

Automatisches - Server Backup Skript

Das folgende Skript, wurde erstellt, um wichtige Dateien eines Applikationsservers in regelmässigen Abständen automatisiert zu sichern. Zur Installation, wird das Skript immer jeweils auf das zu sichernde System kopiert und dort wie unten beschrieben eingerichtet.

Installation

Vor der Installation, muss sichergestellt werden, dass bereits ein funktionierender NFS-Server (Zur Speicherung des Backups) zur Verfügung steht. Die Server Adresse und die NFS-Freigabe muss anschliessend im Skript angepasst werden.

```
# vim /root/server_backup_script.sh
```

[PASTE THE SCRIPT SOURCECODE]

Nun wird das Backupskript zur Ausführung berechtigt und anschliessend im **crontab** der zu sichernden Maschine eingetragen. In meinem Beispiel, wird hier immer **täglich** um **12:00** das Skript automatisiert ausgeführt.

```
# chmod +x /root/server_backup_script.sh
```

```
# vim /etc/crontab
```

```
0 12 * * * root /root/server_backup_script.sh
```

Skript Sourcecode

Filename: **server_backup_script.sh**

```
#!/bin/bash
#####
#####
***** Backup Linux Server Script by Michael Reber - v 1.0
*****#
#####
#####

#####
#####
```

```
## Variable Definition & System Vorbereitungen:
backupdir=`hostname`
today=`date +%Y-%m-%d`
OLDBACKUP=`date -d "7 days ago" +%Y-%m-%d`

## System dependency Check:
if [ -n "$(command -v apt-get)" ]; then
    if [ $(dpkg-query -W -f='${Status}' nfs-common 2>/dev/null | grep -c "ok
installed") -eq 0 ]; then
        apt-get install nfs-common -y;
    fi
else
    if [ $(yum -q list installed nfs-utils &>/dev/null && echo "1" || echo
"0") -eq 0 ]; then
        yum install nfs-utils -y;
        systemctl start rpcbind && systemctl enable rpcbind;
    fi
fi
if [ ! -d "/mnt/server-backup" ]; then
    mkdir /mnt/server-backup
fi
#####
#####
## Mount und bereitstellen des Backupverzeichnis:
mount -t nfs -o rw,hard 192.168.1.21:/volumel/server-backups /mnt/server-
backup 2>> /var/log/blackSERV-backup.log

if [ ! -d "/mnt/server-backup/$backupdir" ]; then
    mkdir /mnt/server-backup/$backupdir
    echo "$today - Creating new backupdir: $backupdir for this server.." >>
/var/log/blackSERV-backup.log
fi
#####
#####
## Start des System-Backups:
echo "$today - Starting with backup for server: $backupdir .." >>
/var/log/blackSERV-backup.log

mkdir -p /mnt/server-backup/$backupdir/$today/etc
cp /etc/fstab /mnt/server-backup/$backupdir/$today/etc/
cp /etc/crontab /mnt/server-backup/$backupdir/$today/etc/
cat ~/.bash_history | grep -v '#' >> /mnt/server-
backup/$backupdir/$today/history.txt

# Backup Webserver Content..
if [ -d "/var/www" ]; then
    mkdir /mnt/server-backup/$backupdir/$today/webfiles
    cp -r /var/www/* /mnt/server-backup/$backupdir/$today/webfiles/
fi
```

```
# Backup Webserver Configuration..
if [ -d "/etc/apache2" ]; then
    mkdir /mnt/server-backup/$backupdir/$today/etc/apache2
    cp -r /etc/apache2/* /mnt/server-backup/$backupdir/$today/etc/apache2/
fi
if [ -d "/etc/httpd" ]; then
    mkdir /mnt/server-backup/$backupdir/$today/etc/httpd
    cp -r /etc/httpd/conf /etc/httpd/conf.d /etc/httpd/conf.modules.d
/mnt/server-backup/$backupdir/$today/etc/httpd/
fi

# Backup /opt Verzeichnis
if [ -d "/opt" ]; then
    mkdir /mnt/server-backup/$backupdir/$today/opt
    cp -r /opt/* /mnt/server-backup/$backupdir/$today/opt/
fi

# Dump database into SQL file
#mysqldump --user=$user --password=$password --host=$host $db_name >
/mnt/server-backup/$backupdir/$today/$db_name-$today.sql

#####
#####
## Löschen von 7 Tage alten Backups:
if [ -d "/mnt/server-backup/$backupdir/$OLDBACKUP" ]; then
    rm -fR /mnt/server-backup/$backupdir/$OLDBACKUP
fi
#####
#####
##
umount /mnt/server-backup
```

Last update: **2018/01/08 13:25**