

Restic Backup

☐ Restic Backup & Restore

Vollständige Anleitung für Linux | Sicher • Verschlüsselt • Zuverlässig

1. Was ist Restic?

Restic ist ein modernes, kostenloses Open-Source-Backup-Programm für Linux, macOS und Windows. Es zeichnet sich durch folgende Eigenschaften aus:

Eigenschaft	Beschreibung
Verschlüsselung	Alle Backups werden automatisch mit AES-256 verschlüsselt
Deduplizierung	Gleiche Daten werden nur einmal gespeichert – spart Speicherplatz
Inkrementell	Nur geänderte Dateien werden gesichert – sehr schnell
Viele Ziele	Lokal, externe Festplatte, Nextcloud (WebDAV), S3, SFTP, u.v.m.
Open Source	Kostenlos, transparent, aktiv weiterentwickelt
Cross-Platform	Backup auf Linux erstellen, auf Windows wiederherstellen

2. Installation

Debian / Ubuntu

```
sudo apt update && sudo apt install restic
```

Fedora / Rocky Linux / AlmaLinux

```
sudo dnf install restic
```

Aktuellste Version manuell installieren

Die Version in den Paketquellen ist manchmal veraltet. Die neueste Version so aktualisieren:

```
sudo restic self-update
```

☐ Nach der Installation immer mit `restic version` prüfen, ob die Installation geklappt hat.

3. Repository einrichten

Restic speichert Backups in einem sogenannten **Repository** (kurz: Repo). Dieses muss einmalig initialisiert werden.

3.1 Lokales Repository (externe Festplatte)

```
restic init --repo /mnt/externe-festplatte/backups
```

Restic fragt nach einem Passwort – dieses ist sehr wichtig! Ohne Passwort kann das Backup **nicht** wiederhergestellt werden.

⚠ Das Passwort unbedingt sicher aufbewahren! Am besten in einem Passwortmanager (z.B. KeePassXC) oder ausgedruckt an einem sicheren Ort.

3.2 Nextcloud Repository (WebDAV)

Restic kann direkt in deine Nextcloud sichern. Dazu wird das Tool **rclone** als Brücke verwendet.

Schritt 1: rclone installieren

```
sudo apt install rclone
```

Schritt 2: rclone konfigurieren

```
rclone config
```

Im interaktiven Menü folgendes auswählen:

- **n** für neues Remote
- Name: `nextcloud`
- Typ: `webdav`
- URL: `https://deine-nextcloud.de/remote.php/dav/files/BENUTZERNAME/`
- Vendor: `nextcloud`
- Benutzername und Passwort eingeben

Schritt 3: Repository initialisieren

```
restic init --repo rclone:nextcloud:/Restic-Backups
```

Der Ordner "Restic-Backups" wird automatisch in deiner Nextcloud angelegt.

4. Backup erstellen

4.1 Home-Verzeichnis sichern (empfohlen)

```
restic backup --repo /mnt/externe-festplatte/backups /home/BENUTZERNAME
```

Für Nextcloud entsprechend:

```
restic backup --repo rclone:nextcloud:/Restic-Backups /home/BENUTZERNAME
```

4.2 Ganzes System sichern

Um das komplette System zu sichern, sollten einige Systemverzeichnisse ausgeschlossen werden:

```
restic backup --repo /mnt/externe-festplatte/backups / \  
--exclude /proc \  
--exclude /sys \  
--exclude /dev \  
--exclude /run \  
--exclude /tmp \  
--exclude /mnt \  
--exclude /media
```

⚠ Für System-Backups müssen Root-Rechte verwendet werden: `sudo restic backup ...`

4.3 Backup mit Tags

Tags helfen beim Organisieren und späteren Wiederherstellen:

```
restic backup --repo /mnt/externe-festplatte/backups /home/BENUTZERNAME --tag wöchentlich
```

