Datensicherung für Faule

Das Backup-Karussell - Reloaded

Hans-Jürgen Beie
<hjb@pollux.franken.de>

Motivation

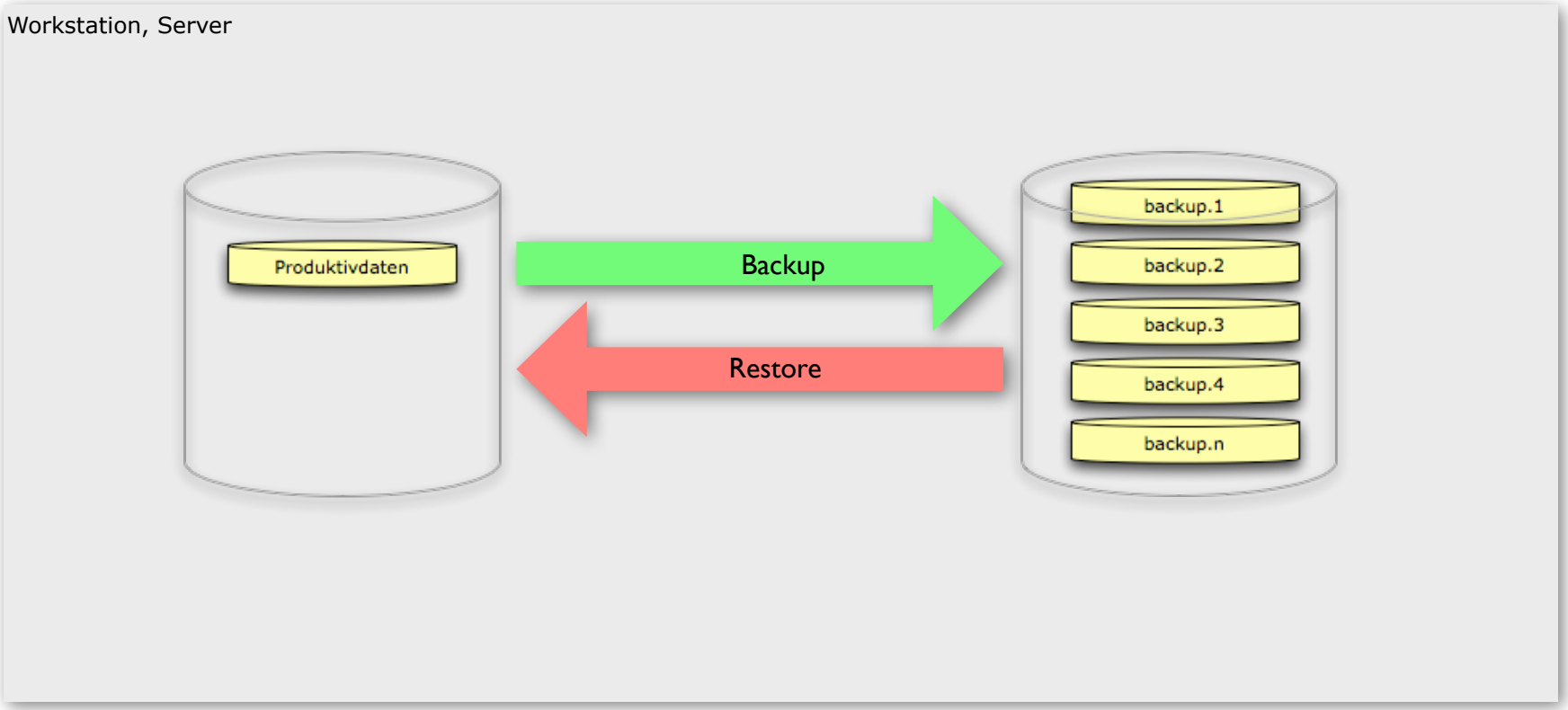
Was wollen wir?

