aufgebaut werden.
- **Checksummen:** ZFS prüft Daten und Metadaten mit Checksummen und kann dadurch nicht nur einen ausgefallenen Datenträger, sondern auch beschädigte Daten erkennen.
- **Kompression:** Daten können transparent komprimiert werden. Gerade VM-Datenträger erreichen dabei teilweise erhebliche Kompressionsraten.
- **ARC und L2ARC:** ZFS verwendet freien Arbeitsspeicher intensiv als Cache. Optional können zusätzliche schnelle Datenträger als weiterer Lesecache eingesetzt werden.
- **SLOG:** Für synchrone Schreibzugriffe kann ein separates schnelles Log-Gerät eingesetzt werden. Das ist vor allem bei entsprechenden Storage-Workloads relevant.
- **SMART und Wearout:** Neben dem eigentlichen ZFS-Pool müssen auch die physischen SSDs und HDDs überwacht werden. Besonders bei SSDs sind Schreibvolumen und Verschleißwerte wichtige Betriebsparameter.

3. Snapshots und Recovery

ZFS-Snapshots ermöglichen sehr schnelle Wiederherstellungspunkte. Sie ersetzen kein Backup, sind aber besonders hilfreich bei Updates, Fehlkonfigurationen oder versehentlich veränderten Daten.

- **Automatische Snapshots:** Mit zfs-auto-snapshot können beispielsweise stündliche, tägliche und wöchentliche Wiederherstellungspunkte automatisch erzeugt werden.
- **Rollback:** Ein Dataset kann direkt auf einen früheren Snapshot zurückgesetzt werden. Da dies destruktiv sein kann, muss vor dem Rollback geprüft werden, welche neueren Änderungen dadurch verloren gehen.

- **Clone statt Rollback:** Für eine nicht-destruktive Wiederherstellung kann ein Snapshot geklont und separat bereitgestellt werden. Daten lassen sich anschließend gezielt aus diesem Zustand zurückholen.

4. Installation und Post-Installation

Die Proxmox-Installation wurde vollständig durchgeführt. Dabei ging es nicht nur darum, den Installer durchzuklicken, sondern bereits bei der Installation Storage- und Netzwerkentscheidungen bewusst zu treffen.

- **ZFS RAID1 als Systemstorage:** Zwei Datenträger können direkt während der Installation als ZFS-Mirror eingerichtet werden. Dadurch ist auch das Proxmox-System selbst redundant.
- **Management-Netzwerk:** Hostname, feste IP-Adresse, Gateway und DNS sollten vor der Installation geplant sein. Spätere Änderungen sind möglich, können bei Clustern aber deutlich aufwendiger werden.
- **Repositories:** Nach der Installation muss entschieden werden, ob das Enterprise-Repository oder beispielsweise das No-Subscription-Repository verwendet wird.
- **Post-Install-Konfiguration:** Zusätzlich wurden unter anderem VirtIO-Treiber, ZFS-Hilfsprogramme, AutoTrim, Snapshot-Einstellungen und weitere Systemparameter eingerichtet.

5. Virtuelle Maschinen richtig konfigurieren

Die virtuelle Hardware beeinflusst Leistung, Kompatibilität und spätere Migrationen. Deshalb wurden VMs nicht einfach mit beliebigen Standardwerten angelegt.

- **Q35 und UEFI:** Für moderne Betriebssysteme bietet sich eine moderne virtuelle Hardwareplattform mit UEFI und EFI-Disk an.
- **VirtIO SCSI:** VirtIO-Geräte vermeiden unnötige Hardwareemulation und bieten bei Storage- und Netzwerkzugriffen eine wesentlich geeignetere virtuelle Schnittstelle.
- **CPU-Konfiguration:** Der CPU-Typ beeinflusst sowohl die verfügbare Leistung und Features als auch die Möglichkeit, eine VM später auf einen anderen Host zu migrieren.
- **Ballooning und KSM:** Proxmox kann Arbeitsspeicher zwischen VMs flexibler verwalten. Diese Funktionen sollten jedoch bewusst eingesetzt und nicht als Ersatz für ausreichenden physischen RAM verstanden werden.
- **QEMU Guest Agent:** Der Agent verbessert die Kommunikation zwischen Proxmox und dem Gastsystem und liefert beispielsweise zusätzliche Statusinformationen.

