

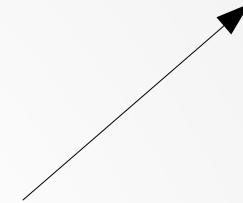


# Backups mit `restic[profile]`

Jan Grosser, TroLUG, 2020/05/07

# Inhalt

- Wie macht Ihr Backup?
- restic
  - Überblick
  - Backups erstellen
  - Wiederherstellen
  - Grafisches Interface (GUI)?
- resticprofile
  - Überblick
  - Konfig-Datei
  - Backup erstellen
  - Bsp-Profile
- [Verifizieren und Reparieren]
- Zusammenfassung



Wenn Ihr das Symbol seht,  
spielen wir an der Konsole!

Wie macht Ihr Backup?  
Jeder ~ 1 Minute



A word cloud of backup tools. The word 'restic' is the largest and most prominent, centered in the image. Other tools are scattered around it in various sizes and orientations. The background is white with a green gradient at the top.

# restic

rear

dd

BorgBackup

dar

Deja Dup

Unison

Obnam

KDar

rsync

csync

duplicity

cp

fsarchiver

Fullsync

Grsync

Drag&Drop

rsnapshot

rdiff-backup

flyback

Bacula

rbackup

# restic - Überblick

- <https://restic.net/>
- „restic is a program that does backups right“
  - Einfach
  - Schnell
  - Verifizierbare Backups
  - Sicher – AES-256 Verschlüsselung
  - Effizient – inkrementelle Backups, De-Duplizierung
  - Remote Backups möglich – Amazon S3, Google Cloud, SFTP, uvm
  - Frei – BSD 2-Clause License
  - v0.9.6, wir aktiv weiterentwickelt.



# restic -Begriffe



## Repository

Snapshot  
40508607

Snapshot  
0baa29f9

Snapshot  
2d4e9151

Snapshot  
da978ab7

index

tree

blob

# restic – Backups erstellen



- Umgebungsvariable setzen (Beispiel):  
`$ export RESTIC_REPOSITORY="/backup/restic"`
- Repository initialisieren  
`$ restic init`
- Backups erzeugen  
`$ restic backup /home/user/testdata/`
- Snapshots anzeigen  
`$ restic snapshots`

# restic - Wiederherstellen (1/2)



- Datei in Backup finden

```
$ restic find -l <Datei>
```

- Backup wiederherstellen

```
$ restic restore <id> --target <dir>
```



# restic - Wiederherstellen (2/2)



- Repository mit FUSE mounten

```
$ restic mount <dir>
```

# restic – GUI?

- restic ist reines Kommandozeilen-Tool
- Daher gut automatisierbar bspw. mittels cron Scheduler
- GUIs
  - Restatic (Alpha),  
<https://github.com/Mebus/restatic>
  - ???

# resticprofile

# resticprofile - Überblick

- <https://pypi.org/project/resticprofile/>
- „Wrapper“ für restic
- Steuerung von restic über Konfigurations-Dateien
- Mehrere, voneinander abhängige Profile möglich
- Vor und nach Backup können Befehle oder Skripte ausgeführt werden
- Frei - GPLv3+
- v0.5.2 - wird aktiv weiterentwickelt.

# resticprofile Konfig-Datei (1/3)

/etc/restic/restic-backup.conf

[default]

repository = "sftp:username@sftp.hidrive.strato.com:/users/username/restic-backup"

password-file = "/etc/restic/restic-backup.pwd"

initialize = false

[default.env]

TMPDIR = "/tmp"

[default.backup]

verbose = true

one-file-system = true

source = [ "/home/foo", "/etc", "/usr/local" ]

exclude-caches = true

exclude-file = "/etc/restic/restic-backup.exclude"

# resticprofile Konfig-Datei (2/3)

/etc/restic/restic-backup.conf

[default.retention]