- regelmäßige, automatische Sicherung von Daten (täglich, wöchentlich, ...)
- schneller Datentransfer auf preiswertes Medium (Festplatte)
- Verwendung von Standard-Software
- Sicherung für Server, Workstations, Desktop-Rechner oder mobile Systeme
- flexible Konfiguration für automatische, zeitgesteuerte Backups
- einfache Datenwiederherstellung: Backup an sich ist gut, einfache Restauration ist genau so wichtig!
- Datensicherung für „Faule"

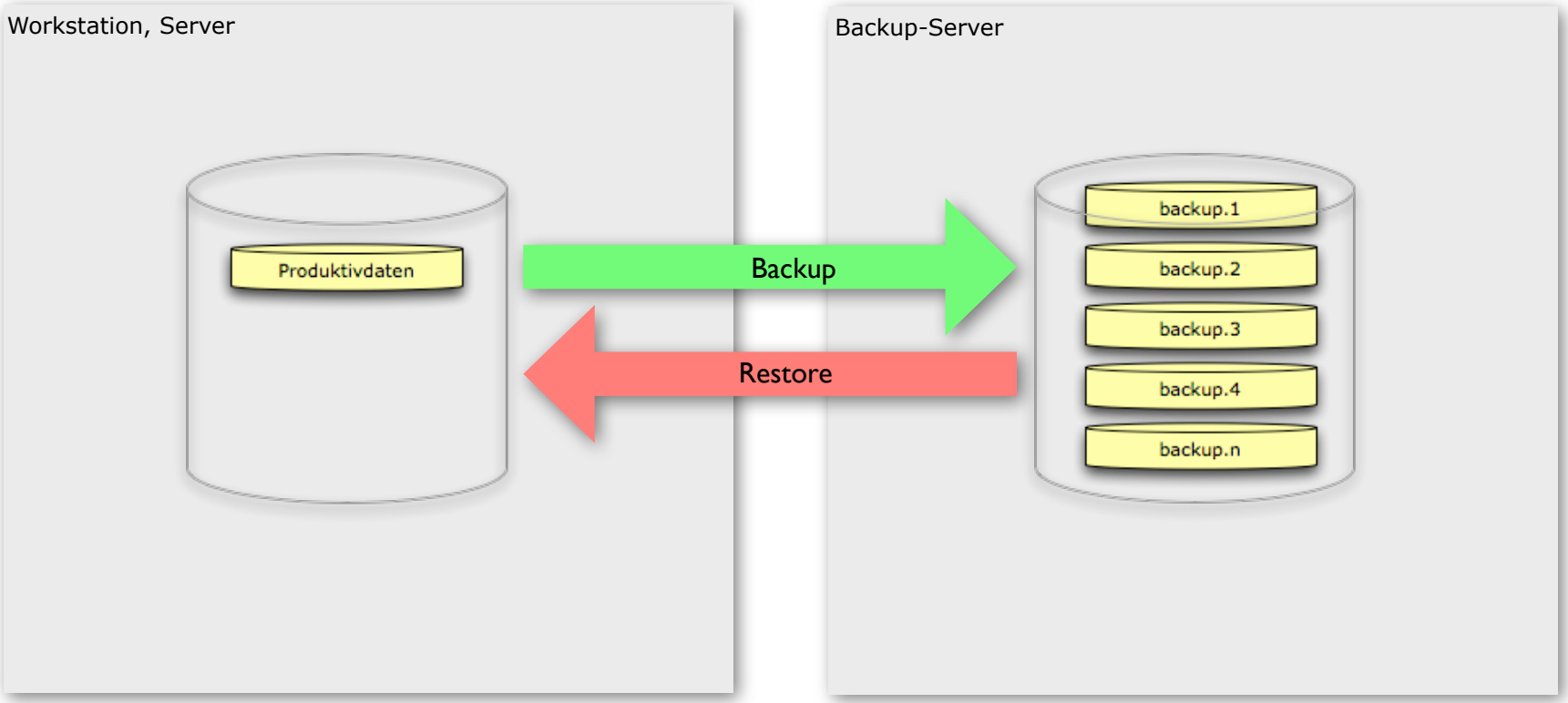
Was wollen wir nicht?

