

cs-freeze4snap

Handbuch

Konsistente ZFS-Snapshots für Proxmox VM/LXC-Gäste und ESXi-VMs -
einfrieren, wenn möglich, Snapshot in jedem Fall

Version 1.3.0 · 13. Oktober 2026

Inhalt

1. Warum einfrieren	3
2. Optionen	3
3. Funktionsweise	4
4. Einfrier-Ketten	5
5. Proxmox	6
5.1 Der memory-Schritt	. 6
5.2 /etc/pve muss Teil des Snapshots sein	. 6
5.3 Wiederherstellung (Proxmox)	. 7
6. ESXi	7
6.1 Wie die VMs gefunden werden	. 7
6.2 Verbindung: --cfg	. 7
6.3 Einfrieren und Auftauen als getrennte Schritte	. 8
6.4 Wiederherstellung (ESXi)	. 9
7. Nutzung aus napp-it CS Jobs	9
8. Howto Einrichtung (napp-it 4ai)	9
9. Ergebnis-Format	10
10. Voraussetzungen, Tests, Lizenz	10

1. Warum einfrieren

ZFS-Snapshots können einen Pool nie beschädigen (atomare txg-Commits) - aber ein Snapshot, der während eines Schreibvorgangs im Gast entsteht, entspricht aus Sicht des Gastes dem Ziehen des Netzsteckers: Was in diesem Moment auf der Platte stand, sieht der Gast beim nächsten Start.

cs-freeze4snap hält die Gäste (oder nur deren Schreibzugriffe) unmittelbar vor dem Snapshot an, sodass der erfasste Stand ein sauberer Haltepunkt ist statt eines beliebigen Moments mitten im Schreiben. Es **blockiert den Snapshot nie**: Ein Gast, der sich nicht einfrieren lässt, wird so wie er ist gesichert und in **warnings** gemeldet.

Es gehört zur napp-it 4ai (Client-Server) Cluster-Werkzeugfamilie. napp-it CS installiert und aktualisiert es pro Member über das Menü **About > Download cs-tools**. Der ESXi-Modus braucht v1.3.0 oder neuer (napp-it CS warnt, wenn es einen älteren Build findet).

Kurzes Beispiel:

```
cs-freeze4snap snap --dataset rpool/data --name 20260809_1530

{"dataset":"rpool/data","snapshot":"20260809_1530","status":"ok","freeze_ms":6,
 "guests":[
  {"vmid":100,"type":"vm","platform":"proxmox-qemu","frozen":true,"strategy":"qga","thawed":true},
  {"vmid":101,"type":"vm","platform":"proxmox-qemu","frozen":true,"strategy":"qmp-
    pause","thawed":true},
  {"vmid":103,"type":"lxc","platform":"proxmox-
    lxc","frozen":true,"strategy":"cgroup","thawed":true}
 ]}
```

2. Optionen

```
cs-freeze4snap snap --dataset <ds> --name <snapname> [optionen]
cs-freeze4snap snap --hypervisor esxi --cfg <datei> --dataset <ds> --name <snapname> [optionen]
cs-freeze4snap discover|freeze|thaw|cleanup --cfg <datei> ... (ESXi)
```

Option	Bedeutung
<code>--dataset string</code>	zu sichernde ZFS-Dataset (Pflicht), z. B. <code>rpool/data</code>
<code>--name string</code>	Snapshot-Name (Pflicht), z. B. <code>20260809_1530</code>
<code>--recursive</code>	<code>-r</code> an <code>zfs snapshot</code> weitergeben (Standard true)
<code>--exclude string</code>	kommagetrennte VMIDs, die nicht eingefroren werden
<code>--include-only string</code>	kommagetrennte VMIDs; ersetzt die Erkennung vollständig
<code>--timeout duration</code>	maximale Zeit je Einfrier-Schritt eines Gastes (30 s, ESXi 120 s; Timeouts der Kette haben Vorrang)
<code>--forcefreeze</code>	OHNE Snapshot abbrechen, wenn ein Gast nicht eingefroren werden konnte
<code>--policy 'kette'</code>	Einfrier-Kette, wiederholbar (siehe Kapitel 4), oder <code>memkeep=N</code>
<code>--pre-snap-cmd 'cmd'</code>	nach dem Einfrieren unmittelbar vor dem ZFS-Snapshot ausführen (ein Fehler ist nur eine Warnung)

