

Externe Block Devices in Proxmox – Fluch, Segen und zwei Auswege

Konzepte und Vergleich für Snapshot- fähigen Shared-Storage

Alexander Wirt <alexander.wirt@credativ.de>

Who am I?

Hier ein kurzer Einblick in meinen Werdegang und meine Erfahrungen:

Open Source-Engagement

Seit 1997 aktiv in der Open Source-Community, einschließlich der Mitgliedschaft in verschiedenen OSS-Projekten (z.B. Debian).



Debian Kern-Rollen

Verantwortlich als Debian Listmaster, Backports ftpmaster und salsa.d.o Admin.

Professionelle Laufbahn

Seit 2004 bei credativ, das inzwischen Teil von NetApp ist wieder credativ ist ;).



Führungsrollen

Erfahren in den Rollen Technical Lead, Project Lead, Scrum Master und nun anscheinend auch CTO.

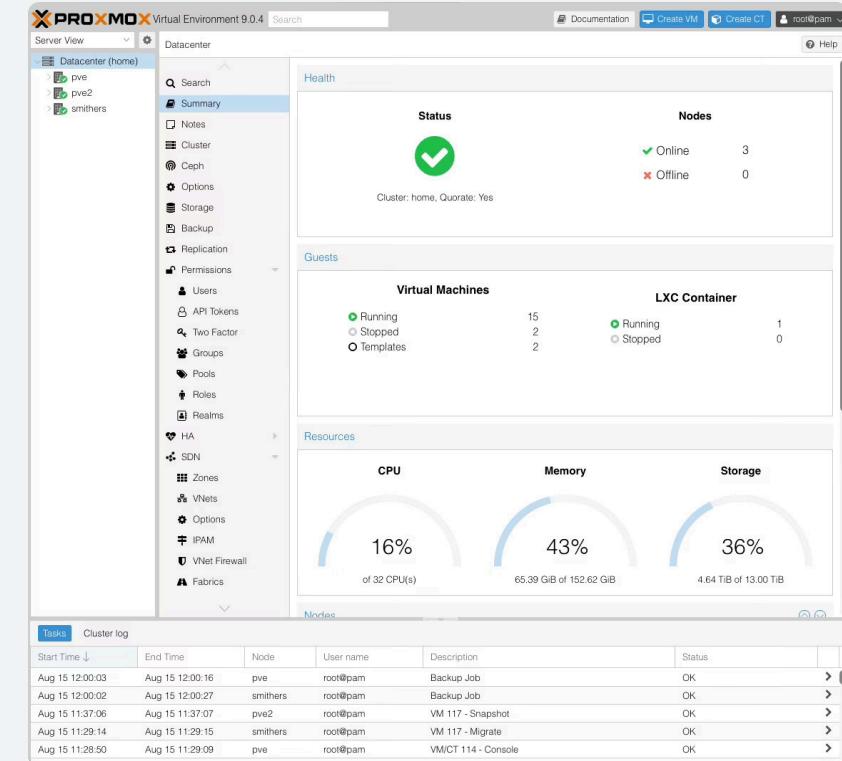


Einführung

Proxmox Virtual Environment bietet verschiedene Storage-Optionen für unterschiedliche Anforderungen. Ein langjähriges Problem war jedoch die fehlende Snapshot-Funktionalität auf klassischem SAN-Storage:

- Traditionelles iSCSI/FC mit LVM Thick: keine Snapshots möglich
 - Ceph (komplex, eigene Infrastruktur)
 - ZFS (teilweise shared)
 - NFS mit QCOW2 (Performance, Latenz)
 - Backup-Only-Lösungen (kein Live-Snapshot)

Neu in Proxmox 9: Snapshots als Volume Chains (Technology Preview)



Typische Storage-Backends & Snapshot-Fähigkeit

Lokales Filesystem

QCOW2 intern/extern

✓ Snapshots möglich

Nicht shared

LVM Thick

Shared möglich (iSCSI/FC)

Bisher keine Snapshots

→ Ziel der neuen Lösung

LVM-Thin

Lokale Volumes

✓ Snapshots möglich

Nicht shared

ZFS

Copy-on-Write Filesystem

✓ Snapshots möglich

~ Teils shared (zfs send/recv)

Ceph RBD

Verteiltes Storage-System

✓ Snapshots möglich

✓ Shared über Cluster

NFS/CIFS

Netzwerk-Filesysteme

✓ Snapshots mit QCOW2

~ Performance-Kompromisse

Das Ziel: Snapshots auf hochperformantem shared Block-Storage ohne Ceph-Komplexität.

