


# Backupstrategie - Leitfaden für eine belastbare Datensicherung

## Ziel:

Eine praxiserprobte, einfach umsetzbare Strategie, die Ausfälle (Hardware, Ransomware, Bedienfehler), versehentliche Löschungen und Katastrophen (Brand, Diebstahl) abfedert - und **\*\*Wiederherstellung\*\*** planbar macht.

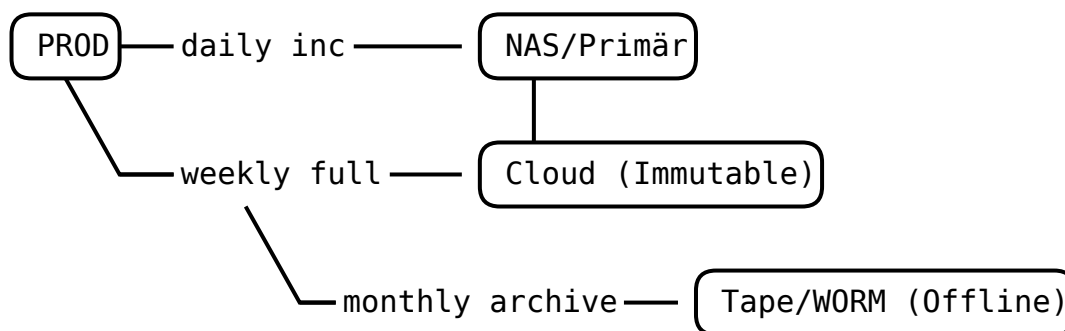
## 1) Grundlagen & Ziele

- **RPO** (Recovery Point Objective): Wie **alt** dürfen Daten im schlimmsten Fall sein? (z. B. max. 24 h → tägliche Sicherung)
- **RTO** (Recovery Time Objective): Wie **schnell** muss es wieder laufen? (z. B. 4 h → schnelle Medien/Automatisierung nötig)
- **Scope**: Welche Systeme/Dienste/Daten sind drin? (Server, VMs, Docker-Volumes, Datenbanken, Konfigs, Zertifikate, Endgeräte, SaaS)

 **Merke:** Eine Backupstrategie ist nur so gut wie die **Rücksicherung**. Ohne Restore-Test ist alles nur Hoffnung.

## 2) Die 3-2-1-1-0-Regel (Best Practice)

- **3** Kopien: Produktiv + 2 Backups
- **2** verschiedene Medien/Technologien (z. B. Disk/NAS **und** Cloud/Tape)
- **1** Kopie **außer Haus** (Offsite)
- **1** Kopie **offline/immutable** (z. B. S3 Object Lock, WORM-Tape)
- **0** Fehler nach **regelmäßiger Verifikation** (Prüfsummen, Restore-Tests)



### 3) Risikoanalyse kurz & knackig

Risiko	Beispiel	Maßnahme in der Strategie
Ransomware	Verschlüsselte Shares/VMs	Immutable Ziel + Offline Kopie + getrennte Admin-Konten
Fehlbedienung	„rm -rf“, versehentliches Löschen	Versionierung + tägliche Inkremente + schnelle Restore
Hardwaredefekt	RAID-Ausfall, defekte SSD	Mind. 2 Ziele + Offsite
Standortverlust	Brand/Diebstahl	Offsite/Cloud/Tape
Dateninkonsistenz	Offene DB-Transaktionen	Applikations-aware Backups (VSS, LVM, Snapshots)

### 4) Backup-Arten & wann sie Sinn machen

- **Vollbackup (Full):** komplette Kopie; Basis für lange Retention (z. B. wöchentlich/monatlich).
- **Inkrementell (Inc):** nur Änderungen seit dem **letzten** Backup (schnell, platzsparend).
- **Differenziell (Diff):** Änderungen seit dem **letzten Vollbackup** (Kompromiss).
- **Synthetic Full:** Vollbackup, das aus früheren Voll+Inkrementen **zusammengebaut** wird (entlastet Quelle).

**Praxis:** Täglich inkrementell, wöchentlich synthetisches Full, monatlich „echtes“ Full als Archiv.

### 5) Zeitplan (Beispiel GFS - Grandfather/Father/Son)

Ebene	Zyklus	Aufbewahrung	Medium/Ziel
Son	täglich (inkrementell)	7-14 Tage	schneller Storage/NAS
Father	wöchentlich (Full/Synthetic)	4-8 Wochen	NAS + Offsite
Grandfather	monatlich (Full)	6-12 Monate	Offsite (Cloud immutable/Tape)
Yearly	jährlich (Full)	5-10 Jahre*	Tape/WORM (Compliance) <sup>1)</sup>

### 6) Speicherziele & Medien

- **Disk/NAS:** schnell für tägliche Restores; **RAID ≠ Backup**.
- **Cloud/Object Storage:** S3-kompatibel mit **Object Lock** (immutability, Legal Hold).
- **Tape (LTO):** günstig für lange Archivierung; echte **Offline-Air-Gap**.
- **Snapshots** (ZFS/Btrfs/LVM/Hypervisor): super **Ergänzung**, aber **kein** Ersatz für **externes Backup**.

