

# ZFS Autosnapshot auf zvol und Datasets mounten

Du möchtest dich gerne für unsere Hilfe erkenntlich zeigen 🙏 . Gerne. Wir bedanken uns bei dir für

deine Spende! 🙌



Hauseigenes Apt-Repo: <https://apt.iteas.at>



GITLAB Enterprise:

- Automatische Snapshots von VM's und CT's in definierten Zeiten anfertigen lassen, z.B. alle 10 Minuten
- Rollback einzelner virtuellen Festplatten, z.B. nur Laufwerk C:\ bei Windows
- Direktes recovern von Dateien in VM-Images via cp oder Filezilla
- Klonen von bestimmten Snapshotständen um z.B. Tests durchzuführen
- Verschlüsselungstrojaner haben keine Chance
- Proxmoxsnapshots und Replicationen sind weiterhin nutzbar

## Installation und Konfiguration

Annahme hier ist ein Proxmox 7.3 System mit Kernel 5.19. Für Installation benutzen wir wie immer „apt“.

```
apt install zfs-auto-snapshot
```

Nach der Installation ist Autosnapshot bereits für alle Zvol's und Datasets aktiviert. Viel wichtiger ist die Frage ob ich die Autosnapshots überall haben möchte. Ich mach das genau umgekehrt. Ich deaktiviere global und aktiviere dann einzelne Disks wo ich das für sinnvoll halte.

Um das ganze manuell unter Kontrolle zu haben musst gleich nach der Installation die Autosnapshotfunktion für alle Pool's deaktivieren. Damit kannst du danach einzelnen Disks aktivieren.

```
zfs set com.sun:auto-snapshot=false rpool
zfs set com.sun:auto-snapshot=false Poolname-was-auch-immer
```

rpool	com.sun:auto-snapshot	false	local
rpool/ROOT	com.sun:auto-snapshot	false	inherited from rpool
rpool/ROOT/pve-1	com.sun:auto-snapshot	false	inherited from rpool
rpool/data	com.sun:auto-snapshot	false	inherited from rpool

Enzellene Disks aktivierst du so:

```
zfs set com.sun:auto-snapshot=true pool/data/vm-110-disk-0
```

Umgekehrt gehts auch mit `zfs inherit`

Welche virtuellen HDD's nun für Autosnapshot aktiviert sind, siehst du mit

```
zfs get com.sun:auto-snapshot | grep true
```

Zeiten für die Snapshots kannst du hier konfigurieren:

```
/etc/cron.d/zfs-auto-snapshot # hier stelle ich gerne 10 Minuten ein  
/etc/cron.daily/zfs-auto-snapshot # nach blieben anpassen  
/etc/cron.hourly/zfs-auto-snapshot # nach blieben anpassen  
/etc/cron.monthly/zfs-auto-snapshot # nach blieben anpassen  
/etc/cron.weekly/zfs-auto-snapshot # nach blieben anpassen
```

Die Files sind selbsterklärend.

**Achtung: Snapshot Zeit ist immer UTC damit man bei Zeitumstellungen nicht durcheinander kommt.**

**Achtung2: Du kannst weiterhin Snapshots in Proxmox nutzen. Vor der Nutzung Autosnapshot global deaktivieren, weil du bei einem Rollback des Proxmoxsnapshots alle in der Zwischenzeit getätigte Snaps von Autosnapshot manuell auf der CMD löschen musst.**

Autosnapshot deaktivieren:

```
chmod -x /usr/sbin/zfs-auto-snapshot
```

Autosnapshot wieder aktivieren:

```
chmod +x /usr/sbin/zfs-auto-snapshot
```

## Snapshots mounten (vdev)

Natürlich möchtest du auch wieder Daten aus den Snapshots herausholen. Hierfür muss Snapshots als DEV sichtbar machen. Du kannst das auf ganze Pools oder auf einzelne Disks anwenden.

```
zfs set snapdev=visible pool/data/vm-110-disk-0
```

