Backup

Rsnapshot

ist ein einfach zu bedienendes Backupprogramm. Rsnapshot hasiert auf rsync, es ist einfach zu bedienen und eignet sich sehr gut für Backups auf USB Festplatten.
ein anderes einfaches Backup-Skript

```
#!/bin/sh
NOW=$(date +"%d-%m-%Y")
# set mysql login info
MUSER="MySQL-UserNAME"                 # Username
MPASS="MySQL-SERVER-PASSWORD"   # Password
MHOST="MySQL-SERVER-IP-ADDRESS"  # Server Name
# guess binary names
MYSQL="$(which mysql)"
MYSQLDUMP="$(which mysqldump)"
GZIP="$(which gzip)"

# get all db names
DBS="$($MYSQL -u $MUSER -h $MHOST -p$MPASS -Bse 'show databases')"
for db in $DBS
do
 FILE=mysql-$db.$NOW-$(date +"%T").gz
 $MYSQLDUMP -u $MUSER -h $MHOST -p$MPASS $db | $GZIP -9> $FILE
done
```

Verzeichnisse sichern

```
#!/bin/bash
# Simple backup with rsync
# local-mode, tossh-mode, fromssh-mode

SOURCES=(/root /etc /home /boot /lampp)
TARGET="/media/backup"

# edit or comment with "#"
#LISTPACKAGES=listdebianpackages        # local-mode and tossh-mode
MONTHROTATE=monthrotate                 # use DD instead of YYMMDD

RSYNCCONF=(--delete)
#MOUNTPOINT="/media/daten"               # check local mountpoint
#MAILREC="user@ domain"

#SSHUSER="sshuser"
#FROMSSH="fromssh-server"
```

```
#TOSSH="tossh-server"
SSHPORT=22

### do not edit ###

MOUNT="/bin/mount"; FGREP="/bin/fgrep"; SSH="/usr/bin/ssh"
LN="/bin/ln"; ECHO="/bin/echo"; DATE="/bin/date"; RM="/bin/rm"
DPKG="/usr/bin/dpkg"; AWK="/usr/bin/awk"; MAIL="/usr/bin/mail"
CUT="/usr/bin/cut"; TR="/usr/bin/tr"; RSYNC="/usr/bin/rsync"
LAST="last"; INC="--link-dest=$TARGET/$LAST"

LOG=$0.log
$DATE> $LOG

if [ "${TARGET:${#TARGET}-1:1}" != "/" ]; then
  TARGET=$TARGET/
fi

if [ "$LISTPACKAGES" ] && [ -z "$FROMSSH" ]; then
  $ECHO "$DPKG --get-selections | $AWK '!/deinstall|purge|hold/'|$CUT -f1 |
$TR '\n' ' '">> $LOG
  $DPKG --get-selections | $AWK '!/deinstall|purge|hold/'|$CUT -f1 |$TR '\n'
' '>> $LOG  2>&1
fi

if [ "$MOUNTPOINT" ]; then
  MOUNTED=$($MOUNT | $FGREP "$MOUNTPOINT");
fi

if [ -z "$MOUNTPOINT" ] || [ "$MOUNTED" ]; then
  if [ -z "$MONTHROTATE" ]; then
    TODAY=$($DATE +%y%m%d)
  else
    TODAY=$($DATE +%d)
  fi

  if [ "$SSHUSER" ] && [ "$SSHPORT" ]; then
    S="$SSH -p $SSHPORT -l $SSHUSER";
  fi

  for SOURCE in "${SOURCES[@]}"
    do
      if [ "$S" ] && [ "$FROMSSH" ] && [ -z "$TOSSH" ]; then
        $ECHO "$RSYNC -e \"$S\" -avR \"$FROMSSH:$SOURCE\" ${RSYNCCONF[@]}
$TARGET$TODAY $INC">> $LOG
        $RSYNC -e "$S" -avR "$FROMSSH:\"$SOURCE\"" "${RSYNCCONF[@]}"
"$TARGET"$TODAY $INC>> $LOG 2>&1
        if [ $? -ne 0 ]; then
          ERROR=1
        fi
      fi
```

```
      if [ "$S" ]  && [ "$TOSSH" ] && [ -z "$FROMSSH" ]; then
        $ECHO "$RSYNC -e \"$S\" -avR \"$SOURCE\" ${RSYNCCONF[@]}
\"$TOSSH:$TARGET$TODAY\" $INC ">> $LOG
        $RSYNC -e "$S" -avR "$SOURCE" "${RSYNCCONF[@]}"
"$TOSSH:\"$TARGET\"$TODAY" $INC>> $LOG 2>&1
        if [ $? -ne 0 ]; then
          ERROR=1
        fi
      fi
      if [ -z "$S" ]; then
        $ECHO "$RSYNC -avR \"$SOURCE\" ${RSYNCCONF[@]} $TARGET$TODAY $INC">>
$LOG
        $RSYNC -avR "$SOURCE" "${RSYNCCONF[@]}" "$TARGET"$TODAY $INC>> $LOG
2>&1
        if [ $? -ne 0 ]; then
          ERROR=1
        fi
      fi
  done

  if [ "$S" ] && [ "$TOSSH" ] && [ -z "$FROMSSH" ]; then
    $ECHO "$SSH -p $SSHPORT -l $SSHUSER $TOSSH $LN -nsf $TARGET$TODAY
$TARGET$LAST">> $LOG
    $SSH -p $SSHPORT -l $SSHUSER $TOSSH "$LN -nsf \"$TARGET\"$TODAY
\"$TARGET\"$LAST">> $LOG 2>&1
    if [ $? -ne 0 ]; then
      ERROR=1
    fi
  fi
  if ( [ "$S" ] && [ "$FROMSSH" ] && [ -z "$TOSSH" ] ) || ( [ -z "$S" ] );
then
    $ECHO "$LN -nsf $TARGET$TODAY $TARGET$LAST">> $LOG
    $LN -nsf "$TARGET"$TODAY "$TARGET"$LAST>> $LOG 2>&1
    if [ $? -ne 0 ]; then
      ERROR=1
    fi
  fi
else
  $ECHO "$MOUNTPOINT not mounted">> $LOG
  ERROR=1
fi
$DATE>> $LOG
if [ -n "$MAILREC" ]; then
  if [ $ERROR ];then
    $MAIL -s "Error Backup $LOG" $MAILREC <$LOG
  else
    $MAIL -s "Backup $LOG" $MAILREC <$LOG
  fi
fi
```

Rsnapshot-Tagessicherung mit Überwachung und Fehlermail

Zusammenfassung

Diese Lösung kombiniert:

* Tagessicherung
* Überwachung
* Logging
* Fehlermail
* Schutz vor Doppelstarts

in **einem einzigen Skript**.

Daher reicht **ein einziger Cronjob** völlig aus.