Alternative Lösung: HA-Storage-VM mit iSCSI + NFS4



SAN-Storage



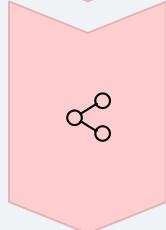
Physischer Block-Storage über iSCSI/FC mit LUN-Zuweisung an Storage-VM



Storage-VM



Virtuelle Maschine mit HA-Konfiguration, mountet SAN-LUN, nutzt QCOW2/ZFS intern



NFS4-Export

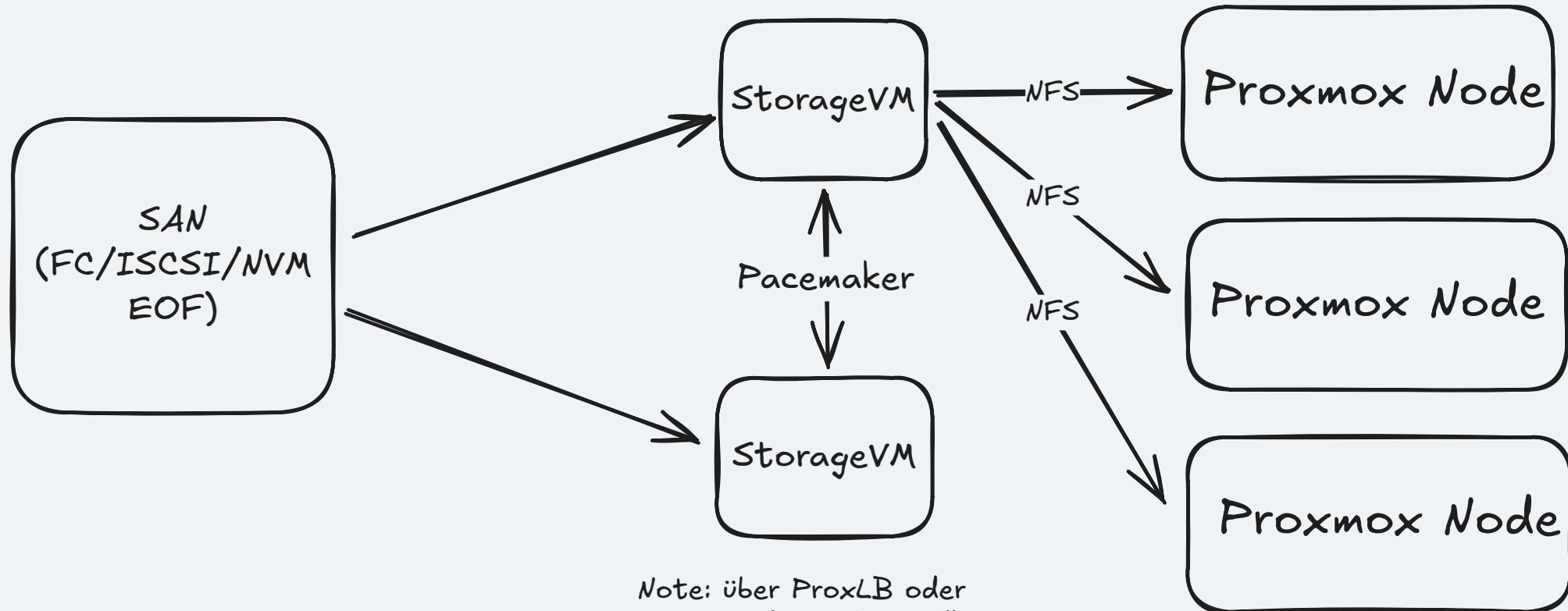


VM exportiert Storage als NFS4 an Proxmox-Cluster, ermöglicht Snapshots und Shared-Access

Hochverfügbarkeit wird durch Proxmox-HA/Pacemaker für die Storage-VM sichergestellt. Bei Ausfall eines Hosts erfolgt automatischer Failover der VM.



