invis Server Datensicherung

invis-Server verfügen über ein mehrschichtiges Datensicherungssystem, bestehend aus regelmäßigen internen Sicherungen, kombiniert mit einer externen Sicherung auf externe Festplatten oder einen Datensicherungsserver.

Für die Überwachung und ggf. die Durchführung der externen Datensicherung ist primär der Betreiber eines invis-Servers verantwortlich. Dies ist beispielsweise der Geschäftsführer einen Unternehmens und nicht etwa der Systemadministrator. Selbstverständlich können diese Aufgaben innerhalb eines Unternehmens delegiert werden, sprich die Verantwortlichkeit für Durchführung und Überwachung an Mitarbeiter weitergegeben werden. Dies sollte in jedem Falle vertraglich geregelt werden. Einen externen IT-Dienstleister für die Datensicherung verantwortlich zu machen ist eher unsinnig. Er (wenn vorhanden) sollte allerdings für gelegentliche Kontrollen und Wiederherstellungstests herangezogen werden.

Auch wenn die Verantwortlichkeit für die Datensicherung innerhalb eines Unternehmens an einen oder mehrere Mitarbeiter delegiert wurde, sollte es ureigenstes Interesse der Geschäftsleitung sein die damit zusammenhängenden Aufgaben regelmäßig zu kontrollieren. Bezüglich der Haftung bei Schäden durch Datenverlust steht die Geschäftsleitung bzw. der Unternehmer immer in der Mitverantwortung. Versäumnisse bei der Datensicherung werden rechtlich stets als Mitverschulden im Sinne des § 254 (Mitverschulden) BGB bewertet.

Für bilanzierungspflichtige Unternehmen schreibt der Gesetzgeber eine tägliche Datensicherung vor. Dies ist gerade bei Verwendung von externen Festplatten als Sicherungsziel, die dann auch täglich gewechselt werden müssen zu bedenken. Mit einem Server als Sicherungsziel ist das durch Automation natürlich einfacher zu realisieren.

Unabhängig vom gewählten Sicherungsziel gilt, dass eine Datensicherung nur dann als Datensicherung gilt, wenn die gesicherten Daten räumlich getrennt von den Originaldaten aufbewahrt werden. Dabei meint „räumlich getrennt“ zumindest einen anderen Brandabschnitt. Dies kann auch in kleineren Büros die nicht über getrennte Brandabschnitte verfügen durch Aufbewahrung der Sicherungsmedien in einem Brandschutztresor gewährleistet werden. In einem solchen Fall kommt ein lokal betriebener Datensicherungsserver allerdings nicht in Frage. Sie können ihn ja kaum im Tresor betreiben…

interne Sicherungen

Interne Sicherungen sind zeitgesteuerte Sicherungen wichtiger Datenbestände. Während der Installation werden verschiedene Datensicherungsaufgaben angelegt:

Aufgabe / Script	Zyklus	Bemerkung
kbackup	Mo, Mi, Fr, So jeweils um 3:00 Uhr	Sicherung der Kopano-Konten / ermöglicht das wiederherstellen einzelner Objekte (Mailordner, Mails usw. z.B. bei versehentlichem Löschen
kdbdump	Di, Do, Sa jeweils um 3:00 Uhr	Sicherung der Kopano Datenbank / dient der Wiederherstellung der Kopano Datenbank im Notfall
dwdatasnapshot	Samstags um 1:30 Uhr	Sicherung des Dokuwiki-Datenbestandes

Aufgabe / Script	Zyklus	Bemerkung
alldump	Samstags um 6:00 Uhr	Sicherung aller Datenbanken des Servers mit Ausnahme der Kopano-Datenbank
adbackup	Täglich um 23:30	Sicherung des Active Directory / ab invis-Server 14.0

Die genannten Zyklen sind die Voreinstellungen nach der Installation eines invis-Servers. Sie lassen sich individuell anpassen. Vorgenommen werden die Einstellungen in:

```
/etc/cron.d/invis.cron
```

Diese Sicherungen erfolgen zunächst auf die lokalen Festplatten des Servers selbst und sind in der Freigabe Archiv im Unterverzeichnis „sicherungen“ zu finden. Damit diese Sicherungen nicht bis ins unermessliche Platz belegen werden die verschiedenen Sicherungsverzeichnisse zyklisch bereinigt. Das maximale Alter der Sicherungen ist in „Tagen“ in

```
/etc/invis/invis.conf
```

einstellbar und in der Regel auf 21 oder 42 Tage voreingestellt. Ältere Sicherungen werden gelöscht.

Alle internen Sicherungen werden von der externen Sicherung mit erfasst.