## 7) Konsistenz & Applikations-Awareness

- **Datenbanken:** konsistente Dumps oder Hot-Backup-Mechanismen (z. B. `mysqldump --single-transaction`, `pg_dump`, `xtrabackup`).
  - **VMs:** applikationskonsistente Snapshots (VSS bei Windows-Gästen).
  - **Dateiserver:** Snapshots + Backup; offene Dateien via VSS oder LVM-Snapshot sichern.
  - **Container/Docker: Volumes** sichern (nicht nur Images), außerdem Compose-Dateien, `.env`, Secrets, TLS-Zertifikate, Datenbanken **im Container** per Dump sichern.
- 

## 8) Sicherheit: Verschlüsselung, Schlüssel, Zugriffe

- **Transport & Ruhe:** Ende-zu-Ende verschlüsseln (z. B. `restic/borg`, Key Management).
  - **Schlüssel/Passphrasen:** separat und **offline** hinterlegen (Tresor, Sealed Envelope).
  - **Trennung:** Backup-Server/Repo mit **eigenen** Admin-Konten, MFA, Netzwerksegmentierung, kein Domain-Admin.
  - **Immutable/WORM:** aktivieren, um Löschungen/Manipulation zu verhindern.
- 

## 9) Überwachung & Tests

- **Monitoring/Alarmer:** tägliche Report-Mails/Webhook (z. B. Healthchecks), fehlgeschlagene Jobs = **Pager**.
  - **Integrität:** `restic check`, `borg check`, Prüfsummen.
  - **Restore-Drills:** Mind. **quartalsweise** Testwiederherstellung (Datei, komplette VM, Datenbank).
  - **RTO/RPO-Beweis:** Dauer messen & dokumentieren.
- 

## 10) Prozesse, Rollen, Dokumentation

- **Backup-Policy** (Was, wie oft, wohin, wie lange, wer darf löschen?).
  - **Runbook** für Notfälle (Wer ruft wen an? Reihenfolge der Restores?).
  - **Change-Mgmt:** neue Systeme automatisch ins Backup aufnehmen.
  - **Revision/DSGVO:** Datenklassen, Löschkonzepte, Auftragsverarbeitung, Verschlüsselung.
- 

## 11) Praxisbeispiele (Linux/Windows/Docker/Hypervisor)

### 11.1 Linux - mit `restic` (verschlüsselt, effizient)

```
# 1) Repo initialisieren (Beispiel: S3-kompatibles Ziel)

export
RESTIC\_REPOSITORY="s3:[https://s3.example.com/bucket](https://s3.example.com/bucket)"
export RESTIC\_PASSWORD="STRONG-PASSPHRASE"
export AWS\_ACCESS\_KEY\_ID="AKIA..."
export AWS\_SECRET\_ACCESS\_KEY="..."

restic init

# 2) Täglich: inkrementelles Backup inkl. wichtiger Verzeichnisse

restic backup \
/etc /var/backups /home /opt/stacks \
--tag daily

# 3) Retention (Beispiel GFS)

restic forget --keep-daily 14 --keep-weekly 8 --keep-monthly 12 --prune

# 4) Integritätscheck (z. B. wöchentlich)

restic check
```

## 11.2 BorgBackup (schnell, lokal/SSH)

```
export BORG\_REPO="ssh://backup@backuphost:/repositories/node1"
export BORG\_PASSPHRASE="STRONG-PASSPHRASE"

borg init --encryption=repokey-blake2 "\$BORG\_REPO"
borg create --stats --progress "\$BORG\_REPO" :: "{now:%Y-%m-%d\_%H-%M}" \
/etc /var/backups /home /opt/stacks
borg prune -v --list "\$BORG\_REPO" --keep-daily=14 --keep-weekly=8 --keep-monthly=12
borg check -v "\$BORG\_REPO"
```

## 11.3 Datenbanken

```
# MariaDB/MySQL (konsistent via single-transaction)

mysqldump --single-transaction --routines --triggers --databases appdb \
| gzip > /var/backups/mysql/appdb\_$(date +%F).sql.gz

# PostgreSQL
```

```
pg\_dump -Fc appdb > /var/backups/pgsql/appdb\_\\$(date +%F).dump
```