# Multiple policies will be ORed together so as to be as inclusive as possible

# for keeping snapshots.

before-backup = false

after-backup = true

# Only consider snapshots from this host (mymachine). Leave snapshots from other

# hosts untouched

host = "mymachine"

keep-last = 10

keep-daily = 7

keep-weekly = 8

keep-monthly = 12

keep-yearly = 2

keep-tag = [ "keep" ]

compact = false

prune = false

# resticprofile Konfig-Datei (3/3)

/etc/restic/restic-backup.exclude

/home/foo/Downloads

/home/foo/cloud

/home/foo/temp

/home/foo/.cache

/usr/local/src

# resticprofile Backup erstellen



- Backup mit Profil „default“ starten

```
$ sudo resticprofile \  
  -c /etc/restic/restic-backup.conf \  
  backup
```

- Wie oben, nur mit **zusätzlichem Tag „demo“**

```
$ sudo resticprofile \  
  -c /etc/restic/restic-backup.conf \  
  backup --tag demo
```



# resticprofile - Bsp-Profile (1/3)

```
[nextcloud]
```

```
inherit = "default"
```

```
[nextcloud.backup]
```

```
tag = "cloud"
```

```
run-before = [
```

```
    "sudo -u www-data php /var/www/html/nextcloud/occ maintenance:mode --on",
```

```
    "mysqldump --single-transaction -h localhost databasename > /tmp/nextcloud.sql"
```

```
]
```

```
run-after = [
```

```
    "sudo -u www-data php /var/www/html/nextcloud/occ maintenance:mode --off"
```

```
]
```

```
source = [
```

```
    "/var/www/html/nextcloud",
```

```
    "/var/local/nc-data",
```

```
    "/etc/apache2",
```

```
    "/etc/php",
```

```
    "/etc/letsencrypt",
```

```
    "/tmp/nextcloud.sql"
```

```
]
```

# resticprofile - Bsp-Profile (2/3)

```
[etc]
```

```
Inherit = "default"
```

```
[etc.backup]
```

```
tag = "etc"
```

```
run-before = [
```

```
    "apt list --installed > /tmp/apt_installed.list"
```

```
]
```

```
source = [
```

```
    "/etc",
```

```
    "/tmp/apt_installed.list",
```

```
    "/opt",
```

```
    "/usr/local"
```

```
]
```

```
exclude-caches = true
```

```
one-file-system = true
```

# resticprofile - Bsp-Profile (3/3)

[groups]

full-backup = [ "nextcloud", "etc" ]

daily = [ ... ]

weekly = [ ... ]

monthly = [ ... ]

# Verifizieren und Reparieren

# restic - Verifizieren



- Repository überprüfen  
\$ restic check
- Option `--read-data` überprüft Prüfsummen aller Data Blobs (kann sehr lange dauern)
- Mit Option `--read-data-subset=n/m` kann man Überprüfung des Repositories aufteilen

# restic - Robustheit (1/2)



- Kopie des Repositories anlegen:

```
$ cp -ar /backup/restic /backup/restic2
```

- Datei manipulieren:

```
$ echo "bang" >> /backup/restic2/data/...
```

- Repository überprüfen:

```
$ restic -r /backup/restic2 check \  
--read-data
```

# restic - Robustheit (2/2)



- Index neu bauen:  

```
$ restic -r /backup/restic2 rebuild-index
```
- Dann restic-check erneut ausführen:  

```
$ restic -r /backup/restic2 check
```
- Wenn z.B. ein defekter tree angezeigt wird, kann man diesen mit restic-find einem Snapshot zuordnen:  

```
$ restic -r /backup/restic2 find --tree <tree-id>
```
- Dieser Snapshot kann dann gelöscht werden:  

```
$ restic -r /backup/restic2 forget \  
<snapshot-id> --prune
```