Hinweis: *Alle in der Tabelle genannten Sicherungsscripts können auch manuell auf der Kommandozeile des Servers aufgerufen werden.*

Einige, aber nicht alle der Sicherungs-Scripts versenden im Fehlerfall eine Wanrmail an eine während der Installation des Servers voreingestellt Mail-Adresse. Unabhängig davon sollten Sie regelmäßig einen Blick in die zuvor genannten Sicherungsverzeichnisse werden und kontrollieren, ob die Sicherungen wie gewünscht erfolgen.

externe Sicherungen

Basierend auf unserer Empfehlung eine invis-Server Installation unter Nutzung von Logical-Volume-Management durchzuführen haben wir ein eigene Datensicherungswerkzeuge entwickelt, welche Datensicherung durch Kombination von LV-Snapshots und „***rdiff-backup***“ oder „***borg***“ durchführt.

- **invis-rdbu** - Nutzt rdiff-backup
- **invis-bbu** - Nutzt borg Backup

Tipps zur Auswahl:

- ListenpunktVon beiden Paketen ist **invis-bbu** das neuere. Grund für die Entwicklung eines zweiten Tools waren immer wieder auftretende Probleme mit den von ***rdiff-backup*** erzeugten Backup-Repositories. In einzelnen Fällen traten vor allem durch eine ungewollte Unterbrechung der Datensicherung Fehler in den Repositories auf, die diese unbrauchbar machten. In der Folge führt dies dazu, dass alle weiteren Datensicherungen in ein solchermaßen beschädigtes Repository fehl schlagen.
- Zur Stabilität der von Borg-Backup erzeugten Repositories können wir derzeit, einfach aufgrund mangelnder Erfahrung, noch keine Aussagen machen.

- Ein großer Vorteil von borg Backup ist die Tatsache, dass dessen Repositories von vorne herein verschlüsselt werden können. D.h. es ist keine Datenträgerverschlüsselung erforderlich. Gerade bei Sicherungsmedien die sensible Datenenthalten und außer Haus aufbewahrt werden ist dies ein unschätzbarer Vorteil.
- Beide Systeme erlauben das Wiederherstellen älterer Dateiversionen aus der Sicherung, was mit borg ein gutes Stück einfacher geht als mit rdiff-backup.

Die Installation ist denkbar einfach. Installieren Sie zunächst eines der beiden Software-Pakete, unabhängig davon, ob Sie auf externe Platten oder einen Sicherungsserver sichern wollen.

```
linux:~ # zypper in invis-bbu
```

Als Sicherungsziele kommen wahlweise oder in Kombination externe USB bzw. eSATA Festplatten sowie ein gesonderter Sicherungsserver in Frage.

Sicherungsserver können via SSH oder SMB-Freigaben angesprochen werden.

Achtung: *Bei der Nutzung von SMB-Freigaben gehen leider die Besitz- und Zugriffsrechte der gesicherten Dateien verloren.*

Beim Einsatz externer Festplatten nutzen wir die Möglichkeiten der UDEV-Hardware Verwaltung unter Linux. Die Sicherungsplatten werden dem Server bekannt gemacht und er startet die Datensicherung automatisch, wenn „genau diese“ Festplatten mit dem Server verbunden werden. Andere Festplatten lösen keine Sicherung aus. Wichtig zu wissen, ist dass das Verbinden der Festplatte der auslösende Moment ist.

Achtung: *Die Festplatte mit dem Server verbunden zu lassen sorgt **nicht** für kontinuierlich Datensicherungen.*

Achtung: *Vor dem Gesetz gilt eine Datensicherung nur dann als Datensicherung, wenn die Sicherungen **räumlich getrennt** von den originalen Daten aufbewahrt werden. Dabei meint „räumlich getrennt“ mindestens einen anderen Brandabschnitt. D.h. Gesicherte Daten müssen durch mindestens eine Brandschutzwand und Brandschutztür von den Originaldaten getrennt sein. Ein Tresor mit Brandschutzfunktion erfüllt diese Forderung ebenfalls.*

Weiterhin fordert der Gesetzgeber von Bilanz-pflichtigen Unternehmen eine tägliche Datensicherung. Dies bedeutet für die Sicherung auf externe Festplatten, dass Sie mindestens 2 Platten im Wechsel nutzen sollten, so dass eine immer an einem gesicherten Aufbewahrungsort liegt.

Sicherung auf externe Festplatten

Einrichten und registrieren der Datensicherungsfestplatten

Bei der Initialisierung externer Sicherungsfestplatten spielt es keine Rolle, für welches Sicherungssystem Sie sich entschieden haben. Die Vorgehensweise ist bei beiden Systemen identisch.