- Langzeitarchivierung
- Versionsmanagement
- spezielle Software/Hardware
- das Rad neu erfinden

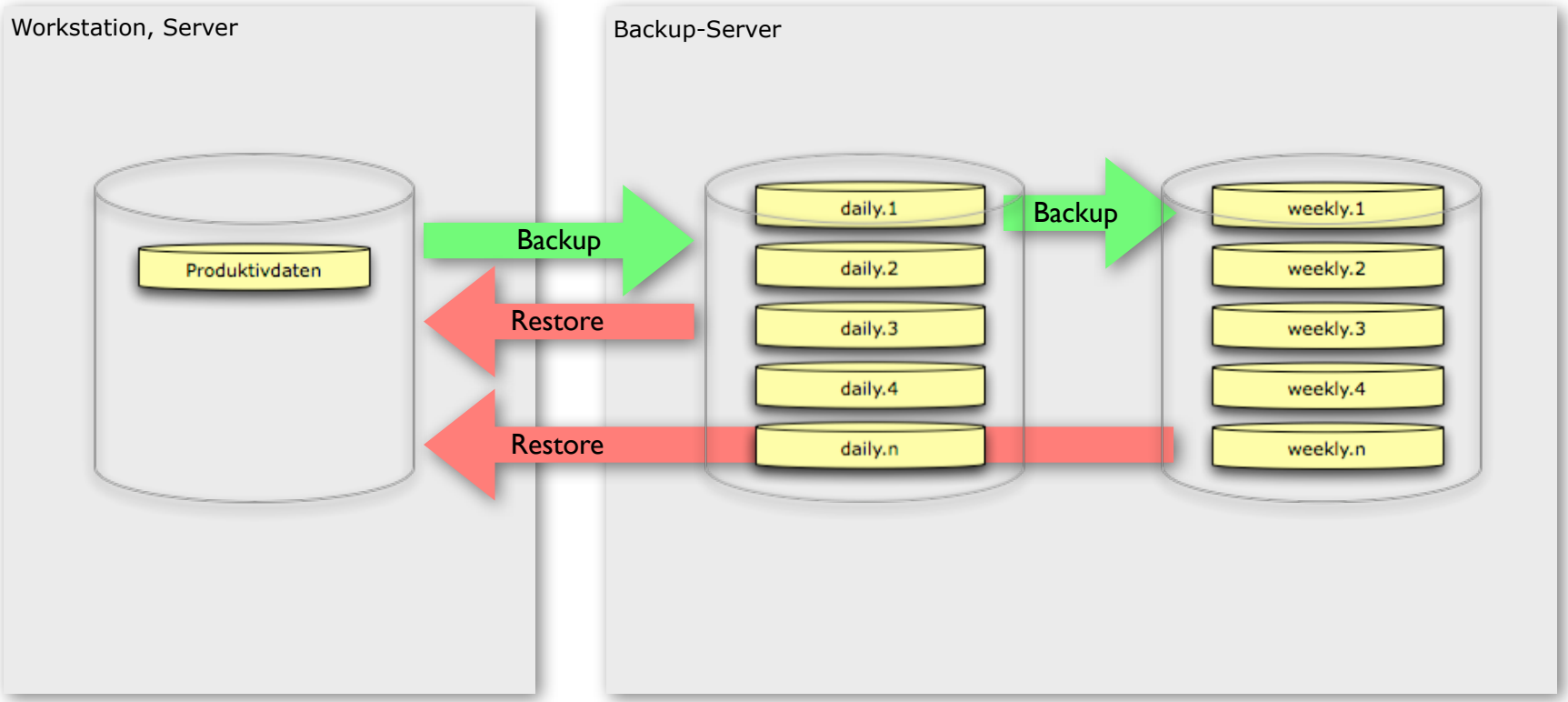
Szenario 1 — Rechner mit zweiter Backup-Platte



Szenario 2 — Zwei Rechner: Produktivsystem und Backup-Server



Szenario 3 — Zwei Rechner: Produktivsystem und Server mit zwei Backup-Platten



Grundlagen — hard links

Was ist ein „hard link“?

- Eine Datei wird im File-System durch einen Block an Informationen - den so genannten **inode** - repräsentiert.
- Ein **hard link** ist ein Zeiger auf eine Datei. Ein hard link zeigt auf einen inode und gilt nur innerhalb ein und desselben File-Systems (Partition).
- Jeder Dateiname ist ein hard link.

Platz sparen mit hard links

- Mit dem Kommando
  `ln backup.1/kongress07.knf backup.1/kongress07.knf`
  wird der Datei `backup.2/kongress07` ein neuer (zusätzlicher) Bezeichner `backup.1/kongress07.knf`, also ein weiterer hard link, zugewiesen.
- Das ist physikalisch immer noch **eine** Datei, nur ist sie mit dem gleichen Namen in zwei unterschiedlichen Verzeichnissen verfügbar.