Architektur



Note: über ProLB oder
Affinityrules sicherstellen
das die VMs auf
unterschiedlicher HW laufen.
STONITH nicht vergessen
(fence_pve)

Inside Pacemaker

Full List of Resources:

```
* fence_storage1    (stonith:fence_pve):      Started storage2
* fence_storage2    (stonith:fence_pve):      Started storage1
* Resource Group: nfs_group:
  * nfs_fs        (ocf:heartbeat:Filesystem):  Started storage2
  * nfs_server     (ocf:heartbeat:nfsserver):   Started storage2
  * nfs_export     (ocf:heartbeat:exportfs):    Started storage2
  * IP-nfs         (ocf:heartbeat:IPaddr2):    Started storage2
```

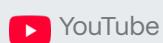
```
Waiting for the cluster to apply configuration changes...
location constraint created to move resource 'nfs_group' has been removed
Waiting for the cluster to apply configuration changes...
resource 'nfs_group' is moving on node 'storage1'
root@storage1:~#
```



```
Cluster Summary:
* Stacks: corosync (Pacemaker is running)
* Corosync Version: 3.0.10 (2022-05-25.0)
* Last updated: Fri Aug 15 13:44:22 2025 via storage
* Last change: Fri Aug 15 13:44:20 2025 by root via root on storage
* 2 nodes configured
* 6 resource instances configured

Node List:
v Online: [ storage storage2 ]
```

```
Aug 15 11:44:18 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<3 timer=724.8 us
Aug 15 11:44:18 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<3 timer=674.9 us
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<3 timer=765.1 us
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<4 timer=764.8 us
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<6 timer=724.8 us
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<6 timer=724.8 us
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<8 timer=731.7 us
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<8 timer=731.7 us
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<10 timer=644.6 us
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<10 timer=644.6 us
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<11 timer=3.38 s (fast)
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<11 timer=3.38 s (fast)
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<13 timer=3.38 s (fast)
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<13 timer=3.38 s (fast)
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<15 timer=3.38 s (fast)
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<15 timer=3.38 s (fast)
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<17 timer=3.38 s (fast)
Aug 15 11:44:17 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<17 timer=3.38 s (fast)
Aug 15 11:44:20 4 KIB ccc . (exit) /dev/nodal 29.7 0(0); request<18 timer=1.49 ms (fast)
```