ESXi-Modus (das Tool kann auf jedem Rechner laufen, der den ESXi-Host erreicht):

Option	Bedeutung		
<code>--hypervisor esxi</code>	ESXi-VMs per Hot-Snapshot sichern statt Proxmox-Gäste		
<code>--cfg datei</code>	Verbindungsdatei: Host-Tabelle oder key=value (siehe Kapitel 6)		

Option	Bedeutung		
`--proto auto\`	ssh\`	soap\`	Transport (Standard auto = soap, ssh falls soap nicht erreichbar)
--host, --user	ESXi-Host und Login (Standard-Benutzer root; das Passwort ist nie eine Kommandozeilen-Option)		
--storage nfs	VMs nach Storage auswählen (nur nfs)		
--nfs-pat h pfad	NFS-Export-Pfad, wie ESXi ihn eingebunden hat (Standard: Mountpoint von --dataset)		
--nfs-ser ver host	NFS-Server, wie ESXi ihn kennt (nötig, wenn der Pfad doppelt exportiert wird)		
`--vms all\`	id,name,...\`	welche VMs (Standard all; --snap ist ein Alias)	
`--mode freeze\`	mem\`	plain\`	eine Kette für jede VM: freeze,plain,zfs / memory,plain,zfs / plain,zfs
--allow-m ixed	auch VMs sichern, die Platten auf anderen Datastores haben		
--include -off	auch ausgeschaltete oder suspendierte VMs sichern		
--thaw-ti meout dur	maximale Zeit, um einen VM-Snapshot zu entfernen (2 m)		
--zfs-cmd 'befehl'	den ZFS-Snapshot entfernt erstellen, z. B. <code>ssh nas zfs snapshot -r {fullname}</code>		
--state datei	freeze/thaw: Datei, die die VM-Snapshots trägt		

Unterbefehle (ESXi): `discover` listet die NFS-Datstores und VMs auf, die gesichert würden (JSON), und ändert nichts; `freeze` erstellt die VM-Snapshots, schreibt `--state` und lässt sie bestehen; `thaw` entfernt die in `--state` aufgeführten VM-Snapshots; `cleanup` entfernt übrig gebliebene VM-Snapshots dieses Tools (Tag `cs-freeze4snap`) auf dem NFS.

Die Gast-Erkennung ist automatisch: `zfs list -r <dataset>` plus die Proxmox-Namenskonvention (`vm-<id>-disk-*`, `subvol-<id>-disk-*`) - es gibt keine VMID-Liste zu pflegen. `--exclude` und `--include-only` decken den Ausnahmefall ab. Das Ergebnis ist ein JSON-Objekt auf stdout, das der Aufrufer auswerten soll (siehe Kapitel 9).

3. Funktionsweise

Alle Gäste der Dataset werden erkannt und **parallel** eingefroren, dann wird **ein** `zfs snapshot -r` erstellt (der alle im selben Augenblick erfasst), danach wird alles parallel wieder aufgetaut. Der langsamste Gast bestimmt die Dauer der Einfrierphase, nicht die Summe aller Gäste - Gäste mit gemeinsamem Zustand stehen also ungefähr im selben Moment still.

Gäste der Dataset erkennen

|
v

VM-A einfrieren --+

VM-B einfrieren ---+> auf alle warten --> EIN zfs snapshot -r --> alle auftauen

4. Einfrier-Ketten

Was "Einfrieren" eines Gastes bedeutet, ist eine **Kette** von Schritten, die von links nach rechts probiert werden - der erste funktionierende Schritt gewinnt, danach wird der ZFS-Snapshot erstellt.

Schritt	ESXi	Proxmox VM	Proxmox LXC
freeze (Alias quiesce)	VM-Snapshot, Gast-Dateisystem durch VMware Tools stillgelegt	QEMU Guest Agent fsfreeze	fsfreeze , sonst cgroup-Freeze
memory (Alias mem)	VM-Snapshot samt RAM (Hot-Snapshot, langsamer; eine Wiederherstellung setzt den laufenden Zustand fort)	qm snapshot --vmstate	-
plain	VM-Snapshot nur der Platten (crash-konsistent)	-	-
pause	-	QMP stop/cont (die ganze VM pausiert)	-
zfs	Einfrieren aufgeben, den ZFS-Snapshot so nehmen, wie es ist	gleich	gleich

