

4. Datensicherung mit rsync

4.1 Beispiel

```
1 #!/bin/sh
2 tag=$(date +%A)
3 export DISPLAY=:0.0 &&
4 notify-send -t 120000 "Die automatische Sicherung der Daten wurde gestartet ...
5 Bis zur Fertigmeldung den Computer nicht herunterfahren!"
6 aplay $HOME/sound/success.wav
7 pg_dump handwerk > $HOME/HWDB/handwerk.sql
8 rsync --delete -a -v --exclude=.cache/ $HOME/ /media/daten/BACKUP/$tag
9 echo Sicherung ausgeführt am $tag den $(date +"%d.%m.%Y %H:%M:%S")
10 >> $HOME/AdminDok/Sicontrol.txt
11 sleep 5s
12 pkill -x xfce4-notifyd
13 sleep 2s
14 export DISPLAY=:0.0 &&
15 notify-send -t 30000 "Die tägliche Sicherung der Daten ist abgeschlossen!
16 Protokolliert im Ordner AdminDok, Datei Sicontrol.txt" &
17 aplay $HOME/sound/login.wav
18
```

Das Shell-Script **AISic** im Einzelnen:

#!/bin/sh »ist der einleitende Kommentar für ein Shell-Script.
tag=\$(date +%A) »hier wird der Variablen tag der aktuelle Wochentag zugewiesen
export DISPLAY=:0.0 && »ist notwendig, um auf die Grafikausgabe zu schreiben
**notify-send -t 120000 "Die automatische Sicherung der Daten wurde gestartet ...
Bis zur Fertigmeldung den Computer nicht herunterfahren!"**
 »weist das Meldungsprogramm notify an, 12 Sekunden diesen Text auszugeben
aplay \$HOME/sound/success.wav »die Soundausgabe aplay gibt den Sound wieder
pg_dump handwerk > \$HOME/HWDB/handwerk.sql »Datenbanksicherung ausführen
rsync --delete -a -v --exclude=.cache/ \$HOME/ /media/daten/BACKUP/\$tag
 »die eigentliche Sicherung synchronisiert \$HOME als Quelle mit dem Ziel
**echo Sicherung ausgeführt am \$tag den \$(date +"%d.%m.%Y %H:%M:%S") >>
\$HOME/AdminDok/Sicontrol.txt**
 »die Ausgabe hinter echo wird mit >> in die angegebene Datei geschrieben, das
 Umleitungszeichen > einfach überschreibt, das >> doppelte hängt den Text an.
sleep 5s »Wartezeit 5 Sekunden
pkill -x xfce4-notifyd »beendet das Meldungsfenster, falls noch aktiv
sleep 2s »Wartezeit 2 Sekunden
export DISPLAY=:0.0 && »siehe oben
**notify-send -t 30000 "Die tägliche Sicherung der Daten ist abgeschlossen!
Protokolliert im Ordner AdminDok, Datei Sicontrol.txt" &**
aplay \$HOME/sound/login.wav
 »wieder eine Ausgabe mit dem Meldungsprogramm notify, diesmal für 3 Sekunden
 und anschließend noch eine Soundausgabe

4.2 Anwendung von rsync

Die Syntax des Befehls im Terminal [\[3\]](#) sieht folgendermaßen aus:

```
rsync [OPTIONEN] <Quelle> <Ziel>
```

Anstelle des Platzhalters <Quelle> gibt man den Ort an aus dem gelesen werden soll und das <Ziel> bestimmt den Ort in den rsync zu kopieren hat.

4.3 Optionen

- -a fasst folgende Optionen zusammen:
 - -r kopiert Unterverzeichnisse
 - -l kopiert symbolische Links
 - -p behält Rechte der Quelldatei bei
 - -t behält Zeiten der Quelldatei bei,
 - -g behält Gruppenrechte der Quelldatei bei
 - -o behält Besitzrechte der Quelldatei bei (nur root)
 - -D behält Gerätedateien der Quelldatei bei (nur root)

Es ist empfehlenswert, die Option -a zu benutzen, um alle Rechte und Eigentümer der Quelldatei auf dem Zielmedium zu übernehmen (entspricht -r^lptgoD).