6. Linux- und Windows-Gastsysteme

Im Workshop wurden sowohl Linux- als auch Windows-Systeme praktisch installiert. Dadurch konnten unterschiedliche Anforderungen an virtuelle Hardware und Treiber direkt gezeigt werden.

- **Linux:** Installation, EFI, Partitionierung, zusätzliche virtuelle Datenträger und QEMU Guest Agent wurden praktisch behandelt.
- **Windows:** Windows benötigt für optimale Leistung VirtIO-Treiber. Während der Installation wurden Storage-Treiber sowie anschließend Netzwerk- und weitere VirtIO-Treiber eingebunden.

7. Netzwerk und Bridges

Virtuelle Maschinen kommunizieren in Proxmox normalerweise über Linux Bridges. Eine Bridge verbindet dabei virtuelle Netzwerkkarten der VMs mit physischen Interfaces oder VLANs.

- **Management-Netz:** Die Weboberfläche und SSH-Zugänge des Hypervisors sollten möglichst nicht im normalen Client- oder Gastnetz erreichbar sein. Ein getrenntes Management-Netz reduziert die Systeme, von denen ein Angriff auf den Hypervisor überhaupt möglich ist.
- **VM-Netze:** Virtuelle Server können über eigene Bridges und VLANs in getrennte Netzwerksegmente eingebunden werden.
- **Storage-Netze:** Ceph, iSCSI oder Replikation erzeugen teilweise erheblichen Netzwerkverkehr. Separate Interfaces oder VLANs verhindern, dass dieser Verkehr das normale Management- oder VM-Netz unnötig belastet.
- **/etc/network/interfaces:** Wenn die Weboberfläche wegen einer fehlerhaften Netzwerkkonfiguration nicht mehr erreichbar ist, kann die Linux-Netzwerkkonfiguration direkt auf der Konsole analysiert und repariert werden.

8. Cluster, Quorum und Migration

Mehrere Hosts können zu einem Proxmox-Cluster verbunden werden. Dadurch werden wesentliche Konfigurationen über die Nodes verteilt und die Systeme gemeinsam verwaltet.

- **Corosync:** Corosync übernimmt einen wichtigen Teil der Kommunikation und Mitgliedschaftsverwaltung innerhalb des Clusters.
- **Quorum:** Das Mehrheitsprinzip verhindert, dass getrennte Teile eines Clusters unabhängig voneinander widersprüchliche Änderungen vornehmen.
- **VM-Migration:** Laufende VMs können zwischen Nodes verschoben werden. Wie viel dabei übertragen werden muss, hängt stark davon ab, ob lokaler oder gemeinsam

genutzter Storage verwendet wird.
- **Shared Storage:** Liegt die virtuelle Festplatte bereits auf Ceph oder SAN, muss bei einer Live Migration nicht zusätzlich die gesamte virtuelle Disk zwischen den Hosts kopiert werden.

9. ZFS-Replikation und High Availability

ZFS-Replikation und Proxmox HA lösen unterschiedliche Probleme und können miteinander kombiniert werden.

- **ZFS-Replikation:** Nach einer ersten vollständigen Übertragung werden nur die Unterschiede zwischen ZFS-Snapshots übertragen. Dadurch kann ein zweiter Host relativ aktuell gehalten werden.
- **Replikationsintervall:** Da die Replikation nicht permanent synchron erfolgt, bestimmt das eingestellte Intervall den möglichen Datenverlust bei einem plötzlichen Ausfall.
- **HA-Ressourcen:** Proxmox kann festlegen, welche VMs nach einem Node-Ausfall automatisch auf einem anderen Host gestartet werden sollen.
- **Node Affinity:** Über Prioritäten lässt sich definieren, auf welchen Hosts bestimmte virtuelle Maschinen bevorzugt laufen sollen.
- **Fencing und Watchdog:** Vor einem automatischen Neustart einer VM muss der Cluster sicherstellen, dass die VM nicht gleichzeitig noch auf dem ausgefallenen beziehungsweise isolierten Host aktiv ist.

10. SAN, iSCSI und Multipath

Als Alternative zu lokalem ZFS und Ceph wurde klassisches Shared Storage mit TrueNAS und iSCSI behandelt.