Schritte, die es auf einer Plattform nicht gibt, werden übersprungen; eine Kette kann also ESXi und Proxmox gleichzeitig bedienen.

```
[freeze,memory,zfs,30]          # globale Kette: freeze, sonst memory, sonst ZFS; 30 s je
    Schritt
192.168.2.48:*,freeze,plain,zfs  # jede VM dieses ESXi-Hosts
192.168.2.48:vm100,memory,zfs    # eine VM (vm100, ct100 oder 100)
192.168.2.203:vm101,freeze,memory # Proxmox-Member; kein zfs am Ende -> strikt
```

Vorrang (stärkster zuerst): **--mode**, **--policy**, eine Zeile **host:vmid**, eine Zeile **host:***, die globale Zeile [...], der eingebaute Standard (**freeze,memory,zfs** für ESXi und Proxmox-VMs, **freeze,zfs** für LXC). **--policy** ist wiederholbar und nimmt dieselben Zeilen ohne Host-Präfix. Eine Zahl ist ein Timeout in Sekunden (**memory:300** für einen Schritt, eine einzelne Zahl für jeden Schritt ohne eigenen Wert). Die angewandte Kette steht in `guests[].chain`.

freeze,memory,zfs liefert einen dateisystem-konsistenten Punkt, wenn die Gast-Tools funktionieren, und einen Laufzustand-Punkt (inklusive RAM), wenn nicht. **plain** und **pause** sind crash-konsistent und müssen ausdrücklich gewählt werden. Eine Kette nur aus **zfs** berührt den Gast nie (**zfs-only**).

Best Effort oder strikt. Eine Kette, die mit **zfs** endet, ist Best Effort: Schlagen alle Schritte fehl, wird der Gast als nicht eingefroren gemeldet und der ZFS-Snapshot trotzdem erstellt. Eine Kette **ohne zfs** ist für ihre Gäste strikt: Funktioniert kein Schritt, bricht der Lauf vor dem ZFS-Snapshot ab (Exit 1, **status: error**) - wie **--forcefreeze**, aber je Gast. Eine nicht laufende Proxmox-VM gilt als konsistent (**stopped**, kein Snapshot) und stört eine strikte Kette nicht.

Timeouts. Ein Schritt, dem die Zeit ausgeht, gilt als fehlgeschlagen, und der nächste beginnt. Bei soap wird zuerst der vSphere-Task abgebrochen; bei ssh lässt sich ein laufendes **vim-cmd** von außen nicht abbrechen (ein Snapshot, den der Host eventuell noch erstellt, wird entfernt) - wer sich auf Schritt-Timeouts verlässt, nimmt besser soap. Bei Proxmox begrenzt der Timeout jeden Befehl eines Schritts.

Nie ein harter Fehler, außer eine Kette ist strikt oder **--forcefreeze** ist gesetzt: Ein Gast, der sich nicht einfrieren lässt, bekommt **frozen: false** und eine Warnung, ein nicht erreichbarer ESXi-Host oder ein falsches Passwort wird genauso gemeldet, und der ZFS-Snapshot wird in jedem Fall erstellt.

5. Proxmox

Das Tool läuft **auf dem Proxmox-Host** (nicht in einem Gast): Es ruft `zfs` und `qm/pct` auf und spricht die QGA/QMP-Sockets an, die nur dort existieren.

- Gäste werden je Dataset anhand der Namenskonvention gefunden - keine VMID-Liste.
- Eine VM mit Platten auf mehr als einem Pool wird in `warnings` gemeldet: `zfs snapshot -r` überschreitet nie Pool-Grenzen, im Snapshot sind also nur die Platten unter der gewählten Dataset (`vmid 101 (vm) has disk(s) outside rpool/data that will NOT be included in this snapshot`).
- `freeze` braucht den `qemu-guest-agent` im Gast. Ohne ihn fällt die Kette auf `memory` zurück (VM-Snapshot mit RAM) oder, wenn so gewünscht, auf `pause` (QMP `stop/cont`).
- LXC: `fsfreeze` wird auf dem Host am Mountpoint des Containers angewandt (das funktioniert auch bei unprivilegierten Containern). Unterstützt das Dateisystem `FIFREEZE` nicht - OpenZFS tut das oft nicht, prüfen mit `fsfreeze -f <mountpoint> -`, synchronisiert das Tool und friert den Container per `cgroup v2` ein.

