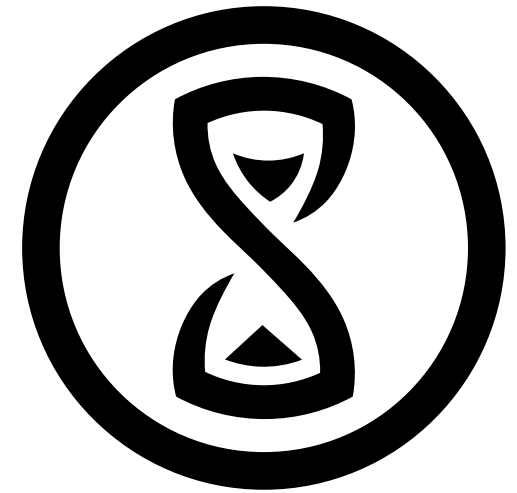


Backups mit Restic

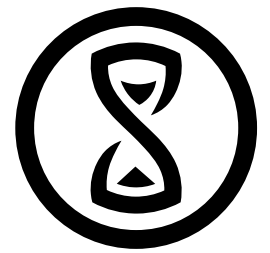
Joke

12.07.2024



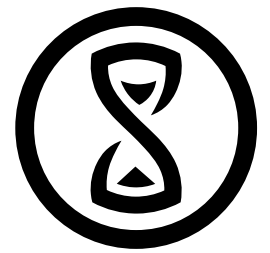
Stratum 0

Wozu?



Stratum 0

- Ich musste neulich auf der Arbeit ein Backup einrichten
- Genauer gesagt: Es gab schon ein Skript auf rsync-Basis, das sollte auf andere Hosts übernommen werden
- Verhielt sich aber nicht, wie es sollte
- Privat nutze ich seit zwei Monaten restic, seitdem ich auf Kristian Köhntopps Blogpost dazu gestoßen bin:
<https://blog.koehntopp.info/2023/12/30/restic.html>
- Also bekamen besagte Server auch ein Skript auf restic-Basis

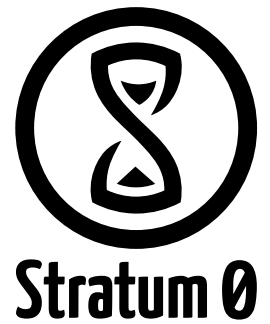


Stratum 0

Was ist dieses restic?

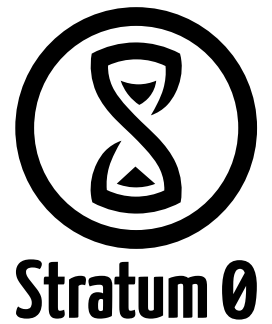
- Ursprünglich von Alexander „fd0“ Neumann entwickeltes in go geschriebenes Backupprogramm, mittlerweile mit helfenden Händen
- Vorträge von fd0 auf CCC-Erfas und froscon:
<https://media.ccc.de/search/?q=restic>
 - Damals gab es noch weniger Features, aber Grundkonzept war schon alles da
- „Backups done right“

Erwähnenswerte Features



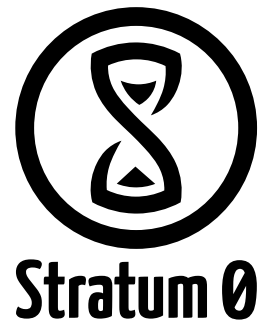
- Statisch gelinktes single binary
- Sehr performant (initiales Backup dauert, jedes weitere dagegen sehr schnell, solange sich nicht viel ändert)
- Deduplizierung plus Kompression: Sehr große gespeicherte Datenmengen bei sehr wenig Speicherverbrauch
 - Bei meinen privaten Backup 11,424 TB „ausgepackt“, 230 GB Repogröße
- Backups IMMER verschlüsselt und mit Passwort geschützt
- Erlaubt in Kombination eine timemachine ala Apple
- Läuft auf Linux, BSDs, MacOS X und Windows
- Unterstützt lokalen Storage, sftp, REST-Api und diverse Cloudspeicher (nativ oder über rclone)

Installation



- Aktuelle Binary von <https://restic.net/> herunterladen und im Path ablegen
- Falls Bedarf:
 - Shell completion files mit `restic generate` erzeugen:
 - `restic generate --zsh-completion restic-zsh`
 - Geht auch für bash, fish und Powershell
 - Manual pages generieren:
 - `restic generate -man /usr/local/share/man1/`

Repositories

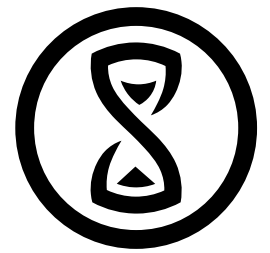


- Restic speichert die Daten in Repositories
- Das muss man sich erstmal erzeugen
- *restic init -r repository_url*
- Dabei wird man nach einem Passwort gefragt
- Das muss man sich merken, sonst kommt man nicht mehr an die Backups dran!
- Deduplizierung erfolgt innerhalb eines Repositories
- Abwägung: Will ich Speicherplatz sparen oder in Zweifelsfall Daten eines nicht mehr existenten Hosts oder Users schnell loswerden?