- **ZVOL:** TrueNAS stellt ein ZFS-Volume als Blockgerät bereit, das anschließend per iSCSI an die Proxmox-Hosts exportiert werden kann.
- **Shared Storage:** Mehrere Proxmox-Nodes können auf denselben zentralen Speicher zugreifen. Das erleichtert Live-Migration und klassische Clusterkonzepte.
- **Multipath:** Produktiver SAN-Zugriff sollte nicht von einem einzigen Netzwerkpfad abhängen. Multipath kombiniert mehrere unabhängige Wege zum selben Storage-Gerät.
- **WWID:** Anstelle wechselnder Linux-Gerätenamen wie /dev/sde oder /dev/sdf wird das Storage-Gerät über seine eindeutige WWID identifiziert.
- **Pfadausfall:** Fällt eine Netzwerkverbindung aus, kann ein korrekt konfiguriertes Multipath-System den Zugriff über einen verbleibenden Pfad fortsetzen.

11. Ceph

Ceph stellt verteilten Storage direkt über mehrere Cluster-Nodes bereit. Dadurch wird kein separates klassisches SAN benötigt.

- **Datenverteilung:** Daten werden auf mehrere Systeme verteilt und entsprechend der konfigurierten Replikation mehrfach gespeichert.
- **Shared Storage:** Alle beteiligten Proxmox-Nodes greifen auf denselben logischen Storage-Bestand zu.
- **Netzwerk:** Ceph erzeugt zusätzlichen internen Datenverkehr. Netzwerkbandbreite und Latenz sind deshalb entscheidende Faktoren für die Gesamtleistung.
- **Kapazität:** Durch die Replikation steht nicht die gesamte physisch installierte Rohkapazität für Nutzdaten zur Verfügung.

12. Backup und Disaster Recovery

HA, Replikation und Snapshots reduzieren Ausfallzeiten, ersetzen aber kein unabhängiges Backup. Ein Fehler oder Angriff kann ansonsten sehr schnell auf mehrere replizierte Systeme übertragen werden.

- **Proxmox Backup Server:** PBS integriert sich direkt in Proxmox VE und bietet deduplizierte und komprimierte Backups von VMs und Containern.
- **ZFS Send/Receive:** ZFS-Snapshots können vollständig oder inkrementell auf ein anderes ZFS-System übertragen werden.
- **Offsite-Sicherung:** Mindestens eine Sicherung sollte räumlich getrennt vom Produktivsystem liegen, damit beispielsweise Brand, Diebstahl oder ein größerer Hardwareausfall nicht Produktion und Backup gleichzeitig trifft.
- **Offline- beziehungsweise getrennte Backups:** Eine Sicherung, auf die das Produktivsystem nicht dauerhaft schreiben kann, bietet zusätzlichen Schutz gegen Ransomware und Fehlbedienung.

13. Monitoring

Ein Virtualisierungscluster darf nicht nur anhand der grünen Symbole in der Proxmox-Weboberfläche überwacht werden. Gerade Storage- und Hardwareprobleme müssen aktiv kontrolliert werden.

- **ZFS Pool Status:** Ein Pool kann bereits DEGRADED sein, obwohl die darauf laufenden VMs zunächst weiterhin funktionieren.
- **SMART und SSD-Wearout:** Physische Datenträger müssen auf Fehlerwerte, Betriebsstunden und Verschleiß überwacht werden.

- **Multipath:** Ein SAN kann weiterhin funktionieren, obwohl bereits einer von zwei redundanten Pfaden ausgefallen ist. Ohne Monitoring würde dieser Zustand möglicherweise erst beim Ausfall des zweiten Pfades auffallen.
- **Checkmk:** Ein externes Monitoring sammelt diese Zustände zentral und kann Alarmierungen auslösen, bevor aus einer verlorenen Redundanz ein tatsächlicher Ausfall wird.

14. Sicherheit und Hardening

Der Proxmox-Hypervisor ist die zentrale Kontrolle über viele virtuelle Server. Ein kompromittierter Proxmox-Host kann deshalb wesentlich größere Auswirkungen haben als der Angriff auf eine einzelne virtuelle Maschine.