5.1 Der memory-Schritt

Ohne Gast-Agent fällt die Standardkette `freeze,memory,zfs` auf `memory` zurück. Das Tool führt aus `qm snapshot <vmid> cs4s_<snapshot> --vmstate 1 --description "cs-freeze4snap <snapshot>"`

was keine Gast-Tools braucht: `qm` hält die VM kurz an (Sekunden, abhängig von der RAM-Größe), schreibt das RAM in ein Volume `vm-<id>-state-...` und sichert die Platten im selben Punkt. Eine Wiederherstellung liefert eine laufende VM in genau diesem Zustand.

- Der VM-Snapshot wird nach dem ZFS-Snapshot **nicht** entfernt: Auf ZFS besteht der konsistente Stand aus dem Paar `vm-<id>-disk-N@cs4s_<snapshot>` plus dem `vmstate`-Volume. Der rekursive `pool@<snapshot>` entsteht einen Moment später und enthält Platten, die etwas neuer sind als der RAM-Stand.
- Ein solcher VM-Snapshot lebt so lange wie der in seiner Beschreibung genannte ZFS-Snapshot: Jeder Lauf entfernt die `cs4s_`-Snapshots einer VM, deren ZFS-Snapshot nicht mehr existiert (10 Minuten Schonfrist für parallele Jobs). Angefasst werden nur Snapshots mit Namen `cs4s_*` und der Beschreibung `cs-freeze4snap ...`, und ein Snapshot, den eine laufende Wiederherstellung noch braucht, bleibt erhalten.
- `memkeep=N` (globale Zeile, `host:memkeep=N` oder `--policy 'memkeep=N'`) begrenzt die Memory-Snapshots je VM (0 = keine Grenze). Proxmox reserviert das `vmstate`-Volume `thick` (etwa 2,2 x RAM); bei RAM-lastigen VMs mit langer Aufbewahrung also zum Beispiel `memkeep=3` setzen.
- Ein fehlgeschlagener oder per Timeout abgebrochener `qm snapshot` wird wieder entfernt; der Standard-Timeout des Schritts ist 120 s.
- Das Ergebnis einer solchen VM hat `strategy: "qm-mem"`, `frozen: true` und eine Warnung, die einen zuvor fehlgeschlagenen VM-Snapshot nennt.

5.2 /etc/pve muss Teil des Snapshots sein

Der Memory-Snapshot ist ein Abschnitt von `/etc/pve/qemu-server/<id>.conf`, und `qm rollback` braucht ihn nach einer Wiederherstellung - die Konfiguration muss also mit dem Snapshot reisen. napp-it CS fügt `/etc/pve` bei `freeze=proxmox` automatisch zu `include=` hinzu und synchronisiert es unmittelbar vor dem ZFS-Snapshot erneut. Beim eigenen Aufruf des Tools:

```
cs-freeze4snap snap --dataset tank/vm --name s1 \
  --pre-snap-cmd 'rsync -a --delete /etc/pve/ /tank/vm/_include/_etc_pve'
```

Dieselbe Kombination lohnt sich auch für normale Snapshot-Jobs: napp-it CS `include=<ordner>` synchronisiert normale Ordner vor dem Snapshot nach `<dataset>/_include`, sodass ein **rekursiver** Snapshot Gastkonfiguration und eingefrorene Plattendaten gemeinsam enthält:

1. include -> /etc/pve wird nach <dataset>/_include/_etc_pve synchronisiert
2. freeze -> die Kette pausiert die Gäste
3. snapshot -> ein atomarer rekursiver zfs snapshot -r erfasst beides

5.3 Wiederherstellung (Proxmox)

VM gestoppt, Platten und Konfiguration aus dem ZFS-Snapshot (oder vom Replikationsziel) zurück:

```
zfs rollback -r tank/vm/vm-100-disk-0@cs4s_<snapshot>    # jede Platte der VM
qm rollback 100 cs4s_<snapshot>                        # RAM-Stand, die VM startet laufend
```

Zuerst `zfs rollback -r` ist Pflicht: Proxmox verweigert `qm rollback`, solange auf der Platte ein neuerer ZFS-Snapshot existiert. Ist die Konfiguration weg (neuer Host), wird sie zuerst aus `<dataset>/_include/_etc_pve/qemu-server/<id>.conf` zurückkopiert.