Umgebungsvariablen

- Falls man keine Lust auf Tippen hat:
 - *RESTIC_REPOSITORY=sftp://restic_host/repoordner*
 - *RESTIC_PASSWORD=GEHEIM!*
- Standardmäßig werden alle CPU-Cores verwendet
 - Kann man mit GOMAXPROCS einschränken:
export GOMAXPROCS=4
- Alle weiteren (für AWS, Azure..) sind in der (auch sonst sehr ausführlichen) Doku:
https://restic.readthedocs.io/en/stable/040_backup.html#environment-variables

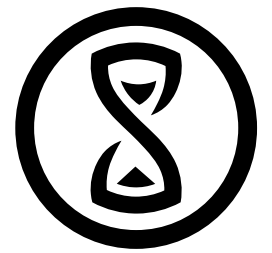
Backups handhaben



Stratum 0

- *restic backup ordner # Test auf HOA für ca 900 MB 45 Minuten, jedes weitere unter einer Minute*
- Anpassung erfolgt NICHT über Konfigurationsdatei sondern Optionen, z.B -r für Repository, --exclude ..
- Informationen abrufen:
 - *restic snapshots # Vorhandene Snapshots anzeigen*
 - *restic stats # Statistiken (Datenmenge und so) anzeigen*
 - *restic check # Repository auf Fehler kontrollieren*
 - *restic ls snapshot_id # Dateien im Snapshot anzeigen*
 - *restic ls latest # Dateien im aktuellsten Snapshot im Repo anzeigen*
 - *restic find pattern # Suche im im Snapshots*

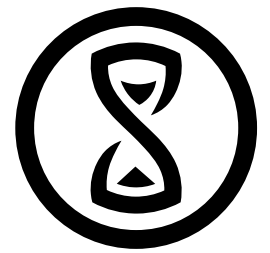
Nobody wants backup, everyone wants restore



Stratum 0

- Für große Datenmengen am Schnellsten und geht überall:
 - `restic restore snapshotid --target /restore_ordner`
 - `restic restore latest --target /restore_latest`
 - `restic restore latest --path /etc --target /restore_etc`
 - `restic restore latest --host ruebezahl --target /restore_etc`
 - `restic dump latest datenbank.sql | mysql`
- Unter Linux, FreeBSD und MacOS X:
 - `restic mount /mnt/restore`
 - Fuse-Mount zum Browsen

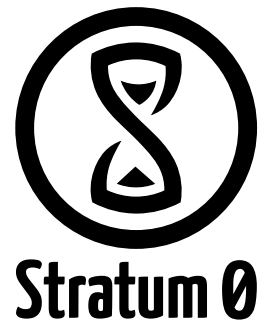
Backups aufräumen



Stratum 0

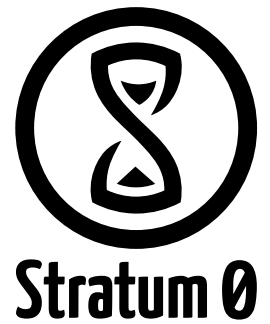
- `restic forget snapshotid`
- Beispiel Kristian Köhntopp:
`restic forget --keep-daily 30 --keep-monthly 12 --keep-within 2d --keep-tag keep`
- Daten belegen aber weiter Platz!
- Bereinigung dann mit `restic prune` oder `restic forget -prune`
- Achtung: Ressourcenhungrig!

Automatisierung



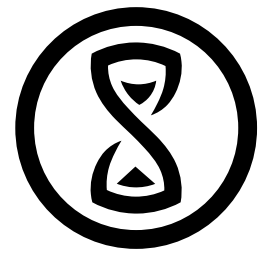
- Keine Einstellungsdatei, was nun?
- Shellskript schreiben und per cron-job oder systemd-timer aufrufen
- Alternative: restic-profile als Frontend nutzen:
 - <https://creativeprojects.github.io/resticprofile/>
 - Ermöglicht Backupprofile als Konfigurationsdatei abzuspeichern
 - Das dann wieder über cronjob oder systemd-timer aufrufen ;)
- Disclaimer: Ich habe da bisher kaum Erfahrung mit

restic-Server



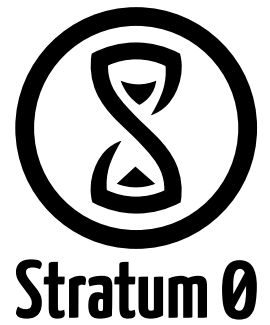
- restic spezifiziert eine bestimmte REST-API
- Idee: Jedes Storage, dass diese API implementiert, kann genutzt werden
- Nebenprojekt: <https://github.com/restic/rest-server>
- Implementiert diese API
- Kann in append-only Mode laufen
 - Es können nur neue Daten hinzugefügt werden durch clients, nichts gelöscht!
 - Wichtig, falls der zu backupende Host gecybert wird!
 - Der Backupserver muss dann aber natürlich sehr gut gesichert werden
 - forget- und prune- kann nicht mehr vom Client stattfinden, sondern muss als regelmäßiger Cron-/SystemD-Job auf den Server laufen