4.4 Passwort-Eingabe automatisieren

Um das Passwort nicht jedes Mal eintippen zu müssen, kann es in einer Datei gespeichert werden:

```
echo 'dein-sicheres-passwort' > ~/.restic-passwort  
chmod 600 ~/.restic-passwort
```

Dann beim Aufruf:

```
restic backup --repo /mnt/externe-festplatte/backups --password-file ~/.restic-passwort /home/BENUTZERNAME
```

5. Backups anzeigen und verwalten

5.1 Alle Snapshots anzeigen

```
restic snapshots --repo /mnt/externe-festplatte/backups
```

Beispielausgabe:

ID	Zeit	Host	Pfade
a3b4c5d6	2025-02-20 10:00:00	laptop	/home/max
e7f8a9b0	2025-02-22 10:00:00	laptop	/home/max

5.2 Inhalt eines Snapshots anzeigen

```
restic ls --repo /mnt/externe-festplatte/backups latest
```

5.3 Alte Snapshots löschen (Aufräumen)

Restic kann automatisch alte Snapshots löschen und nur eine bestimmte Anzahl behalten:

```
restic forget --repo /mnt/externe-festplatte/backups \  
--keep-daily 7 \  
--keep-weekly 4 \  
--keep-monthly 6 \  
--prune
```

Diese Einstellung behält: **7 tägliche + 4 wöchentliche + 6 monatliche** Snapshots.

`--prune` entfernt auch die eigentlichen Daten und gibt Speicherplatz frei.

6. Backup wiederherstellen (Restore)

6.1 Komplette Wiederherstellung

```
restic restore latest --repo /mnt/externe-festplatte/backups --target /
```

⚠ Beim Wiederherstellen auf `/` (Root) werden vorhandene Dateien überschrieben! Nur im Notfall verwenden.

6.2 In ein anderes Verzeichnis wiederherstellen (empfohlen)

Sicherer ist es, erst in ein temporäres Verzeichnis wiederherzustellen:

```
restic restore latest --repo /mnt/externe-festplatte/backups --target /tmp/wiederherstellung
```

Danach kannst du gezielt einzelne Dateien zurückkopieren.

6.3 Bestimmten Snapshot wiederherstellen

Zuerst alle Snapshots anzeigen:

```
restic snapshots --repo /mnt/externe-festplatte/backups
```

Dann den gewünschten Snapshot mit seiner ID wiederherstellen:

```
restic restore a3b4c5d6 --repo /mnt/externe-festplatte/backups --target /tmp/wiederherstellung
```

6.4 Einzelne Datei oder Ordner wiederherstellen

Mit `--include` kannst du nur bestimmte Pfade wiederherstellen:

```
restic restore latest --repo /mnt/externe-festplatte/backups \  
--target /tmp/wiederherstellung \  
--include /home/max/Dokumente
```

6.5 Backup als Dateisystem mounten

Du kannst Snapshots wie ein Dateisystem einhängen und darin stöbern – ohne etwas wiederherstellen zu müssen:

```
mkdir /tmp/restic-mount  
restic mount --repo /mnt/externe-festplatte/backups /tmp/restic-mount
```

Danach kannst du mit dem Dateimanager in `/tmp/restic-mount` alle alten Backups durchsuchen. Mit **Strg+C** beendest du den Mount.

7. Automatisches Backup einrichten

7.1 Backup-Skript erstellen

Erstelle die Datei `/usr/local/bin/restic-backup.sh`:

```
#!/bin/bash
REPO=/mnt/externe-festplatte/backups
PASSWORT_DATEI=/root/.restic-passwort

# Backup erstellen
restic backup \
  --repo $REPO \
  --password-file $PASSWORT_DATEI \
  /home/BENUTZERNAME

# Alte Snapshots aufräumen
restic forget \
  --repo $REPO \
  --password-file $PASSWORT_DATEI \
  --keep-daily 7 \
  --keep-weekly 4 \
  --keep-monthly 6 \
  --prune
```

```
sudo chmod +x /usr/local/bin/restic-backup.sh
```