6. ESXi

cs-freeze4snap kann einen **NFS-Export, der als ESXi-Datastore dient**, VM-konsistent sichern (v1.3.0 oder neuer): Es findet die VMs auf diesem NFS selbst, erstellt von jeder eingeschalteten VM einen VM-Snapshot (das "Einfrieren"), erstellt den ZFS-Snapshot und entfernt die VM-Snapshots wieder (das "Auftauen"). Der ZFS-Snapshot enthält dann jede VM mit eingebettetem ESXi-Snapshot: Nach einer Wiederherstellung setzt man die VM auf diesen Snapshot zurück und erhält ein dateisystem-konsistentes (`freeze`) oder Laufzustand-Abbild (`mem`) statt eines Crash-Abbilds.

```
cs-freeze4snap snap --hypervisor esxi --cfg /pfad/esxi.cfg \
  --dataset tank/nfs --name auto_20260919_2100 --storage nfs --vms all
```

Der ESXi-Host wird über das Netzwerk erreicht (ssh oder soap), das Tool muss also **nicht** auf dem ZFS-Server laufen - jeder Rechner, der den ESXi-Host erreicht, genügt (ein Frontend, ein Sprung-Host). Den ZFS-Snapshot erstellt das lokale `zfs` oder auf einem anderen Rechner `--zfs-cmd`:

```
cs-freeze4snap snap --hypervisor esxi --cfg esxi.cfg --dataset tank/nfs --name s1 \
  --zfs-cmd 'ssh root@nas zfs snapshot -r {fullname}'
```

{dataset}, {snapshot} und {fullname} werden ersetzt; die Namen werden vorher gegen [A-Za-z0-9_./-] geprüft. Der Befehl läuft über `sh -c (cmd /C unter Windows)`.

6.1 Wie die VMs gefunden werden

1. Der NFS-Export-Pfad stammt aus `--nfs-path` oder vom Mountpoint von `--dataset` (ohne lokales zfs wird `/<dataset>` angenommen und gewarnt).
2. Ausgewählt werden die NFS-Datstores des Hosts, deren Export passt, plus die Exporte von Child-Datasets bei `--recursive` (Standard). Wird derselbe Pfad von mehr als einem NFS-Server exportiert, gibt man ihn mit `--nfs-server` an.
3. Jede VM mit Dateien auf diesen Datastores ist ein Kandidat; gesichert werden die eingeschalteten VMs (`--vms all` oder eine kommasetrennte Liste von IDs und Namen). Übersprungen, mit dem Grund in `skipped`: ausgeschaltete oder suspendierte VMs (sie sind ohnehin konsistent; `--include-off` sichert sie auch), VMs, die auch Platten auf einem anderen Datastore haben (ein Snapshot dieses NFS enthielte nur einen Teil von ihnen; `--allow-mixed` sichert sie trotzdem, mit Warnung) und VMs, die nicht in der `--vms`-Liste stehen.

`discover` zeigt genau das, ohne etwas zu ändern:

```
cs-freeze4snap discover --cfg esxi.cfg --nfs-path /tank/nfs
```

6.2 Verbindung: --cfg

Host-Tabelle - eine Datei für alle Server, ein Server pro Zeile, # für Kommentare:

```
# host,user,password
192.168.2.48,root,secret
192.168.2.49,root,pass,with,commas    # das Passwort ist alles nach dem 2. Komma
192.168.2.50,cert                    # ssh-Schlüssel des laufenden Benutzers (~/.ssh/id_ed25519,
  id_ecdsa, id_rsa), Benutzer root
192.168.2.51,ops,cert                # gleicher Schlüssel, anderer Benutzer
```

```
192.168.2.52,root,cert,/etc/keys/esx # expliziter OpenSSH-Privatschlüssel
```

--host <ip> wählt die Zeile (steht nur eine Zeile in der Datei, kann es entfallen). cert-Einträge funktionieren nur mit ssh (OpenSSH-Schlüssel, keine PuTTY-.ppk); soap braucht ein Passwort. Optionen, die zu einem Server gehören, sind zusätzliche Zeilen <host>:<option>=<wert> in derselben Datei: proto, port, hostkey, tls_sha256, timeout, useragent, zum Beispiel 192.168.2.48:proto=ssh oder 192.168.2.48:hostkey=SHA256:abcd....