Demo :)



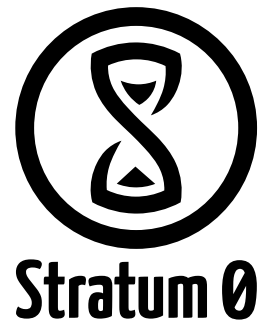
Stratum 0

Stolperfallen



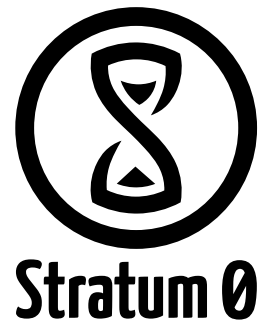
- Initiales Backup kann sehr lange dauern
- Passende Parameter passend zusammenstellen am Anfang hakelig
- In Skript zu gießen auch → restic-profile
- Dokumentation hilft aber sehr plus die Communityforen
- Was ist, wenn man das Passwort vergisst oder es Bitkipper gibt?
 - Zusatzbackup in alternative einplanen!
 - Bei mir: rsnapshot auf externe Festplatte
 - To be fair: Es gibt auch Reperaturmechanismen → Doku
- Ressourcenanforderungen:
 - Standardmäßig werden alle CPU-Kerne genutzt
 - Je mehr Daten, desto höherer RAM-Verbrauch
 - Kann mit Parametern teilweise gesteuert werden

Verwendung in der richtigen Welt



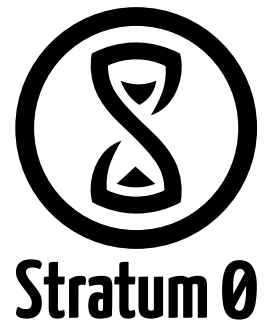
- Cern hat ihr cbackup darauf basiert
- Frontend mit Agents und Jobs um restic als Backend
- Sie vertrauen ihren Usern bezüglich cleanup nicht ;)
- Storage auf ceph-Cluster (S3)
- Stand 2024: 18 PB (18000 TB!), 2.5 Mrd Dateien
- Langsam wird das teuer (Festplatten kosten), cold storage wäre cool
→ Fork mit Support für Tapes als Cold-Storage und Trennung Metadaten/Archivdaten
 - Details:
<https://cds.cern.ch/record/2855376?ln=de>
 - https://www.epj-conferences.org/articles/epjconf/abs/2024/05/epjconf_chep2024_01027/epjconf_chep2024_01027.html

Moment, mal Fork?



- Wie so viele OpenSource Projekte:
 - Zu viele Feature- und Mergerequests, zu wenig Leute die reviewen/dabei helfen
 - Cold storage wird diskutiert (Details in den github issues), aber es ist auch kein triviales Problem (siehe cern-Paper)
 - Naive Lösung die Repos direkt aufs tape zu schreiben problematisch

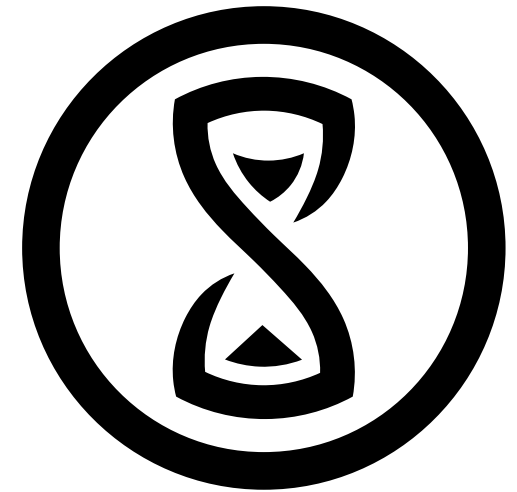
rustic



- Projekt eines Menschen, der auch viel bei restic mit diskutiert und MergeRequests stellt
- rustic implementiert restic in Rust plus aus Sicht des Autoren wünschenswerte Features (wie cold storage)
- Soll kompatibel bleiben
- <https://github.com/rustic-rs>
- Aber:
"rustic currently is in beta state and misses regression tests. It is not recommended to use it for production backups, yet."

Danke für eure Aufmerksamkeit!

Stratum 0 e.V. Braunschweig
<https://stratum0.org>



Stratum 0