- -u überträgt nur die Dateien, die an der Quelle geändert wurden
- -v zeigt während des Synchronisierens alle Schritte die ausgeführt werden an
- -P aktiviert folgende Optionen:
 - --progress Fortschrittsanzeige beim Transfer anzeigen
 - --partial Fortsetzung des Transfers bei Abbruch
- -n simuliert nur was passieren würde ("dry run")
- --bwlimit z.B. '--bwlimit=30' - bwlimit limitiert die Bandbreite, die genutzt werden soll (Hilfreich, da rsync sonst die komplett verfügbare Bandbreite in Anspruch nimmt und sonstige Anwendungen damit blockiert)
- -z aktiviert die Komprimierung für die Datenübertragung (diese Option ist sinnvoll, wenn zwischen Quelle und Ziel eine langsame Verbindung besteht)
- -e damit lässt sich die remote shell auswählen, die meisten werden [SSH](#) nutzen also -e ssh

- `--exclude=Muster` schließt ein bestimmtes Muster von der Sicherung aus
- `--exclude=ordner1/ --exclude=ordner2/` schließt `ordner1` und `ordner2` von der Sicherung aus (bezieht sich immer auf `<Quelle>` und deren Unterordner!)
- `--delete` sorgt dafür, dass Dateien, die im Quellverzeichnis gelöscht wurden, auch im Ziel gelöscht werden
- `-b` sorgt dafür, dass durch die Option `--delete` gelöschte sowie alle veränderten Objekte gesichert werden
- mit `--backup-dir=Verzeichnis` kann man ein Verzeichnis dafür angeben
- `-c` sorgt dafür, dass zum Vergleich der Dateien Checksummen gebildet werden und nicht nur die Größe und der Timestamp verglichen werden. Diese Methode kann etwas länger dauern.
- `--iconv` sorgt für eine Konvertierung der Dateinamen zwischen Systemen mit verschiedenen Codepages. Dieser Parameter kann erforderlich werden, wenn Dateien mit z.B. Umlauten im Namen übertragen werden.
- `--stats` zeigt einen ausführlicheren Report am Ende einer Übertragung an.
- `--size-only` sorgt dafür, dass Dateien mit gleicher Dateigröße übersprungen werden, unabhängig davon, ob sie sich in anderen Eigenschaften unterscheiden.

4.4 Wichtige Hinweise

Vor dem Ausführen von `rsync` muss man sicher sein, die korrekten Parameter und Verzeichnisse angegeben zu haben. Fahrlässige Handhabung kann zu Datenverlust führen.

Kopiert man das gesamte [Homeverzeichnis](#), so werden auch private Daten, wie die privaten Schlüssel von [SSH](#) und [GnuPG](#) kopiert.

`rsync` kann nur das kopieren, wozu der Nutzer auch [Rechte](#) hat, z.B. bei

```
rsync -a /etc ziel
```

kann es zu Problemen kommen, was durch das vorangestellte `sudo` (Root-Rechte) vermieden wird.

```
sudo rsync -a /etc ziel
```

Vorsicht ist geboten, wenn man versucht, Konten wechselseitig zu sichern. Man sollte darauf achten, dass man nicht das Backup von A auf B wieder auf A kopiert.

Nicht zuletzt bedeutet die Option `--delete` wirklich delete! Sind in dem Zielverzeichnis andere Daten, werden sie gelöscht!

4.5 Beispiele zur Syntax

Bei allen Angaben zur Quelle und/oder des Ziels ist zu beachten, die slashes richtig zu setzen. Folgendes Beispiel soll dies verdeutlichen:

```
rsync -a /boot/grub /data/backup
```

erzeugt einen Unterordner **/grub** in **/data/backup**, weil *kein* slash hinter **/boot/grub** steht. Das heißt, dass die Daten nicht in **/data/backup** gespeichert werden sondern in **/data/backup/grub**.