key=value-Datei für einen Server - wenn ein Server mehr als Anmeldedaten braucht:

```
host=192.168.2.48
user=root
password=...           # oder CS_ESXI_PASSWORD in der Umgebung - nie eine Kommandozeilen-Option
proto=auto             # auto (soap, ssh falls soap nicht erreichbar), ssh (vim-cmd) oder soap
                        (Port 443)
hostkey=SHA256:...     # ssh: Host-Key festschreiben (der erste Lauf gibt ihn in "warnings" aus)
tls_sha256=...         # soap: Zertifikat festschreiben (sha256 hex)
key=/pfad/id_ed25519   # ssh: OpenSSH-Privatschlüssel statt Passwort
timeout=15             # Verbindungs-Timeout in Sekunden
```

Das Format wird an der ersten Eintragszeile erkannt (host,... ist eine Tabelle, key=... eine key=value-Datei). CS_ESXI_HOST, CS_ESXI_USER und CS_ESXI_PASSWORD überschreiben die Datei; --host, --user und --proto überschreiben beides. Die Datei sollte nur für den Job-Benutzer lesbar sein. Die Einfrier-Ketten können in derselben Datei stehen (napp-it CS: _cfg/freeze4snap/freeze4snap.cfg, beschrieben in freeze4snap.readme daneben).

	ssh	soap
kostenlose ESXi-Lizenz	funktioniert (vim-cmd)	funktioniert: die freie Lizenz akzeptiert Schreibaufrufe nur von einem Client, dessen User-Agent mit "VMware" beginnt - das Tool sendet VMware VI Client/4.0.0 (useragent= überschreibt das)
Geschwindigkeit (16 VMs, gemessen)	etwa 12 s zum Auflisten der VMs, etwa 3 s je Snapshot	etwa 0,2 s zum Auflisten der VMs, etwa 2 s je Snapshot
benötigt	ssh-Dienst auf dem Host	Port 443

--proto auto (Standard) versucht zuerst soap und fällt auf ssh zurück, wenn die vSphere-API nicht erreichbar ist (eine Warnung sagt das). Ein fehlgeschlagenes soap-Login ist endgültig und wird nicht über ssh wiederholt: Dieselben Zugangsdaten würden wieder scheitern, und jedes fehlgeschlagene Login zählt zur Kontosperrung des ESXi-Hosts (standardmäßig 5). Einträge mit cert nutzen direkt ssh. Die ESXi-Uhr wird nicht verwendet (das Tool benennt die Snapshots selbst), eine falsche Host-Uhr schadet also nicht.

6.3 Einfrieren und Auftauen als getrennte Schritte

Für Jobs, bei denen der Snapshot von etwas anderem erstellt wird (ein Replikationsschritt, ein Skript auf dem ZFS-Server):

```
cs-freeze4snap freeze --cfg esxi.cfg --dataset tank/nfs --name s1 --state /var/tmp/s1.json
zfs snapshot -r tank/nfs@s1 # auf beliebigem Weg
cs-freeze4snap thaw --cfg esxi.cfg --state /var/tmp/s1.json
```

freeze schreibt die Handles der VM-Snapshots in die Status-Datei (Modus 0600) und lässt die Snapshots bestehen; thaw entfernt sie und löscht die Status-Datei (bei einem nicht entfernbarer Snapshot bleibt die Datei erhalten, und der Exit-Code ist ungleich null; ein bereits verschwundener Snapshot gilt als entfernt). **VM-Snapshots nicht stehen lassen** - sie wachsen. Nach einem Absturz cleanup ausführen:

```
cs-freeze4snap cleanup --cfg esxi.cfg --nfs-path /tank/nfs
```

6.4 Wiederherstellung (ESXi)

Die VM-Dateien aus dem ZFS-Snapshot wiederherstellen (zum Beispiel den Snapshot klonen und den VM-Ordner registrieren oder kopieren, oder `zfs rollback`), dann die VM auf dem ESXi-Host auf den eingebetteten Snapshot zurücksetzen: `vim-cmd vmsvc/snapshot.revert <vmid> <snapid> 0`, oder in der Oberfläche Snapshots `> cs4s-<snapshot>-<vmid> > Restore`.