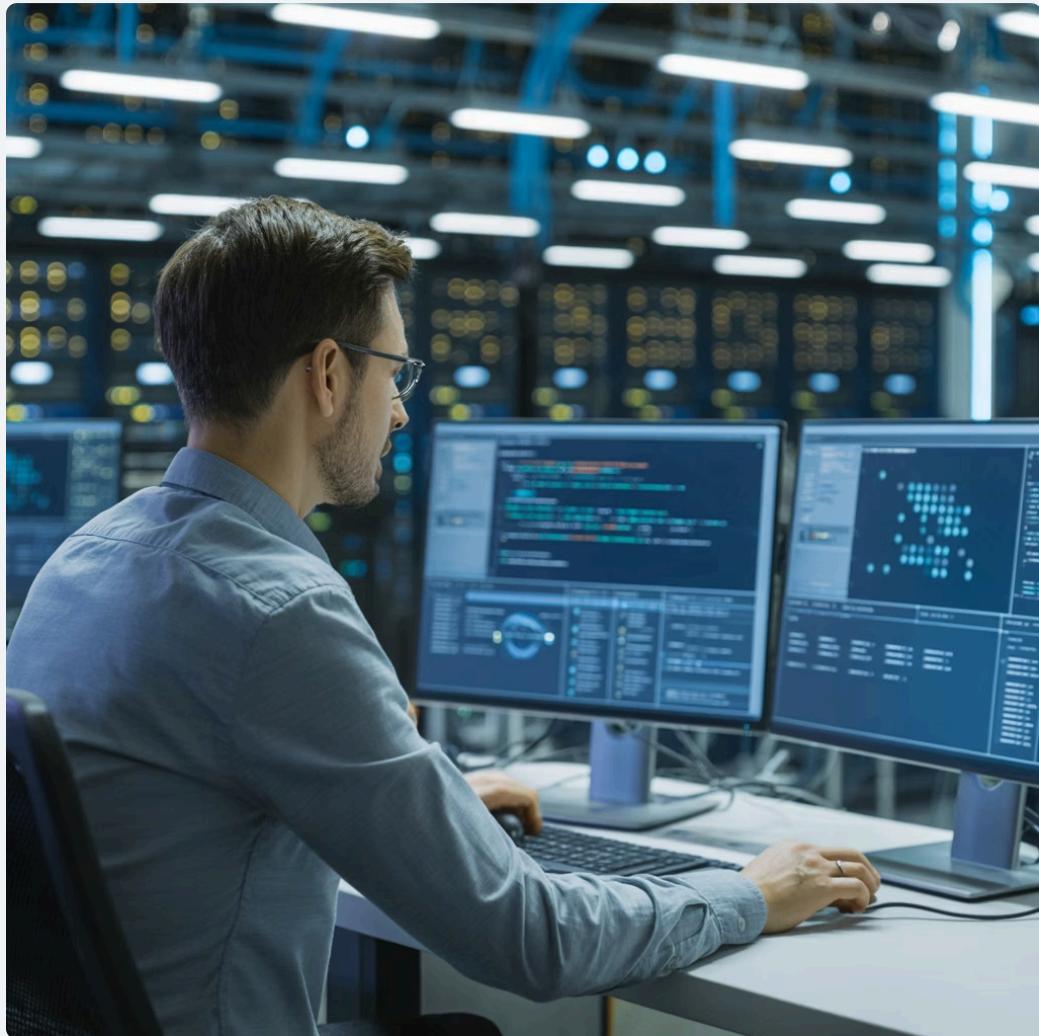


Umschaltung NFS & Proxmox



▶ 00:11

HA-Storage-VM – Vorteile



Bewährte Technologien

Basiert auf etablierten Standards: iSCSI, NFS, Pacemaker

Flexibel

Funktioniert mit jedem SAN-System, das iSCSI, FC o.ä. unterstützt

Snapshot-Funktionalität

Ermöglicht Snapshots und Linked Clones über QCOW2

Flexibilität im Gast

Storage-VM kann zusätzlich ZFS, SMB oder eigene iSCSI-Targets bereitstellen



HA-Storage-VM – Nachteile

Zusätzliche Komplexität

Die Storage-VM stellt einen Single Point of Failure ohne HA-Konfiguration dar. Die HA-Konfiguration erhöht die Komplexität des Gesamtsystems erheblich.

Latenz durch NFS-Layer

Der zusätzliche Abstraktionslayer durch NFS verursacht eine höhere Latenz im Vergleich zu direktem Block-Storage-Zugriff.

Timeout-Tuning notwendig

Für zuverlässige Failover ist ein sorgfältiges Tuning der Timeout-Parameter erforderlich. Falsche Einstellungen können zu Split-Brain oder langen Ausfallzeiten führen.