Kopieren mit hard links

- Mit dem `cp`-Kommando (mit der GNU-Variante!) kann man „virtuelle Duplikate“ von ganzen Verzeichnisbäumen anlegen:
  `cp -al /archive/backup.1 /archive/backup.0`

Grundlagen — rsync

Wozu ist rsync gut?

- Daten in einem Zielverzeichnis rekursiv mit den Daten aus einem Quellverzeichnis synchronisieren:
  ```
  rsync -al --delete source_dir destination_dir/
  ```
  Dabei werden nur die Dateien kopiert, die im Zielverzeichnis nicht vorhanden sind oder sich von denen im Zielverzeichnis unterscheiden. Dateien, die im Zielverzeichnis vorhanden sind aber keine Gegenstücke auf Seiten der Quelle haben, werden gelöscht.

- Das Ganze funktioniert auch zwischen zwei Rechnern. So lässt sich ein lokales Verzeichnis mit dem Inhalt eines Verzeichnisses auf einem anderen Rechner synchronisieren:
  ```
  rsync -al --delete remote.host:/source /backup/
  ```

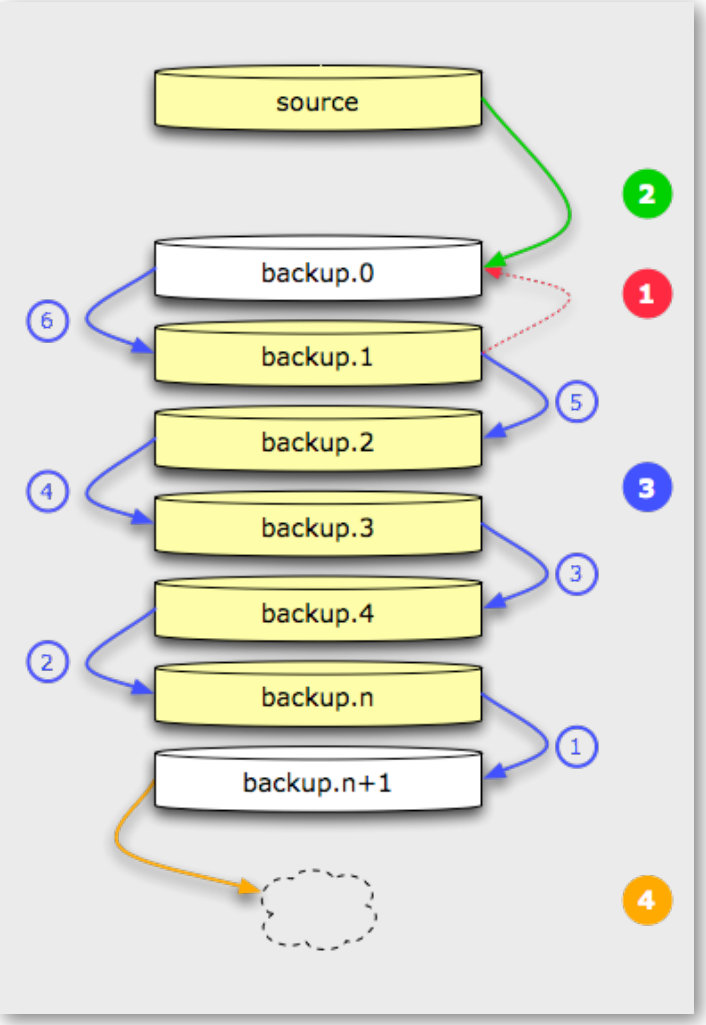
rsync und hard links

Neuere Versionen von **rsync** können Verknüpfungen mit **hard links** auf der Zielseite selbst erzeugen:

- Daten im Zielverzeichnis `backup.0` mir denen im Quellverzeichnis `source` auf einem anderen Rechner synchronisieren und dabei automatisch hard links zu gleichen (unveränderten) Dateien in einem zweiten Referenzverzeichnis `backup.1` anlegen:
  ```
  rsync -al --delete --link-dest=/arc/backup.1 remote.host:/source /arc/backup.0/
  ```

- Das ist äquivalent zu:
  ```
  cp -al /arc/backup.1 /arc/backup.0
  rsync -al --delete remote.host:/source /arc/backup.0/
  ```

Datensicherung mit rsync und hard links — Funktionsschema



Sicherung von Produktivdaten in einem „Karussell" von n rotierenden Backups

1. Mit hard links eine temporäre Kopie der letzten Sicherung erzeugen:

   ```
   cp -al backup.1 backup.0
   ```

   Wenn rsync die Option `--link-dest` unterstützt (siehe 2 a), kann dieser Schritt entfallen.