7. Nutzung aus napp-it CS Jobs

napp-it CS (Snapshot- und Replikations-Jobs) hat eine Job-Einstellung **Freeze VM prior snap = off** | `proxmox` | `esxi`. `proxmox` führt das Tool auf dem ZFS-Member (dem Proxmox-Host) aus; `esxi` führt es auf dem napp-it CS **Frontend** aus und erstellt den ZFS-Snapshot auf dem Member zwischen Einfrieren und Auftauen. ESXi-Hosts, Logins, Protokoll und Einfrier-Ketten kommen aus `_cfg/freeze4snap/freeze4snap.cfg` auf dem Frontend (`freeze4snap.readme` im selben Ordner beschreibt die Datei). Ein fehlgeschlagenes Einfrieren blockiert den Job nie, außer eine Kette ist strikt; ein fehlendes Tool oder eine fehlende cfg ist ein harter Fehler. Bei `proxmox` stellen die Jobs außerdem sicher, dass `/etc/pve` Teil des `include=` des Jobs ist (wird bei Fehlen ergänzt), und geben `--pre-snap-cmd` und `memkeep=` an das Tool weiter.

Die eventbasierte Echtzeit-ZFS-Replikation von `cs-sync` (Primary nach Backup) ruft `cs-freeze4snap` genauso für ihren täglichen applikationskonsistenten Snapshot auf.

8. Howto Einrichtung (napp-it 4ai)

- Das neueste napp-it 4ai laden (Menü **About > Frontend Update**).
- Einen Key für 3 Server besorgen - kostenlos für nicht-kommerzielle Nutzung und Heimanwender (Evaluate & Home use auf www.napp-it.org).
- Die beteiligten Server im Frontend hinzufügen (Server-Gruppe, dann den Server auswählen).

Proxmox: `cs-freeze4snap` muss **auf dem Proxmox-Host** selbst vorhanden sein (**About > Download cs-tools** verteilt es pro Member). `/etc/pve` wird automatisch Teil des Replikations-Snapshots: Bei `Freeze VM prior snap = proxmox` und einem rekursiven Job wird der Ordner zum Include des Jobs hinzugefügt, wenn er fehlt, und direkt nach dem Einfrieren neu synchronisiert, sodass die VM-Konfigurationen im selben Snapshot reisen wie die konsistenten Plattendaten. Anders als die Proxmox-Backup-Werkzeuge arbeitet der Job **je Dataset**: Die VMs, die die Dataset nutzen, werden automatisch gefunden, aber ihre Platten müssen vollständig darauf liegen (eine VM mit einer Platte auf einer anderen Dataset wird übersprungen und gemeldet).

ESXi: `cs-freeze4snap` muss auf dem **Frontend-Rechner** vorhanden sein (**About > Download cs-tools**). Die Serverdatei `/opt/cswb-gui/_cfg/freeze4snap/freeze4snap.cfg` anlegen (Windows-Frontend: `C:\opt\cswb-gui\_cfg\freeze4snap\freeze4snap.cfg`), zum Beispiel:

```
# ESXi-Server, einer pro Zeile: host,user,password oder host,cert[,keyfile]
192.168.1.48,root,1234
```

Replikations-Job: Einen Replikations-Job für das Dateisystem anlegen, das die VMs enthält: Quelle und Ziel sind frei (beliebig zu beliebig), und die Übertragung ist verschlüsselt (`cs-stream`, der Standard bei "Replication type"). `Freeze VM prior snap` auf **proxmox** (Tool auf dem Proxmox-Quell-Member) oder **esxi** (Tool auf diesem Frontend, friert die VMs auf dem Quell-NFS-Dateisystem ein) setzen. Die eingebaute Standardkette ist `freeze,memory,zfs`: zuerst das Einfrieren versuchen (VMware Tools, QEMU Guest Agent); ist das mangels Gast-Tools nicht möglich, den Hot-Snapshot samt RAM nehmen, dann in jedem Fall den ZFS-Snapshot - eine Kette, die auf `zfs` endet, blockiert den Job nie. Die Replikation mit beliebiger Keep/Hold-Aufbewahrung starten, zum Beispiel `hold 4s` (immer die letzten 4 Replikationen behalten; `hold 8` = 8 Tage halten) oder `keep hours:24,days:32,months:12,years:2` (bei stündlichen Snapshots bleiben die letzten 24, dann die letzten 32 Tage, 12 Monate und 2 Jahre).