Wird der Befehl so angewandt

```
rsync -a /boot/grub/ /data/backup
```

wird kein Unterordner beim Ziel erzeugt, da nun ein slash hinter **/boot/grub/** steht. Das heißt, dass alle Daten von **/boot/grub/** tatsächlich einzeln in das Verzeichnis **/data/backup** geschrieben werden. Merke: der Slash hinter einem Ordner steht für 'Inhalt'

4.6 Sicherung von lokalem Rechner auf entfernten Rechner

In diesem Beispiel soll ein lokales [Homeverzeichnis](#) (**/home/benutzer**) auf einen zweiten Rechner mit dem Namen `example.com` gesichert werden, zu dem man per [SSH](#) Zugang hat.

Dazu benötigt man die Option `-a`, damit alle Rechte erhalten bleiben. `-v` ist immer gut, damit man sieht, was passiert. Der Parameter `-z` verringert die Datenmenge, die übertragen werden muss, und `-e` wird benötigt, um `ssh` anzugeben. Also sieht der Befehl so aus:

```
rsync -avze ssh /home/benutzer benutzer@example.com:/backups
```

Dies würde eine Kopie von **/home/benutzer** vom lokalen Rechner zum entfernten Rechner ins Verzeichnis **/backups** kopieren.

4.7 Sicherung von entferntem Rechner auf lokalen Rechner

In diesem Beispiel soll das Verzeichnis **/var/www** vom Rechner `example.com` auf den lokalen Rechner gesichert werden. Außerdem sollen alle Dateien, die in **/var/www** gelöscht wurden, nachdem die letzte Sicherung erstellt wurde, lokal ebenfalls gelöscht werden, nachdem man sie noch ein letztes Mal sichert.

Es werden wieder die selben Optionen wie vorher genommen, sowie `--delete` für das Löschen und `-b` für Backups gelöschter und veränderter Dateien. Diese Backups befinden sich in dem mit `--backup-dir` angegebenen Verzeichnis.

Der Befehl sieht dann so aus:

```
rsync --delete -avzbe ssh benutzer@example.com:/var/www  
/home/benutzer/webserver --backup-dir=~/.old
```

Sollen dem SSH-Befehl weitere Optionen angehängt werden, z.B. `-i` um einen abweichenden Private-Key bei der Anmeldung zu übergeben, muss der SSH-Befehl in einfache Anführungszeichen gesetzt werden, etwa so:

```
rsync --delete -avzbe 'ssh -i /pfad/zur/id_rsa' benutzer@example.com:/var/www  
/home/benutzer/webserver --backup-dir=~/.old
```

Der Pfad zu dem Private-Key File muss dabei absolut sein. Dann den ganzen Befehl mit [Cron](#) zeitgesteuert automatisieren und die Angst vor einer defekten Festplatte oder versehentlich gelöschten Daten ist vorbei.

4.8 Sicherung innerhalb des Systems

Man kann neben dem Transfer über das Netzwerk `rsync` natürlich auch nutzen, um ein Backup auf eine andere interne oder **externe** (dies ist zu bevorzugen) Festplatte zu tätigen. Möchte man zum Beispiel alle Benutzerverzeichnisse sichern, so kann man dies mit

```
sudo rsync -av --progress --delete /home /media/Backup/
```

bewerkstelligen.

4.9 Verzeichnisse von der Sicherung ausschließen

Mit dem Parameter `--exclude` können Verzeichnisse von der Bearbeitung ausgeschlossen werden. Dabei ist zu beachten, dass die Quelle immer als Bezugspunkt interpretiert wird. Wenn also Daten vom Verzeichnis **/home/user/daten/** nach **/home/user/backup/** gesichert werden sollen und dabei das Verzeichnis **/home/user/daten/temp/** nicht berücksichtigt werden soll, muss der Befehl so aussehen:

```
rsync [OPTIONEN] --exclude=temp/ /home/user/daten/ /home/user/backup/
```

Das per `exclude` bezeichnete Verzeichniss ist also kein absoluter Pfad, sondern relativ zum `rsync` Verzeichniss `<Quelle>` zu sehen - hier in diesem Beispiel also `/home/user/daten/`.

Taucht der auszuschließende Ordner **temp** noch an weiteren Punkten der Quellhierarchie auf, so werden diese ebenfalls von der Sicherung ausgeschlossen.

Mehrere auszuschließende Ordner kann man auch in eine Datei speichern und dann im `rsync` Aufruf per `--exclude-from=FILE` übergeben.