2. Inhalt von `backup.0` mit den Quelldaten synchronisieren:

   (a) `rsync -al --delete source backup.0/`

   Wenn rsync die Option `--link-dest` unterstützt, und Schritt 1 ausgelassen wurde, wird das Link-Verzeichnis mit angegeben:

   (b) `rsync -al --delete --link-dest=backup.1 source backup.0/`

3. Backup-Verzeichnisse durch Umbenennen rotieren:

   ```
   mv backup.n backup.n+1
   ...
   mv backup.1 backup.2
   mv backup.0 backup.1
   ```

4. Überzähliges Verzeichnis entfernen:

   ```
   rm -f backup.n+1
   ```

Systemvoraussetzungen und Werkzeugkasten

Was braucht man dazu?

- Backup-Server: Betriebssystem auf Unix-Basis, z.B. Linux, xBSD, Mac OS X
- GNU File Utilities: Bei BSD-basierten Systemen extra installieren!
  http://www.gnu.org/software/fileutils/fileutils.html
- rsync: Bei Unixen meistens schon im Bordwerkzeug
  http://rsync.samba.org
- Windows als Produktivsystem: cygwin, cwRsync
  http://cygwin.com, *http://www.itefix.no/cwrsync*
- Root/Admin-Rechte für Quelle und Ziel, wenn alle Dateittribute restaurierbar sein sollen.

Damit sich das Karusell auf dem Backup-Server möglichst von alleine dreht

- Der Ausgangspunkt von vielen Lösungen: Shell-Skripte von Mike Rubel (proof of concept)
  http://www.mikerubel.org/computers/rsync_snapshots/
- rsnapshot (Perl)
  http://rsnapshot.org
- Shell-Skripte von Heinlein
  http://www.heinlein-support.de/web/support/wissen/rsync-backup/
- rsback (Perl)
  http://www.pollux.franken.de/hjb/rsback/

! *Unvollständige Auswahl*

rsback — ?

Warum?

Aufteilen von Backups in kleinere „Happen“ ist ratsam, weil

- rsync sehr ressourcenhungrig ist
- der Zugriff auf das „backup repository“ u.U. mit unterschiedlichen Rechten geregelt werden muss
- bei Backup-Fehlern der „Schaden“ begrenzt werden soll
- bei (remote) Backups u.U. mehrere Anläufe notwendig sind

Aufteilen bedeutet aber: viele Skripte erstellen und warten.

rsback ...

- bietet für typische, immer wiederkehrende Aufgaben eine einfach zu konfigurierende Verwaltung
- hilft beim Aufteilen von umfangreichen Backups in handliche Pakete (→ tasks)
- startet und überwacht rsync bei den verschiedenen Aufgaben
- dreht das „Karussell“
- sorgt bei Problemen für die Schadensbegrenzung
- liefert kompakte Logs für vielbeschäftigte Admins

rsback

besteht aus ...

- einem Perl-Skript
- einer Konfigurationsdatei (oder auch mehreren)
- optionalen „exclude lists“ für rsync, um Verzeichnisse oder Dateien vom Backup auszuschließen

wird gestartet ...

- über die Konsole

```
usage: rsback [options] backup-tasks(s)
Make a backup of one or more backup tree(s). The backup tasks itself
are described in the configuration file
   /etc/rsback/rsback.conf

options
-h              Display this help
-v              Be verbose
-d              Debug mode (more verbose, runs rsync with option --dry-run)
-i              Init backup repositories only
-c conf-file    Configuration file other than /etc/rsback/rsback.conf
```

- oder über cron jobs

```
11 01 * * 1-7 /usr/local/sbin/rsback -v work-daily  >>/var/log/rsback/work-daily.log
11 05 * * 7   /usr/local/sbin/rsback -v work-weekly >>/var/log/rsback/work-weekly.log
```

rsback — Konfiguration

Gobale Parameter für rsback

In der Sektion `[global]` werden globale Parameter festgelegt, hier die wichtigsten:

Wo sind welche Programme zu finden?