# Zusammenfassung

- restic erzeugt verschlüsselte Backups
- restic kann lokal und remote sichern
- restic erzeugt grundsätzlich inkrementelle Backups basierend auf dem letzten Snapshot mit gleichen Quellverzeichnissen (source)
- Alte Snapshots können entfernt werden, auch automatisch nach definierbaren Regeln
- Mit resticprofile kann man komplexe restic Backup-Lösungen erstellen
- restic Repositories können gemountet werden => intuitive Wiederherstellung möglich
- restic ist robust



# Danke!

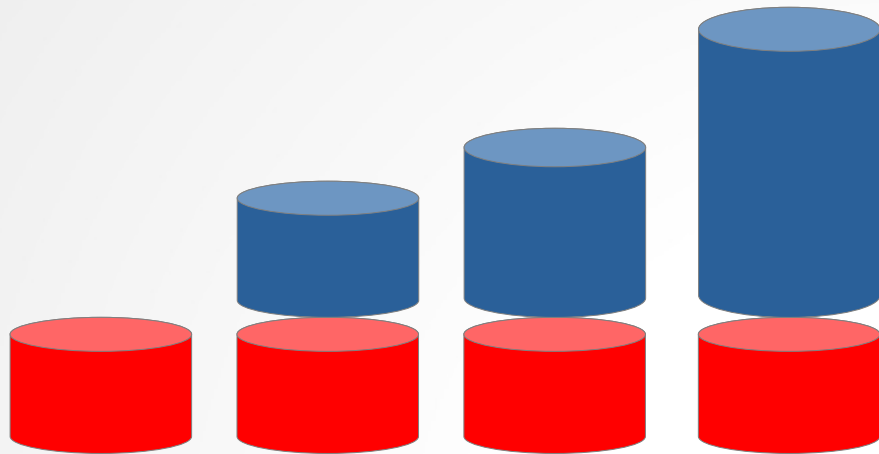
Image Credits:

- Folie 1: restic Logo, <https://restic.readthedocs.io/>
- Icons auf div. Folien: Gnome Project, `/usr/share/icons/gnome`

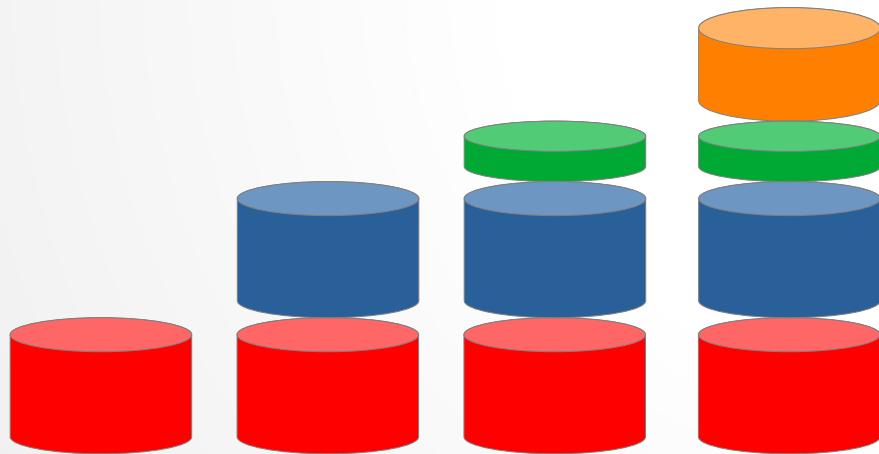
# Backup-Strategie

- Dinge, die man bei einer Backup-Strategie berücksichtigen sollte:
  - Gegen welche Risiken absichern?
  - Was soll gesichert werden?
  - Häufigkeit
  - Automatisierung, Integration
  - Verifikation/Monitoring des Backups
  - Rücksicherung vorbereiten/dokumentieren
  - Sicherung, Obsoleszenz, Kompatibilität
  - Weiterentwicklung der Strategie

# Differentiell vs inkrementell



Differentiell



Inkrementiell