7.2 Automatisch per Cron ausführen

Crontab öffnen:

```
sudo crontab -e
```

Folgende Zeile einfügen (täglich um 02:00 Uhr):

```
0 2 * * * /usr/local/bin/restic-backup.sh >> /var/log/restic.log 2>&1
```

7.3 Systemd Timer (modernere Alternative)

Erstelle `/etc/systemd/system/restic-backup.service`:

```
[Unit]
Description=Restic Backup
```

```
[Service]
Type=oneshot
ExecStart=/usr/local/bin/restic-backup.sh
```

Erstelle `/etc/systemd/system/restic-backup.timer`:

```
[Unit]
Description=Restic Backup täglich

[Timer]
OnCalendar=daily
Persistent=true

[Install]
WantedBy=timers.target
```

Aktivieren:

```
sudo systemctl daemon-reload
sudo systemctl enable --now restic-backup.timer
```

8. Backup-Integrität prüfen

Es ist wichtig, Backups regelmäßig zu prüfen um sicherzustellen, dass sie im Notfall wirklich funktionieren.

Schnelle Prüfung (nur Metadaten)

```
restic check --repo /mnt/externe-festplatte/backups
```

Vollständige Prüfung (alle Daten)

```
restic check --read-data --repo /mnt/externe-festplatte/backups
```

☐ Die vollständige Prüfung kann bei großen Backups lange dauern. Empfehlung: einmal monatlich durchführen.

9. Nützliche Umgebungsvariablen

Um Repository und Passwort nicht bei jedem Befehl angeben zu müssen, kannst du diese in deine `~/.bashrc` eintragen:

```
export RESTIC_REPOSITORY=/mnt/externe-festplatte/backups
export RESTIC_PASSWORD_FILE=/home/BENUTZERNAME/.restic-passwort
```

Danach reicht einfach:

```
restic backup /home/BENUTZERNAME
restic snapshots
restic restore latest --target /tmp/wiederherstellung
```

10. Schnellreferenz – Alle wichtigen Befehle

Befehl	Beschreibung
<code>restic init --repo /pfad</code>	Repository initialisieren
<code>restic backup --repo /pfad /home/user</code>	Backup erstellen
<code>restic snapshots --repo /pfad</code>	Alle Snapshots anzeigen
<code>restic restore latest --repo /pfad --target /ziel</code>	Letztes Backup wiederherstellen
<code>restic restore ID --repo /pfad --target /ziel</code>	Bestimmten Snapshot wiederherstellen
<code>restic ls --repo /pfad latest</code>	Inhalt des letzten Backups anzeigen
<code>restic mount --repo /pfad /mnt/punkt</code>	Backup als Dateisystem einhängen
<code>restic forget --keep-daily 7 --prune</code>	Alte Backups löschen
<code>restic check --repo /pfad</code>	Backup-Integrität prüfen
<code>restic self-update</code>	Restic auf neueste Version aktualisieren

11. Wichtige Tipps

△ **Passwort sichern!** Ohne Passwort ist das Backup wertlos. Das Passwort niemals vergessen oder verlieren – es gibt keinen Wiederherstellungsweg!

☐ **3-2-1 Regel:** 3 Kopien der Daten, auf 2 verschiedenen Medien, davon 1 außerhalb (z.B. Nextcloud). Nur ein lokales Backup reicht nicht!

☐ **Backups regelmäßig testen!** Ein Backup, das man nie getestet hat, ist unzuverlässig. Stelle mindestens einmal pro Quartal eine Testdatei wieder her.

△ **Externe Festplatte nach dem Backup abziehen.** Schadsoftware (Ransomware) kann sonst auch das Backup beschädigen oder verschlüsseln.

Restic Backup & Restore Anleitung – Für Linux | Erstellt Februar 2026

Revision #1

Created 23 February 2026 20:57:32 by Hermann

Updated 23 February 2026 21:01:16 by Hermann