Welche Aufgaben (tasks) gibt es?

Verzeichnis für Lock-Files

Optionen für rsync

Ausschlussliste

Soll rsync selbst hard links anlegen?

Welche „Fehler", die rsync meldet, sind weniger dramatisch?

! *Es gibt noch einige andere Parameter mehr.*

im Stil von Windows INI-Dateien oder rsyncd.conf

```
# rsback.conf
# ---------------------------------
[global]
rsync_cmd = /usr/bin/rsync
cp_cmd = /bin/cp
mv_cmd = /bin/mv
rm_cmd = /bin/rm
mkdir_cmd = /bin/mkdir

tasks = work-daily work-weekly

lock_dir = /var/lock

rsync_options = -al --delete \
   --delete-excluded --numeric-ids

exclude_file = /etc/rsback/rsback.exclude

use_link_dest = yes

ignore_rsync_errors = 24

# ---------------------------------
[work-daily]

...
```

Die tägliche Arbeit: task im rsync-Modus

Für jede Aufgabe (task) muss in einer Sektion festlegt werden, was zu tun ist.

Was soll gesichert werden?

In welches Verzeichnis soll gesichert werden?

rsync soll verwendet werden

Wir wollen 14 Archive, die mit `daily.1` bis `daily.14` benannt sein sollen.

Ausschlussliste

Optionen für rsync
Hier etwas üppiger bestückt, weil wir per ssh über eine DSL-Verbindung mit Bandbreiten-begrenzung transferieren wollen.

Damit uns dieser Job nicht in die Quere kommt, sperren wir ihn.

! *Parameter in `[<task>]` können Parameter in `[global]` überschreiben.*

```
...
# ---------------------------------
[work-daily]

source = rsback@remote.host:/var/work/

destination = /BACKUP/remote-host


mode = rsync

rotate = daily 14


exclude_file = /etc/rsback/work.exclude

rsync_options = -al --safe-links --delete\
    --delete-excluded --stats \
    --numeric-ids --bwlimit=48 \
    -e "ssh -i rsback.id_rsa" \
    --rsync-path='sudo rsync'

lock = work-weekly

# ---------------------------------
[work-weekly]
...
```

Einmal pro Woche: task im link-Modus zum Fixieren der letzten Tagessicherung

Für jede Aufgabe (task) muss in einer Sektion festlegt werden, was zu tun ist.

Wir „duplizieren" die letzte Tagessicherung ...

... ins gleiche Verzeichnis. (dort wird daraus **`weekly.1`**)

Diesmal einfach mit hard links, dazu brauchen wir rsync nicht.