- **Separates Management-Netz:** Proxmox-Webinterface und SSH sollten möglichst nur aus einem administrativen Netz oder über VPN erreichbar sein. Anwender-PCs und normale Server benötigen normalerweise keinen direkten Zugriff auf die Managementoberfläche.
- **SSH Hardening:** Nicht benötigte oder unsichere Anmeldeverfahren werden abgeschaltet. Gleichzeitig sollte der SSH-Dienst nur aus Netzen erreichbar sein, aus denen tatsächlich administriert wird.
- **SSH Keys:** Schlüsselbasierte Anmeldung vermeidet die Übertragung und regelmäßige Eingabe administrativer Passwörter. Private Schlüssel müssen allerdings ebenso sorgfältig geschützt werden wie Passwörter.
- **Passwort-Login für root deaktivieren:** Ist eine funktionierende Schlüsselanmeldung eingerichtet, kann der direkte SSH-Login von root per Passwort abgeschaltet werden. Damit entfällt ein besonders offensichtliches Ziel für automatisierte Passwortangriffe.
- **Zugriff auf das Webinterface begrenzen:** Port 8006 sollte nicht unnötig aus beliebigen Netzen oder direkt aus dem Internet erreichbar sein. Firewall-Regeln und Netzwerksegmentierung beschränken die erreichbaren Management-Systeme.
- **Proxmox Firewall:** Die integrierte Firewall kann Regeln auf Datacenter-, Host- und VM-Ebene anwenden. Dadurch lassen sich sowohl Managementdienste als auch virtuelle Systeme gezielt absichern.
- **Zwei-Faktor-Authentifizierung:** Für administrative Konten ergänzt ein zweiter Faktor das Passwort. Ein gestohlenes oder erratenes Passwort reicht damit allein nicht mehr zur Anmeldung an der Proxmox-Oberfläche.

15. Updates und Troubleshooting

Proxmox basiert auf Debian. Deshalb sind Linux-Kenntnisse gerade bei Updates und Fehleranalysen ein wesentlicher Vorteil.

- **Updates:** Paket- und Kernelupdates werden über die Debian-Paketverwaltung eingespielt. Vor größeren Versionswechseln sollten Kompatibilität, Clusterzustand und vorhandene Backups geprüft werden.
- **ZFS Feature Upgrades:** Neue ZFS-Pool-Features sollten bewusst aktiviert werden. Ein einmal aktualisierter Pool kann gegebenenfalls nicht mehr mit älteren ZFS-Versionen importiert werden.
- **Kommandozeile:** Befehle wie zpool, zfs, smartctl, multipath, lsblk, ip und qm zeigen häufig wesentlich genauer als die Weboberfläche, was im System tatsächlich passiert.
- **Recovery:** Behandelt wurden unter anderem ausgefallene ZFS-Datenträger, beschädigte Netzwerkkonfigurationen, Clusterprobleme und das Wiederherstellen früherer Zustände.

Fazit

Der Workshop verfolgt bewusst keinen Ansatz nach dem Motto „drei Hosts installieren und auf HA klicken“. Stattdessen werden die einzelnen Bausteine einer Proxmox-Infrastruktur getrennt betrachtet und anschließend kombiniert.

Dadurch wird nachvollziehbar, welche Aufgabe ZFS, Replikation, Ceph, SAN, Multipath, Cluster, HA und Backup jeweils übernehmen – und welche Funktion sie ausdrücklich nicht ersetzen.

Das Ziel ist eine Proxmox-Infrastruktur, deren Verhalten auch bei Festplatten-, Netzwerk-, Host- oder Storage-Ausfällen verstanden und vorhersehbar bleibt.

Quellen und technische Referenzen

Die folgenden Quellen dienen als technische Referenz und Vertiefung zu den im Workshop behandelten Themen. Verlinkt sind überwiegend die offiziellen Dokumentationen der eingesetzten Projekte und Hersteller.

Proxmox VE

- **Proxmox VE Administration Guide** – Zentrale Dokumentation zu Installation, virtuellen Maschinen, Containern, Storage, Netzwerk, Cluster, High Availability, Firewall und Benutzerverwaltung.
- **Proxmox VE Cluster Manager** – Technische Referenz zu Cluster-Aufbau, Corosync, Quorum und der Verwaltung mehrerer Proxmox-Nodes.
- **Proxmox Cluster File System – pmxcfs** – Erläutert das verteilte Cluster-Dateisystem hinter `/etc/pve` und die gemeinsame Konfiguration innerhalb eines Proxmox-Clusters.
- **Proxmox VE High Availability** – Dokumentation zu HA-Ressourcen, Fencing, Watchdog, Failover und automatischem Wiederanlauf von virtuellen Maschinen.
- **Proxmox VE Storage Replication** – Beschreibung der ZFS-basierten Replikation virtueller Festplatten zwischen Proxmox-Nodes.
- **Proxmox VE Storage** – Überblick über lokale und gemeinsame Storage-Systeme wie ZFS, LVM, NFS, iSCSI, Ceph und Proxmox Backup Server.
- **Proxmox VE Ceph** – Einrichtung und Betrieb von verteiltem Ceph-Storage innerhalb eines Proxmox-Clusters.
- **Proxmox VE Network Configuration** – Dokumentation zu Linux Bridges, VLANs, Bonds sowie Management-, VM- und Storage-Netzen.