### 11.4 Docker-Volumes & Konfiguration

```
# Named Volume als Tar sichern

docker run --rm\;
-v meine\_daten:/src\:ro\;
-v /var/backups/docker:/dst\;
alpine sh -c 'cd /src && tar czf /dst/meine\_daten\_\\$(date +%F).tgz .'
```

```
# Compose & .env & Labels sichern (Infrastructure as Code)

cp -a /opt/stacks/\* /var/backups/stacks/\$(date +%F)/
```

### 11.5 Proxmox/Hyper-V (Beispiele)

```
# Proxmox: vzdump (vollständig, konsistent)

vzdump 100 --mode snapshot --compress zstd --storage backup-nfs --
mailnotification always

# Windows/Hyper-V: Bare-Metal/Volumes (als Admin PowerShell)

wbadmin start backup -backupTarget\E: -include\C: -allCritical -quiet
```

## 12) Beispiel-Zeitpläne (cron)

Aufgabe	cron	Kommando/Script
Täglich 01:00 Inkrement	0 1 * * *	restic backup ... && curl https://hc/ping/ok
Wöchentlich So 02:00 Full	0 2 * * 0	restic backup --tag weekly && restic forget ...
Monatlich 03:00 Check/Prune	0 3 1 * *	restic check && restic prune
DB-Dumps 00:30	30 0 * * *	mysqldump/pg_dump ...
Volumes 00:45	45 0 * * *	docker run ... tar czf ...

## 13) Notfall-Runbook (Template)

== Incident: Datenverlust/Ransomware ==

Zeitpunkt: \_\_\_\_\_ Ticket: \_\_\_\_\_ Melder: \_\_\_\_\_

1. Lage bewerten: Welche Systeme betroffen? RPO/RT0?
2. Isolieren: Netzwerkzugriff einschränken, Admin-Konten prüfen.
3. Quelle wählen: Jüngstes sauberes Backup (Immutable bevorzugt).
4. Restore-Reihenfolge:
  - a) Verzeichnis-/Datei-Restore für schnelle Wiederanlaufpunkte
  - b) Datenbanken + Applikationen (abhängigkeitsbasiert)
  - c) Dienste/VMs in Startreihenfolge
5. Validierung: Smoke-Tests, Integrität, Anwendertests.
6. Dokumentation: Dauer (RT0), Datenalter (RPO), Lessons Learned.

---

## 14) Checkliste „Sofort anfangen“

- RPO/RT0** pro System festlegen und dokumentieren
- 3-2-1-1-0** konkret planen (Ziele, Medien, Offsite, Immutable)
- Backup-Software** wählen (restic/borg/veeam/Proxmox/...)
- Sensitive Daten** verschlüsseln; **Keys** offline sichern
- Automatisierung** (cron/systemd, Jobs, Webhooks/Alarmer)
- Onboarding-Prozess** für neue Systeme/Container/DBs/
- Restore-Test** (Datei, VM, DB) terminieren & protokollieren
- Retention** (täglich/wöchentlich/monatlich/jährlich) umsetzen
- Rechte & Trennung** (separate Admins, MFA, Netzsegmente)
- Compliance** (DSGVO/GoBD) prüfen: Speicherorte, Löschkonzept

---

## 15) Häufige Fehlerquellen (und Gegenmaßnahmen)

- **Nur Snapshots ≠ Backup** → Immer **externes** Ziel einplanen.
- **Keine Offsite/Immutable-Kopie** → Ransomware-Risiko massiv.
- **DBs „kalt“ gesichert** → inkonsistent → applikations-aware sichern.
- **Keine Restore-Tests** → im Ernstfall Überraschungen.
- **Schlüssel verloren** → verschlüsselte Backups unbrauchbar → Key-Runbook & sichere Offline-Ablage.
- **Alles in einem VLAN/AD** → Angreifer löscht Backups → Trennung, eigene Accounts, WORM.
- **Unklare Retention** → Speicher läuft voll oder Löschung zu früh.
- **SaaS ohne Backup** → Microsoft/Google sind **keine** Backuplösungen → Drittanbieter/Export einplanen.

---

## 16) Minimales Beispiel für kleine Umgebungen

- **Täglich:** restic backup (Dateien, Docker-Volumes), **14 Tage** auf NAS
- **Wöchentlich:** Synthetic Full → NAS **und** S3 (Object Lock 30 Tage)



From:  
<http://wiki.nctl.de/dokuwiki/> - ☐ **Veni. Vidi. sudo rm -rf / vici.**

Permanent link:  
<http://wiki.nctl.de/dokuwiki/doku.php?id=it-themen:allgemein:backup&rev=1755447930>

Last update: **17.08.2025 18:25**

