

# restic

After some searching around i found restic as a solution.

It does file based backups has some sort of versioning, encrypts backups and restores permissions and ownership of data.

Following is how i set stuff up.

oh but dont forget ... [RTFM](#)

## Szenario:

- I do use a server for \$stuff
- I have a local Drive for Backups
- I Drive is mounted to the local computer reachable via the internet (in my case a pi, lol)

## Setup:

1. init repo: done on mashine to backup. basicly test if everything works on mashine storing the backup and init versioning/repo

1.1 init command

```
restic -r sftp://<USER>@<URL/IP>:<PORT>//media/HDD1/backups init
```

1.2 enter repo password: (you know note it down yada yada)

1.3 enter ssh key (i am using pubkey auth with encrypted ssh key)

1.4 if done correct you should see a success message ala `created restic repository <ID>`

2. (optional) alter ssh config for longer timeouts

2.1 open `~/.ssh/config`

2.2 enter or alter

```
ServerAliveInterval 60  
ServerAliveCountMax 240
```

3. setup to back up file

3.1 create file ala `include_from.txt`

3.2 specify folders to backup, for me following works:

```
/etc/**  
/var/**  
/home/**  
/root/**
```

3.3 create file ala `exclude_from.txt`

3.4 specify folders to exclude, for me worked following

```
.cache  
_cacache
```

# Backup:

1. Prepare Machine where backups are stored:

1.1 plug drive in

1.2 mount drive

```
sudo mount /dev/sda1 /media/HDD1/
```

2. Run Backup (run as root!)

2.1 run command

```
restic -r sftp://<USER>@<URL/IP>:<PORT>//media/HDD1/backups backup --files-from  
/root/include_from.txt
```

2.2 enter ssh password

2.3 enter repo password

---

Version #3

Erstellt: 2022-02-25 14:36:44 UTC von Hiajen

Zuletzt aktualisiert: 2023-03-12 10:29:34 UTC von Hiajen