9. Ergebnis-Format

```
{
  "dataset": "rpool/data",
  "snapshot": "20260809_1530",
  "status": "ok",                      // oder "error" - siehe Feld "error"
  "error": "...",                     // nur bei status "error"
  "note": "...",                      // z. B. "no guests discovered"
  "warnings": [...],                 // Platten über Pool-Grenzen, nicht fatale Prüffehler
  "freeze_ms": 6,                    // Echtzeit der Einfrierphase
  "guests": [
    {
      "vmid": 101,
      "type": "vm",                   // "vm" | "lxc"
      "platform": "proxmox-qemu",
      "frozen": true,
      "strategy": "qmp-pause",        // qga | qmp-pause | qm-mem | stopped | fsfreeze | cgroup | zfs-only
      "thawed": true,
      "warning": "...",               // nur wenn dieser Gast nicht sauber eingefroren wurde
    }
  ]
}
```

Der ESXi-Modus ergänzt `hypervisor` ("esxi"), `transport`, `nfs_path`, `datastores` (die passenden ESXi-Datstore-Namen), `skipped` (`vmid`, `name`, `reason`), `state` (freeze) und `removed` (cleanup); jeder Eintrag in `guests[]` hat `platform`: "esxi", `name`, `strategy` (`esxi-quiesce`, `esxi-mem`, `esxi-snap`) und `snap_id`. `CS_ESXI_DEBUG=1` gibt jeden ssh-Befehl und soap-Aufruf mit Dauer auf stderr aus.

Der Exit-Code ist 0 bei `status`: "ok", sonst ungleich null (auch wenn `--forcefreeze` den Job abbricht). Nur ein fehlschlagender `zfs snapshot` selbst (Rechte, Namenskollision, Pool voll, ...) erzeugt `status`: "error" - Probleme beim Einfrieren nie.

10. Voraussetzungen, Tests, Lizenz

- Der **ESXi-Modus** läuft auf jedem Rechner, der den ESXi-Host über ssh (Port 22) oder soap (Port 443) erreicht; den ZFS-Snapshot erstellt das lokale `zfs` oder `--zfs-cmd`. Statische Builds (`CGO_ENABLED=0`) gibt es für Linux (amd64, arm64), macOS (amd64, arm64), Windows, FreeBSD, illumos und Solaris (amd64).
- Der **Proxmox-Modus** läuft auf dem Proxmox-Host (nicht in einem Gast).
- Zum Bauen Go 1.21 oder neuer; keine Laufzeit-Abhängigkeiten außer golang.org/x/sys (für die `FIFREEZE`- und `FITHAW`-ioctls).

Getestet: Unit-Tests mit Prozess-internen Attrappen (ein falscher ssh-Server, ein falscher vSphere-SOAP-Endpunkt, der Schreibaufrufe ablehnt, wenn der User-Agent nicht mit "VMware" beginnt, falsche QGA und QMP), `go test ./... -race`. ESXi 8 mit freier Lizenz und NFS-Datstores eines ZFS-Servers: `discover`, `freeze/thaw`, `snap`, `cleanup` über ssh und soap, die Modi `freeze`, `mem` und `plain`, falsches Passwort und nicht erreichbarer Host (Best Effort und `--forcefreeze`); aus napp-it CS Replikations-Jobs (Windows-Frontend, OmniOS-ZFS-Server) brauchten zwei laufende VMs 9-12 s für Einfrieren, ZFS-Snapshot und Auftauen. Proxmox 9 mit laufenden und gestoppten QEMU-VMs und einem LXC-Container: `qga`, `qmp-pause`, `cgroup`, `stopped` und die Fehlerpfade, ein Gast mit Platten auf zwei Pools in `warnings` markiert, `--forcefreeze` bricht ohne Snapshot ab, die Kombination `include=/etc/pve` im Snapshot verifiziert. Nicht getestet: NFS-4.1-Datstores, ESXi 6.x und 7.x, Wiederherstellung eines Hot-Snapshots.

Hinweis: ZFS unterstützt auf dem getesteten Kernel und OpenZFS kein `FIFREEZE` (prüfen mit `fsfreeze -f <mountpoint>`), daher ist der cgroup-v2-Rückfall in der Praxis der normale LXC-Weg. Nicht ohne Prüfung auf dem eigenen System von einem der beiden Ergebnisse ausgehen.

Mehr: `_cfg/freeze4snap/freeze4snap.readme` im Frontend und howto.ai/cs-freeze4snap.info. Lizenz: BSD 2-Clause (siehe LICENSE), dieselbe wie bei cs-sync und cs-stream.