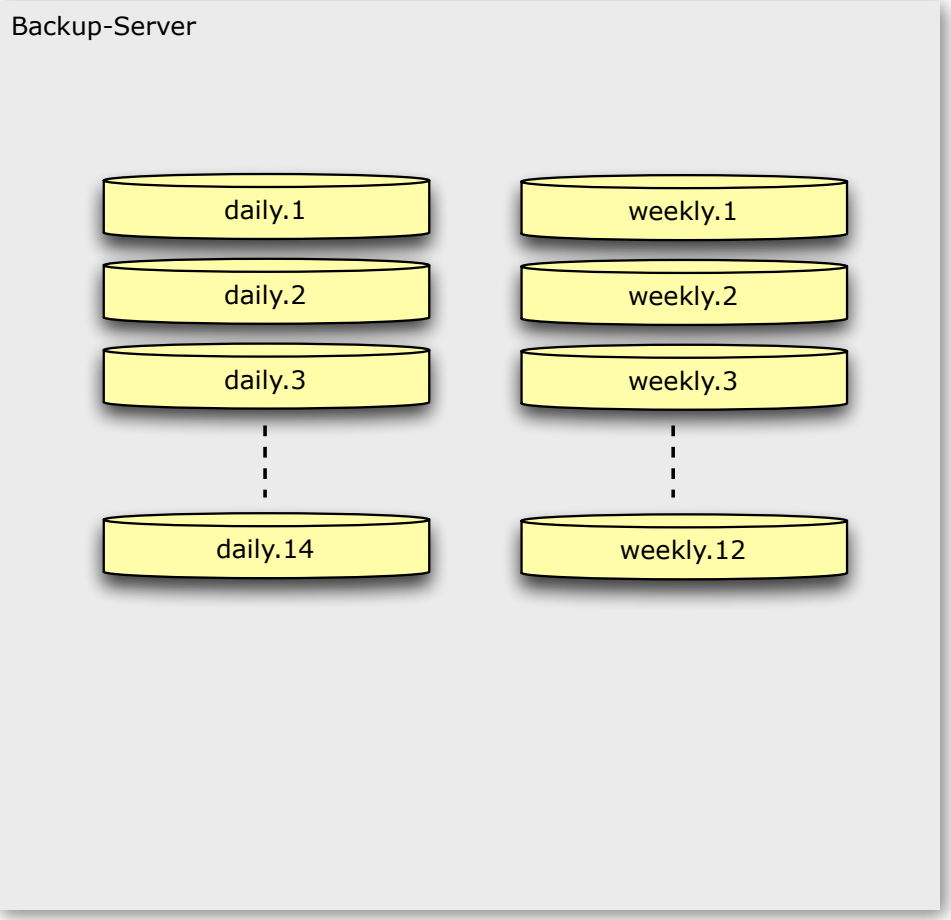
Wir heben 12 Archive auf, die mit **`weekly.1`** bis **`weekly.12`** benannt sind.

*! Archive müssen nicht **`daily`** oder **`weekly`** heißen. Der Name ist frei wählbar.*

```
...
# ---------------------------------
[work-weekly]

source = /BACKUP/remote-host/daily.1/

destination = /BACKUP/remote-host

mode = link

rotate = weekly 12
```

12 Wochen später ...

```
# tree /BACKUP/remote-host/work -L 1
/BACKUP/remote-host/work
|-- daily.1
|-- daily.10
|-- daily.11
|-- daily.12
|-- daily.13
|-- daily.14
|-- daily.2
|-- daily.3
|-- daily.4
|-- daily.5
|-- daily.6
|-- daily.7
|-- daily.8
|-- daily.9
|-- history.daily
|-- history.weekly
|-- weekly.1
|-- weekly.10
|-- weekly.11
|-- weekly.12
|-- weekly.2
|-- weekly.3
|-- weekly.4
|-- weekly.5
|-- weekly.6
|-- weekly.7
|-- weekly.8
`-- weekly.9
```



rsback — Daten wieder restaurieren

Der GAU:

```
hjb@pollux:/work$ ls knf/kongress-2007
ls: knf/kongress-2007: No such file or directory
```

Ich habe vor ungefähr drei Wochen meine Daten zerstört!
Was nun?

Mit rsync wieder zurück holen:

```
# rsync -al -e ssh \
        backup.server:/BACKUP/work/weekly.3/knf/kongress-2007 /work/knf/
```

Wenn die Archive lokal verfügbar sind, geht es mit einer einfachen Kopieraktion:

```
# cp -a /BACKUP/work/weekly.3/knf/kongress-2007 /work/knf/
```

Reloaded!

rsback — Datensicherung eines Live-Systems mit minimierter Totzeit

Shell-Skript zum Sichern eines VSevers

```
#!/bin/sh
# backup vserver ldap2
# ------------------------------------------
# backup running vserver
/usr/local/sbin/rsback -v ladap2-daily

# ------------------------------------------
# shutdown vserver
/usr/sbin/vserver ldap2 running >>/dev/null
VS=$?
if [ $VS -eq 0 ] ; then
    echo "stopping vserver ldap2 ..."
    /usr/sbin/vserver ldap2 stop
fi

# ------------------------------------------
# backup again
/usr/local/sbin/rsback -v ldap2-daily-update

# ------------------------------------------
# start vserver
if [ $VS -eq 0 ] ; then
    echo "starting vserver ldap2 ..."
    /usr/sbin/vserver ldap2 start
    #check if it is running now
    /usr/sbin/vserver ldap2 running >>/dev/null
    VS=$?
    if [ $VS -ne 0 ] ; then
        echo "restart of vserver ldap2 failed!"
    fi
fi
```