Proxmox Backup Server

- **Proxmox Backup Server Dokumentation** – Offizielle Dokumentation zu Backup, Restore, Deduplizierung, Kompression, Pruning, Garbage Collection und Datastores.
- **Proxmox VE und PBS Integration** – Beschreibt die Einbindung eines Proxmox Backup Servers in einzelne Proxmox-Hosts oder Cluster.

OpenZFS

- **OpenZFS Dokumentation** – Zentrale technische Referenz zu Pools, Datasets, Snapshots, Kompression, ARC, Scrubs, Resilvering und Fehlerbehandlung.
- **OpenZFS – zpool status** – Referenz für die Prüfung des Gesundheitszustands eines ZFS-Pools sowie für Scrub- und Resilver-Vorgänge.
- **OpenZFS – ZFS Snapshots** – Dokumentation zur Erstellung und Verwaltung von ZFS-Snapshots.
- **OpenZFS – Send and Receive** – Beschreibung vollständiger und inkrementeller Übertragungen von ZFS-Snapshots für Replikation und Backup.

TrueNAS, iSCSI und Multipath

- **TrueNAS Dokumentation** – Zentrale Dokumentation zu ZFS, Storage Pools, Datasets, ZVOLs, Netzwerk und iSCSI.
- **TrueNAS – ZVOLs** – Erläutert das Erstellen und Verwalten von ZFS Block Devices, die unter anderem als Grundlage für iSCSI-Storage dienen.
- **TrueNAS – Networking und Multipath** – Hinweise zu redundanten Storage-Netzen, iSCSI-Pfaden und Multipath-Konfigurationen.

Checkmk Monitoring

- **Checkmk Dokumentation** – Zentrale Dokumentation zu Hosts, Services, Agenten, Monitoring-Regeln und Benachrichtigungen.
- **Checkmk – Linux überwachen** – Installation und Betrieb des Checkmk-Agenten auf Linux-Systemen.
- **Checkmk – Windows überwachen** – Installation und Betrieb des Checkmk-Agenten auf Windows-Systemen.

Weitere technische Referenzen

- **Debian Dokumentation** – Technische Grundlage für viele administrative Aufgaben auf dem Proxmox-Hostsystem.
- **QEMU Dokumentation** – Hintergrundinformationen zur Virtualisierungsschicht und zur virtuellen Hardware von QEMU/KVM.
- **KVM – Kernel-based Virtual Machine** – Hintergrundinformationen zur Linux-Virtualisierung mit KVM.

Hinweis: Menübezeichnungen, Versionsstände und einzelne Funktionen können sich mit zukünftigen Versionen von Proxmox VE, OpenZFS, TrueNAS und Checkmk ändern. Für produktive Installationen sollte deshalb immer zusätzlich die Dokumentation der tatsächlich eingesetzten Version geprüft werden.

Im Kurs verwendete wichtige Links

- **Proxmox VE – Cluster Manager** – Offizielle Proxmox-Dokumentation zum Aufbau und zur Verwaltung eines Proxmox-Clusters, inklusive Corosync, Quorum und Cluster-Kommunikation.
- **Proxmox VE – Full Mesh Network for Ceph Server** – Beschreibung eines direkten Full-Mesh-Netzwerks zwischen Ceph-Nodes, um Storage-Verkehr ohne zentralen Switch zwischen den Hosts zu übertragen.
- **Proxmox VE – Deploy Hyper-Converged Ceph Cluster** – Praxisanleitung zum Aufbau eines hyperkonvergenten Proxmox-Ceph-Clusters, bei dem Compute- und Storage-Funktionen auf denselben Nodes betrieben werden.

Version #5
Erstellt: 2026-10-02 09:56:36 UTC von Christian Zengel (sysops GmbH)
Zuletzt aktualisiert: 2026-10-02 14:13:17 UTC von Christian Zengel (sysops GmbH)