Danach kannst du das DEV read only mounten. z.B.

```
mount -o noexec,ro /dev/zvol/rpool/data/vm-110-disk-0@zfs-auto-snap-2022-12-31-0638-part1 /mnt/snapshots #Linux  
mount -o ro /dev/zvol/rpool/data/vm-110-disk-0@zfs-auto-snap-2022-12-31-0638-part1 /mnt/snapshots #Windows
```

## Direktes Recovery von VDEV's oder einzelnen Dateien

In beiden Fällen muss die virtuelle Maschine herunter gefahren werden um eine Beschädigung des Filesystems zu vermeiden. Um eine virtuelle HDD direkt zurück zu spielen, machst du einfach ein Rollback.

```
zfs rollback rpool/data/vm-115-disk-2@zfs-auto-snap-2023-01-23-1130 -r
```

Dabei müssen auch alle neueren Snapshots gelöscht werden. Sollte das nicht gewollt sein, erstelle einen Klon. z.B.

```
zfs clone rpool/data/vm-115-disk-2@zfs-auto-snap-2023-01-23-1130  
rpool/data/vm-999-disk-2
```

Nach dem Rollback kannst deine Maschine wieder starten. Möchtest du nur ein paar Daten aus einem Snapshot herausholen, musst du die VDEVs mounten. Erster Mountbefehl (siehe oben). Nun möchtest du ja die Daten auch direkt ohne Umwege zurückspielen. Deshalb mounten wir auch das aktuelle LiveVDEV der VM dazu. Hierzu muss die VM herunter gefahren werden.

```
mount /dev/zvol/rpool/data/vm-110-disk-0 /mnt/snaplive
```

Daten kannst du dann ganz einfach mit z.B. cp von A nach B übertragen. Danach zuerst das Snaplive und dann Snapshot unmounten. VM starten, fertig.

```
umount /mnt/snaplive  
umount /mnt/snapshot
```

## Daten eine laufende VM recovern

Auch das ist möglich. Dann eben über das Netzwerk. In Linux kannst du hierfür SSH verwenden, in Windows verwendest du Filezilla. Sollten die Proxmoxserver so eingestellt sein das eine Auth nur mittels SSH-Key möglich ist, muss die Passwortauth vor dem Kopiervorgang temporär freigeschalten werden.

### sshd\_config

```
...  
# To disable tunneled clear text passwords, change to no here!  
- PasswordAuthentication no  
+ PasswordAuthentication yes  
#PermitEmptyPasswords no  
...
```

Danach den SSH-Server neustarten. Nach dem Kopieren sind die Einstellungen wieder zurück zu setzten. Möglicherweise macht das auch Puppet für dich 😊

## Snapshots mounten (datasets)

Bei Datasets ist es einfacher, kann mit z.B.

```
mount -t zfs rpool/data/home@rep_home_2022-07-05_00:36:48 /mnt/snapshots
```

eingebunden werden.

Um das ganze automatisch und elegant mit ZFS zu gestalten machten den Ordner `.zfs` in Datasets sichtbar. Z.B.

```
zfs set snapdir=visible rpool/data/home
```

Danach wird das gewünschte Snapshot automatisch beim Zugriff auf `.zfs/snapshot/rep_home_2022-07-16_00:01:25` gemountet.

## Links

[Linux ZFS \(weiterführende Dokumentation\)](#)

From: <https://deepdoc.at/dokuwiki/> - DEEPDOC.AT - enjoy your brain

Permanent link: [https://deepdoc.at/dokuwiki/doku.php?id=server\\_und\\_serverdienste:linux\\_zfs\\_autosnapshots\\_auf\\_zvol\\_und\\_datasets\\_mounten](https://deepdoc.at/dokuwiki/doku.php?id=server_und_serverdienste:linux_zfs_autosnapshots_auf_zvol_und_datasets_mounten)

Last update: 2023/01/31 10:54