Monitoring-Aufwand

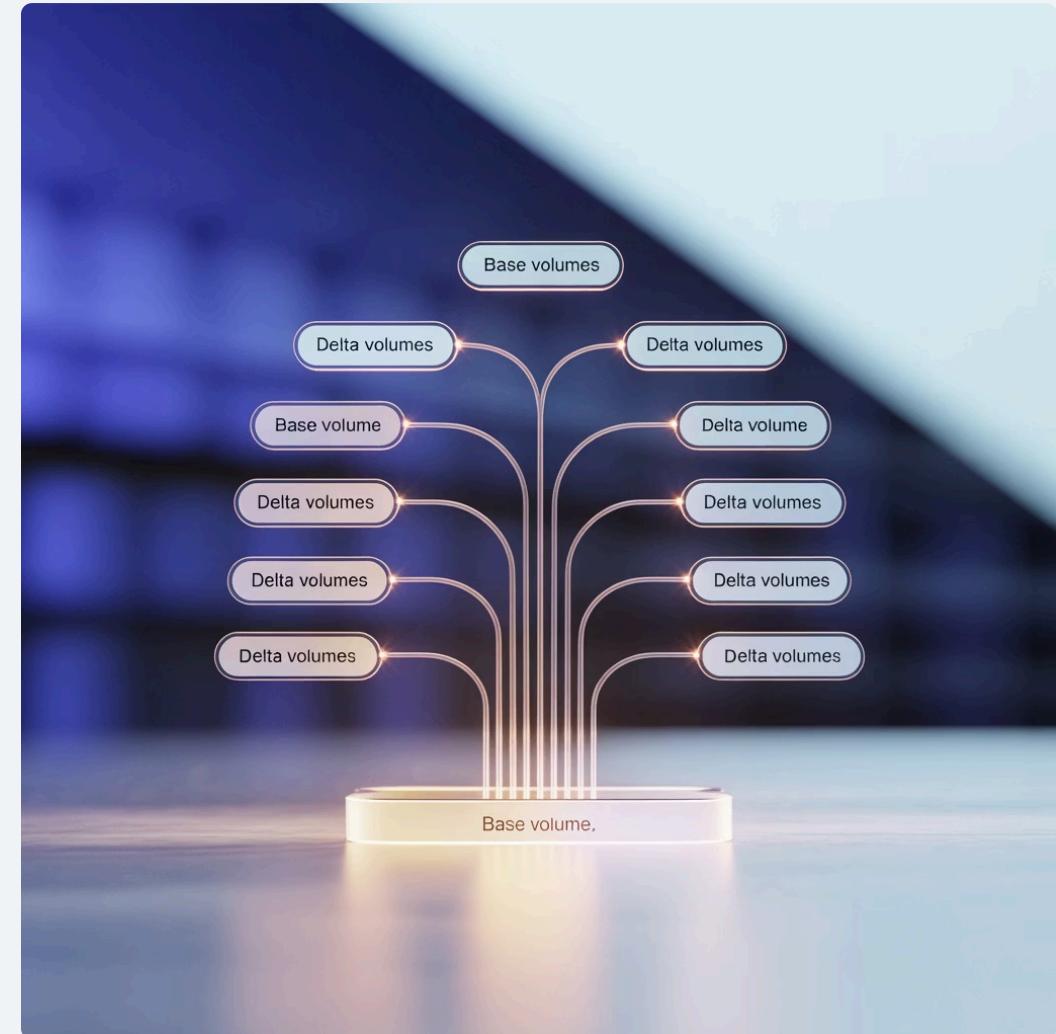
Die Überwachung und Verwaltung der Storage-VM erfordert zusätzliche Tools und Know-how außerhalb der Proxmox-Oberfläche.

Proxmox 9: Snapshots as Volume Chains

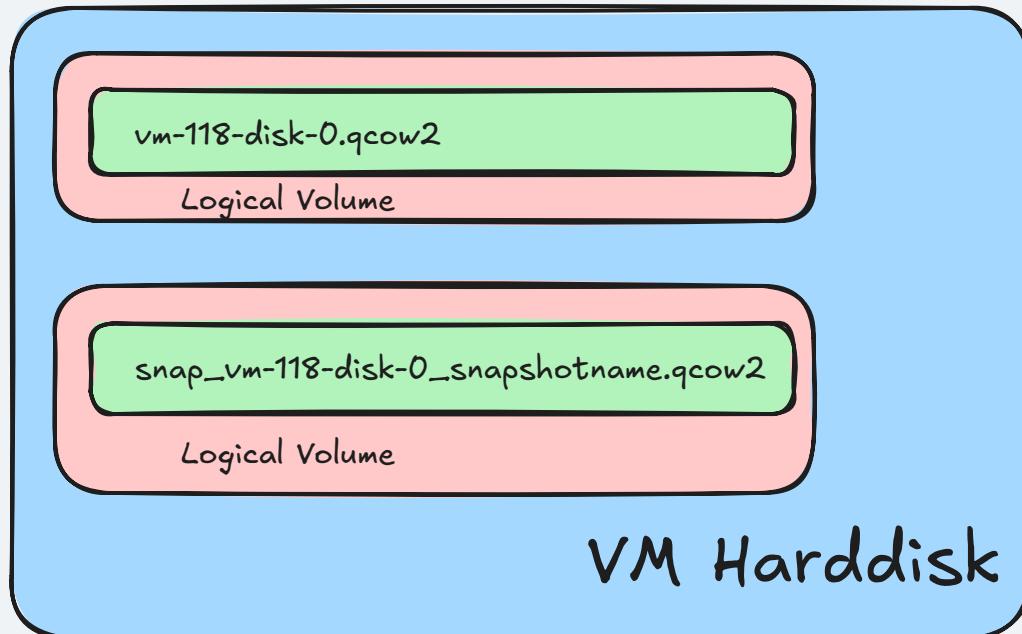
Mit Proxmox VE 9 wurde eine neue Methode für Snapshots auf Thick-Provisioned LVM eingeführt:

- Funktioniert auch auf shared iSCSI/FC-Storage
- Implementierung als Kette separater LVM-Volumes:
 - Base-Volume (Original)
 - Delta-Volumes pro Snapshot
- QEMU liest unveränderte Blöcke aus Parent-Volumes
- Vermeidet den Overhead klassischer LVM-Snapshots
- Aktivierung per Storage-Option: `snapshot-as-volume-chain`
- Grundsätzlich auch auf anderen Storagetechnologien möglich

Aktuell als Technology Preview – nicht für Produktivumgebungen
empfohlen!



Wie sieht das in der Praxis aus?

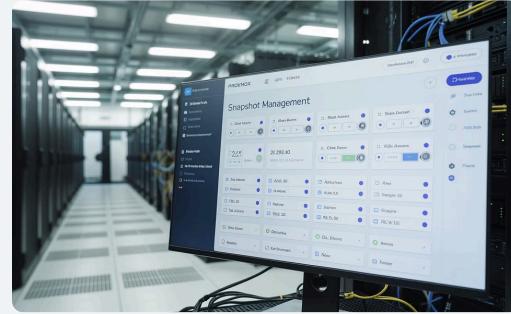


Volume Chains – Vorteile



Snapshots auf Shared-Storage

Ermöglicht Snapshots auf jedem Shared-Blockstorage ohne zusätzliche Software



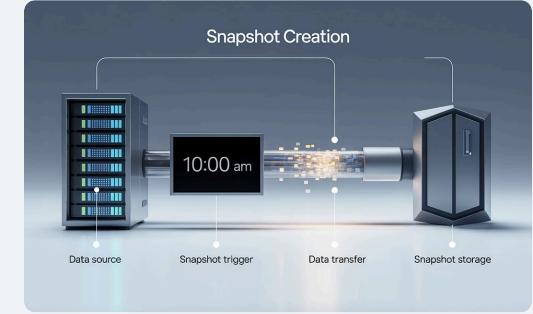
Proxmox-Integration

Vollständig in die Proxmox GUI und CLI integriert, einheitliche Verwaltung



Linked Clones

Ermöglicht platzsparende Linked Clones basierend auf Snapshots



Einfacher Workflow

Gewohnter Snapshot-Workflow wie bei ZFS/Ceph, keine Umstellung nötig

Hauptvorteil: Keine zusätzliche Storage-VM oder Middleware erforderlich – funktioniert direkt mit vorhandenem SAN

Volume Chains – Einschränkungen



Technology Preview Status

Diese Funktion ist aktuell als Technology Preview gekennzeichnet und wird von Proxmox noch nicht für den Produktiveinsatz empfohlen!

Thick Provisioning

Volumes sind immer Thick Provisioned – bei vielen Snapshots kann der Platzbedarf schnell steigen

Backup-Integration

Eingeschränkte Unterstützung für Backup/Restore-Operationen im Vergleich zu vollständig ausgereiften Systemen wie Ceph oder ZFS

Lese-Overhead

Bei langen Snapshot-Ketten kann ein Lesevorgriff auf mehrere Volumes notwendig sein, was die Performance beeinträchtigen kann

Live-Migration

Live-Migration mit lokalen Volume Chains nicht möglich – nur bei shared Storage funktionsfähig