Tasks in der rsback-Konfiguration

```
# rsback.conf
# backup of vserver ldap2
#-------------------------------------------
# first backup: server runs
[ldap2-daily]

source = /vservers/ldap2/
destination = /BACKUP/vservers/ldap2
mode = rsync
rotate = daily 10
rsync_options = -al --delete --stats \
                --numeric-ids
exclude_file = /etc/rsback/vservers.exclude
use_link_dest = yes

#-------------------------------------------
# second backup: server is stopped
# update task [ldap2-daily] without rotation
[ldap2-daily-update]
update = ldap2-daily
```

! Die Option **`update = <task>`** wirkt wie ein „normales“ **`rsync`**-Kommando auf das aktuelle Backup-Archiv

rsback — Datensicherung einer Mac OS X Workstation

Shell-Skript zum Datensichern und Runterfahren

```
#!/bin/sh
# Backup system and user data on OS X
# ----------------------------------------------
# Programs and config
RSYNC="/usr/bin/rsync"
RSBACK="/Applications/Backup/rsback.pl"
RSBACKCONF="/Applications/Backup/rsback.conf"

# ----------------------------------------------
# Mirror system partition
SOURCE="/"
DEST="/Volumes/Mirror"
EXCLUDE="/Applications/Backup/mirror-exclude.txt"
OPTS="-E -v --stats --delete-excluded "
OPTS="$OPTS --exclude-from=$EXCLUDE"

echo "Starting system backup ...'"
$RSYNC $OPTS "$SOURCE" "$MIRROR_DEST"

echo "Making $DEST bootable"
bless -folder "$DEST/System/Library/CoreServices"
echo "System backup finished."

# ----------------------------------------------
# Backup user directories in rotating archives
echo "Starting backup of user data ..."
$RSBACK -vc $RSBACKCONF users-daily
echo "Backup of user data finished."

# ----------------------------------------------
# Shutdown
echo "System will shutdown now!"
shutdown -h now
```

Konfiguration für rsback

```
# rsback.conf
# Backup user data on local system
# ---------------------------------
[global]
rsync_cmd = /usr/bin/rsync
cp_cmd = /sw/bin/cp
mv_cmd = /bin/mv
rm_cmd = /bin/rm
mkdir_cmd = /bin/mkdir

tasks = users-daily users-weekly

lock_dir = /var/lock
rsync_options = -al --delete --delete-excluded \
                --stats --numeric-ids
exclude_file = rsback.exclude
use_link_dest = yes
ignore_rsync_errors = 24

# ---------------------------------
[users-daily]
source = /Users/
destination = /Volumes/Backup/Users
rotate = daily 14
mode = rsync

# ---------------------------------
[users-weekly]
source = /Volumes/Backup/Users/daily.1/
destination = /Volumes/Backup/Users
rotate = weekly 12
mode = link
```

rsback — Datensicherung übers Netz

Probleme

- Grundsätzlich muss man davon ausgehen, dass Viele „mitlesen“ können.
- Mindestens die brisanten Daten müssen verschlüsselt übetragen werden.
- Zur Sicherung/Restaurierung von Daten braucht man i.d.R. Administrationsrechte. Der Remote-Zugriff auf ein System mit administrativen Rechten birgt aber zusätzliche Gefahrenquellen.
- Bei der Übertragung übers Netz muss immer ein Kompromiss zwischen **Datensicherheit** und und Funktionalität der **Datensicherung** eingegangen werden.

Auswege

- rsync bietet die Möglichkeit Daten über ssh zu übertragen.
- Root-Login über ssh ist bei den meisten Admins nicht gerne gesehen.
- Einschränkung der Root-Rechte auf das rsync-Kommando ist möglich, aber nicht einfach zu überwachen (ssh wrapper).
- **rsyncd** (Server-Modus) **kombiniert mit einem Tunnel** (z.B. OpenVPN) ist relativ einfach zu konfigurieren und hält rsync-Kommandos übersichtlich. Sicherheitslöcher durch unachtsame Konfigurationen können damit weitgehend vermieden werden.
  rsback ist für beide Möglichkeiten offen!

Danke für Ihre Geduld!