2025/12/17 01:07 1/4 Backup mit rsync

Backup sind lästig, aber wenn sie funktinieren, hat man entgültig seine Ruhe.

Backup mit rsync

ich habe mir ein Script gebastelt, dass auf eine Funktion von rsync aufsetzt. Gibt man rsync ein Referenz-Verzeichnis, so macht es nur eine Kopie, wenn sich die Datei geändert hat. Bei Dateien ohne änderung macht er einen hardlink auf die alte Datei.

<u>Vorteil:</u> Inkrementelles Backup (Datengröße), aber trotzdem ein FullBackup <u>Nachteil:</u> Das Alter der Daten kann nicht bestimmt werden (mit find kann man alte Daten nicht löschen)

Deshalb wurde eine Datenbankdatei eingefügt, anhand derer die alten Daten gelöscht werden.

Skript



Es muss noch das mySQL-Passwort eingetragen werden

backup with database.sh

```
#!/bin/bash
backupdir="/backups_with_db/"
datum jetzt=`date +%F %H-%M-%S`
#Backup-Verzeichnis erstellen (falls es nicht vorhanden ist)
if [ ! -d $backupdir ]
then
    mkdir -p $backupdir
fi
#Aufruf: backup <Quelle> <Ziel>
#Idee des Gefechts:
#rsync hat die Option (--link-dest), nur die Aenderungen zu speichern
#Dazu muss man aber das zwie Verzeichnisse angeben, ein Backup-
Verzeichnis, und eins, in dem die alten daten liegen
#Wenn man am Ende des Syncs einen symbolischen Link legt (mit Namen
lastbackup), kann man beim naechsten sync diesen stischen Namen als
Basis verwinden
function backup(){
    src=$1
    dest=$2
```

https://wiki.da-checka.de/ Printed on 2025/12/17 01:07

```
for directory in `head -n -$age $path/$counter_file`
        #Loeschen des Verzeichnisses
        #echo -e $directory
        rm -r $directory
    done
    #Loeschen der Verzeichnisse aus der counter-Datei selbst
    cp $path/$counter file $path/dummy
    tail -n $age $path/dummy > $path/$counter file
    rm $path/dummy
    }
#Aufruf: backup database <Datenbank> <Zielverzeichnis>
#Funktion, um eine Datenbank zu sichern. Dabei kann man die maximale
Anzahl an Backups angeben
backup database(){
    database=$1
    dest=$2
    #Sollte Verzeichnis nicht existieren, wird es angelegt
    if [ ! -d $dest ]
    then
        mkdir -p $dest
    fi
    #Backup der Datenbank
    mysqldump --host=localhost --user=root --password=<Passwort> --
databases --add-drop-database $database | bzip2 -9 >
$dest/$database $datum jetzt.txt.bz2
    #mysqldump --host=localhost --user=root --password=<Passwort> --
databases --add-drop-database $database >
$dest/$database $datum jetzt.txt.bz2
    #Counter fuer loeschung setzen
    echo "$dest/$database $datum jetzt.txt.bz2" >>
$dest/database counter.log
echo -e "Backup owncloud"
#Datenverzeichnisse werden gesichert
backup "/var/www/owncloud/" "$backupdir/owncloud/"
#Alle File-Backups, die älter als x sind, werden geloescht
```

```
delete_old "$backupdir/owncloud/" "backup_counter.log"
#Datenbank wird gesichert
backup_database owncloud "$backupdir/owncloud"
#alle Datenbank-Backups, die aelter als x sind, werden geloescht
delete_old "$backupdir/owncloud/" "database_counter.log"
```

From:

https://wiki.da-checka.de/ - PSwiki

Permanent link:

https://wiki.da-checka.de/doku.php/wiki/skripte/backup_with_hardlinks





https://wiki.da-checka.de/ Printed on 2025/12/17 01:07