Vergleich beider Ansätze

Kriterium	Volume Chains	HA-Storage-VM
Architektur	Einfach, direkt auf Block-Storage	Komplexer, mit zusätzlicher VM-Schicht
Snapshots	Native Integration in Proxmox	Über QCOW2/ZFS im Gast
Verfügbarkeit	Abhängig vom SAN, kein VM-SPOF	Failover mit Latenz, VM als SPOF
Performance	Gut (direkter Block-Zugriff)	Mäßig (zusätzliche NFS-Schicht)
Reife	Neu & experimentell	Ausgereift, etabliert
Flexibilität	Eingeschränkt auf Proxmox-Features	Erweiterbar mit VM-Funktionen

Fazit

Volume Chains repräsentieren die **Zukunft für Snapshots auf SAN-Storage in Proxmox**:

- Einfachere Architektur ohne zusätzliche Komponenten
- Native Integration in Proxmox-Oberfläche
- Direkte Nutzung vorhandener SAN-Infrastruktur

Die HA-Storage-VM bleibt sinnvoll für:

- Ältere Proxmox-Versionen (< 9.0)
- Spezielle Anforderungen (z.B. Multi-Protokoll-Storage)
- Produktivumgebungen, bis Volume Chains stabiler werden

Langfristig ist eine verbesserte Integration und Stabilität der Volume Chains zu erwarten – die Zukunft der Snapshot-Funktionalität auf Block-Storage in Proxmox.

Evaluation

Volume Chains in Testumgebungen evaluieren

Feedback

Erfahrungen mit der Proxmox-Community teilen

Migration

Nach Stabilisierung schrittweise Migration planen

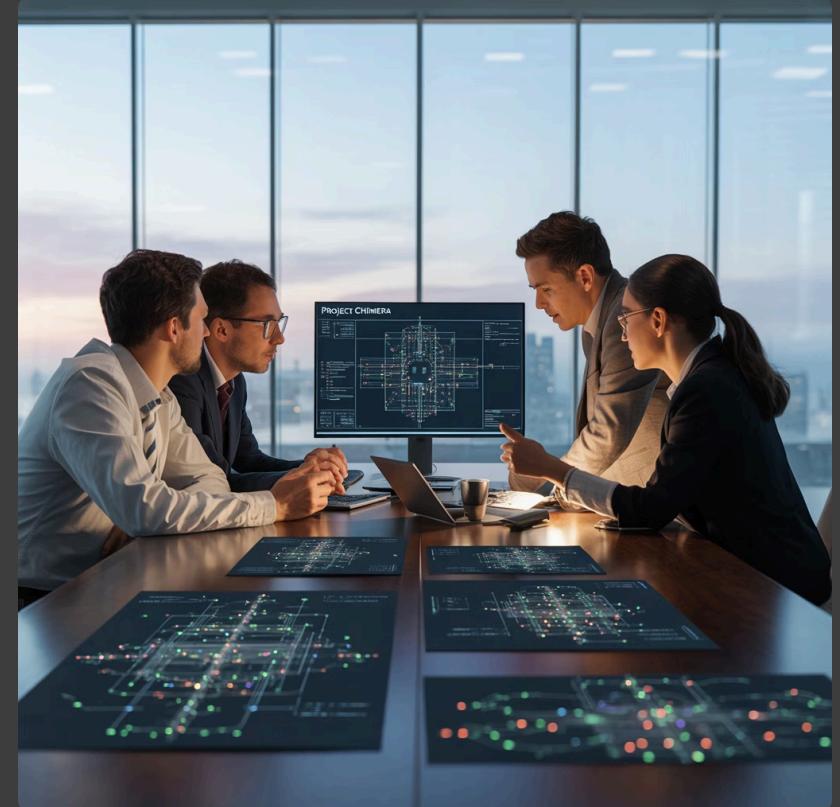
Fragen & Diskussion

Diskussionsthemen:

- Welche Erfahrungen habt ihr mit beiden Ansätzen gemacht?
- Gibt es Möglichkeiten für Kombinationen oder Übergangsszenarien?
- Wie steht es mit der Performance in euren Umgebungen?
- Welche Features fehlen euch noch bei Volume Chains?

Euer Feedback zu Volume Chains ist wertvoll für die Proxmox-Community! Teilt eure Erfahrungen im Forum oder auf GitHub.

Kontakt: forum.proxmox.com | github.com/proxmox



Vielen Dank!



Fragen und Feedback sind immer Willkommen

