

Backup sind lästig, aber wenn sie funktionieren, 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.

Vorteil: Inkrementelles Backup (Datengröße), aber trotzdem ein FullBackup

Nachteil: 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 Änderungen 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 verwenden

function backup(){
    src=$1
    dest=$2
```

```
#Sollte Verzeichnis nicht existieren, wird es angelegt
if [ ! -d $dest ]
then
    mkdir -p $dest
fi

#temporaeres Verzeichnis fuer den ersten sync
if [ `ls $dest | wc -l` = 0 ]
then
    mkdir $dest/temp
    ln -s $dest/temp $dest/lastbackup
fi

#erstellen des aktuellen Backu-Verzeichnisses
mkdir -p $dest/$datum_jetzt

#Schreiben des Counters (fuer etwaige Loeschungen)
echo "$dest/$datum_jetzt" >> $dest/backup_counter.log

##### Eigentlicher Sync (Backup) #####
rsync -a --link-dest=$dest/lastbackup $src $dest/$datum_jetzt

#Link vom alten Backup loesen und auf aktuelles Verzeichnis setzen
unlink $dest/lastbackup
ln -s $dest/$datum_jetzt $dest/lastbackup

#Das temporaere Verzeichnis wird (falls vorhanden) geloescht
if [ -d $dest/temp ]
then
    rm -r $dest/temp
fi

}

#Aufruf: delete_old <Pfad> <counterfile>
#Idee des Gefechts
#im Pfad befindet sich eine Datei mit den Namen aller Backups.
#es werden <age> Stueck behalten. Alles was vorher war, wird geloescht
#Vorsicht!!!: Age heist nicht Alte in Tagen!                      Age / Aufrufe pro
Tag = Tage

function delete_old(){
    #Anzahl der Backups, die Auf der Platte bleiben Default=30
    age=180
    path=$1
    counter_file=$2

    # Verzeichnisse aus backup_counters.log suchen, die aelter als <age>
    sind und dann nach und nach loeschen
```

```
for directory in `head -n -$age $path/$counter_file`
do
    #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 geloesch
```

```
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

Last update: **2015/07/10 19:39**

