restic Encrypted backups

CONCEPTS

> Set RESTIC_REPOSITORY and RESTIC_PASSWORD_FILE once, then every command works without -r.

Files your data
↓ backup
Snapshot deduplicated, encrypted
↓ forget
Unreferenced data still in repo
↓ prune
Space freed repo shrinks

> Every snapshot is a full view, but identical chunks are stored once, so repeated backups are small and fast.

> Lose the password and the data is gone: there is no recovery. Store it separately from the repo.

REPOSITORY

`restic init`
Create a new repository

`restic -r <repo> <command>`
Select a repository for one command

`export RESTIC_REPOSITORY=<repo>`
Set the default repository

`export RESTIC_PASSWORD_FILE=<file>`
Read the password from a file

`--password-command "<cmd>"`
Get the password from a command

`restic -r sftp:<user>@<host>:<path> init`
Repository over SFTP

`restic -r s3:https://<host>/<bucket> init`
S3-compatible repository

`restic -r rest:https://<host>:8000/ init`
Repository on a REST server

`restic -r rclone:<remote>:<path> init`
Any backend via rclone

`restic key list`
List repository passwords

BACKUP

`restic backup <path>...`
Back up paths into a new snapshot

`restic backup -n -v <path>`
Dry run, show what would be added

`restic backup --exclude "<pattern>" <path>`
Skip matching files

`restic backup --exclude-file <file> <path>`
Read exclude patterns from a file

`restic backup --tag <tag> <path>`
Label the snapshot

`restic backup --one-file-system <path>`
Stay on one filesystem

`restic backup --exclude-caches <path>`
Skip dirs marked CACHEDIR.TAG

`restic backup --files-from <file>`
Read the path list from a file

`restic backup --skip-if-unchanged <path>`
No new snapshot if nothing changed

SNAPSHOTS

`restic snapshots`
List snapshots

`restic snapshots --tag <tag>`
Filter by tag

`restic snapshots --host <host> --path <path>`
Filter by host and path

`restic ls latest`
List files in a snapshot

`restic find <pattern>`
Search all snapshots for a file

`restic diff <id1> <id2>`
Compare two snapshots

`restic forget <id>`
Remove one snapshot

RESTORE

`restic restore <id> --target <dir>`
Restore a snapshot

RESTORE (cont.)

`restic restore latest --target <dir>`
Restore the newest snapshot

`restic restore latest:<subdir> --target <dir>`
Restore only one subfolder

`restic restore <id> --include "<path>" --target <dir>`
Restore only matching files

`restic dump latest <file>`
Print one file to stdout

`restic mount <dir>`
Mount the repo via FUSE

`fusermount -u <dir>`
Unmount (or press Ctrl+C)

RETENTION

`restic forget --keep-last <n>`
Keep the n newest snapshots

`restic forget --keep-daily <n> --keep-weekly <n> --keep-monthly <n>`
Tiered retention policy

`restic forget --keep-within 30d`
Keep everything of recent days

`restic forget --dry-run <keep-flags>`
Preview what would be removed

`restic forget --prune <keep-flags>`
Forget and reclaim space

`restic prune`
Delete unreferenced data

MAINTENANCE

`restic check`
Verify repo structure

`restic check --read-data`
Read and verify all pack data

`restic check --read-data-subset 10%`
Verify a random share of data

`restic unlock`
Remove stale locks

`restic copy --from-repo <src> <id>`
Copy snapshots to another repo