Legen sie auf einer externen Festplatte eine einzige Partition an und formatieren Sie diese mit einem „ext3“ oder „ext4“ Dateisystem. Trennen und verbinden Sie die Festplatte nach der Formatierung neu

mit Ihrem Server. Warten Sie ein paar Sekunden.

Während der erwähnten wenigen Sekunden Wartezeit können Sie dem Server bei der Hardware-Erkennung zuschauen:

```
linux:~ udevadm monitor
monitor will print the received events for:
UDEV - the event which udev sends out after rule processing
KERNEL - the kernel uevent
...
UDEV  [84592.431934] add
/devices/pci0000:00/0000:00:14.0/usb1/1-2/1-2:1.0/host4/target4:0:0/4:0:0:0/
block/sdc (block)
UDEV  [84592.462554] add
/devices/pci0000:00/0000:00:14.0/usb1/1-2/1-2:1.0/host4/target4:0:0/4:0:0:0/
block/sdc/sdc1 (block)
```

Wenn sich auf dem Bildschirm nichts mehr tut, ist die Erkennung abgeschlossen. Beenden ***udevadm*** mit der Tastenkombination STRG+C.

Führen Sie jetzt das Script ***udbadddisk*** aus.

```
linux:~ udbadddisk
Gefundene Festplatte: sdc
Gefundene Partition: sdc1

Gefundenes Merkmal Partitionsgröße: 3907027120
Gefundenes Merkmal Seriennummer: FDC0FD20EF00000FD0FCC4F2FFFFFF
Datensicherungsplatte hinzugefügt
```

Das Script sollte Ihnen zunächst die Bezeichnung der Festplatte (sdX) und der darauf angelegten Partition (sdX1) anzeigen. Durchsucht werden die letzten 3 Stunden des Systemjournals auf entsprechende Hotplug-Events. Wird keine geeignete Festplatte gefunden, bricht das Script ab.

Ist eine Sicherungsplatte schon länger als 3 Stunden mit dem Server verbunden und Sie kennen den Namen der darauf eingerichteten Sicherungspartition, können Sie diese Partition auch als Aufrufparameter angeben:

```
linux:~ udbadddisk sdc1
Gefundene Festplatte: sdc
Gefundene Partition: sdc1

Gefundenes Merkmal Partitionsgröße: 3907027120
Gefundenes Merkmal Seriennummer: FDC0FD20EF00000FD0FCC4F2FFFFFF
Datensicherungsplatte hinzugefügt
```

Das Script ***udbadddisk*** ermittelt Informationen zur eindeutigen Identifikation der Festplatte und generiert daraus eine UDEV-Regel in der Datei:

```
/etc/udev/rules.d/80-backupdisk.rules
```

Hier die im oben gezeigten Beispiel entstandene UDEV-Regel:

```
SUBSYSTEMS=="usb", KERNEL=="sd*",
ATTRS{serial}=="FDC0FD20EF00000FD0FCC4F2FFFFFF", ATTR{size}=="3907027120",
SYMLINK+="backup", RUN+="/usr/bin/udbdiskplugged"
```

Sie besagt, wenn ein Gerät des Typs „sd“ am USB-Bus mit der Seriennummer „FDC0FD20EF00000FD0FCC4F2FFFFFF“ und der genannten Partitionsgröße erkannt wird, soll unter „/dev“ ein Symlink unter dem Namen „backup“ angelegt werden, der auf das gefundene Gerät verweist (/dev/sdX1). Danach soll das Script ***udbdiskplugged*** ausgeführt werden.

Sicherung konfigurieren

Sind alle Sicherungsfestplatten vorbereitet, kann die Sicherung selbst konfiguriert werden. Dies erledigt das Script ***udbconf***

Zur Verwendung von ***udbconf*** müssen Sie sich im Klaren sein, was Sie sichern wollen.

Wie eingangs bereits erläutert, setzen „invis-rdbu“ und „invis-bbu“ die Verwendung von Logical-Volume-Management zwingend voraus. Zur Konfiguration der Datensicherung benötigen Sie den Namen der Volume-Group und der darauf liegenden zu sichernden Volumes. Sichern sollten Sie grundsätzlich alle logischen Volumes des Servers, nur dann ist es möglich eine vollständige Server-Installation aus der Sicherung wiederherzustellen. Wenn Sie bei der Partitionierung der Server-Festplatten gemäß unserem Beispiel hier im **Wiki** vorgegangen sind, sind das die Volumes:

- **root**
- **home**
- **srv**
- **var**

Sollten Sie die Namen der Volume-Group und der darin liegenden Volumes nicht mehr wissen, können Sie sie mit folgenden Kommandos anzeigen:

Volume-Group scannen:

```
linux:~ # vgscan -v
```

Logical Volumes scannen:

```
linux:~ # lvscan -v
```

Weiterhin müssen Sie die Größe des zu verwendenden Snapshot-Volumes festlegen. Die maximal mögliche Größe zeigt ***udbconf*** an. Normalerweise sollte eine Snapshot-Volume-Größe von 20GB ausreichen.

udbconf starten:

```
linux:~ # udbconf
Datensicherung udrdbu wird konfiguriert
```

```
Geben Sie bitte den Namen der Volume-Group ein, in der sich die zu
sichernden logical Volumes befinden: system
Geben Sie bitte die Namen der zu sichernden logical Volumes ein. z.B. srv
(Mehrere Volume-Namen durch Leerzeichen trennen.): root srv home var
Geben Sie bitte den vollständigen Namen des zuständigen Administrators ein:
Stefan Schäfer
Geben Sie bitte die Mail-Adresse an, an die Datensicherungsmeldungen
geschickt werden sollen: info@local-net.loc
Geben Sie bitte die Mail-Adresse an, die als Absender verwendet werden
sollen: backup@local-net.loc
Geben Sie bitte die Größe des Snapshotvolumes an. Die Größe muss größer als
die größte zu sichernde Datei und kleiner als 357,50 GiB sein. (Angabe xxG):
20G
Konfiguration abgeschlossen
```

Hier die resultierende Konfigurationsdatei:

```
# Konfigurationsdatei fuer invis-bbudisk
# (c) 2010-2020 Stefan Schaefer - invis-server.org

# Zielhost
targetHost:localhost
targetDirUDEV:/mnt/udevsync/borgbackups

# LVM Daten
volumeGroup:system
# Mehrere Volume Namen durch Leerzeichen trennen
sourceVolume:srv home root var
# Groeße des Snapshots
snapshotSize:10G
# Name des Snapshot-Volumes
snapshotVolume:invisdiskbackup

# Admin
adminitrator:Stefan Schäfer

# Mail-Absender und -Empfaenger
mailTo:hilfe@fsproductions.de
mailFrom:backup@invis-server.org
# nur im Fehlerfall oder immer Mails versenden.
# Werte: always, failure
mailWhen:failure

# borg-backup Optionen
bbOptions:--exclude-from /etc/invis/backup-exclude-list --one-file-system -s

# Dasimonitor aktivieren - sollte nicht erforderlich sein!
dasiMonitorAct:0
```

Grundsätzlich versuchen „invis-rdbu“ und „invis-bbu“ Emails zu versenden, die generell über die Durchführung der Datensicherung informieren, oder nur Fehler melden. Voreingestellt ist das

Versenden von Mails im Fehlerfall. Diese Einstellung kann manuell in der automatisch erzeugten Konfigurationsdatei

```
/etc/invis/rdbudisk.conf
```

oder

```
/etc/invis/bbudisk.conf
```

angepasst werden.

Bei beiden Tools ist es möglich dem eigentlichen Sicherungskommando „rdiff-backup“ bzw. „borg“ Befehlsoptionen mitzugeben. Hier haben wir jeweils Vorgaben gemacht. In der Hauptsache geht es dabei um Excludes, also Regeln darüber was von den Sicherungen ausgeschlossen werden soll.

Die letzte Option `dasiMonitorAct` ermöglicht es das Tool ***dasimonitor*** scharf zu schalten. Wird eine Datensicherung rabiat unterbrochen, kann es sein, dass der Sicherungssnapshot erhalten und gemountet bleibt. Dies würde künftige Datensicherungen verhindern. Ist ***dasimonitor*** scharf geschaltet kontrolliert es alle 3 Minuten, ob solche Relikte vorhanden sind und entfernt sie gegebenenfalls. Das birgt allerdings auch Gefahren. Beispielsweise wenn Sie manuell einen LV-Snapshot anlegen und diesen nach „/mnt/backup“ mounten. ***dasimonitor*** würde das als Relikt ansehen und entfernen. Schalten Sie ***dasimonitor*** nur scharf, wenn es bei der Datensicherung immer mal wieder zu Problemen kommt. Im unscharfen Zustand überwacht dasimonitor lediglich, ob die Datensicherung aktiv ist oder nicht und protokolliert dies für die Anzeige im invis-Portal.

Testen

Vor allem, wenn Sie die Einrichtung remote vornehmen ist es wichtig einen Test der Sicherung durchführen zu können, ohne, dass Ihnen helfende Hände zur Seite stehen. Dazu müssen Sie einen „uevent“ triggern, damit die neue UDEV Erkennungsregel angewendet wird:

```
linux:~ echo change > /sys/block/sdX/uevent
```

Einrichtung der Datensicherung auf einen Sicherungsserver

Der Sicherungsserver selbst benötigt einfach nur eine Linux-Installation mit genügend Festplattenplatz. Auf ihm muss der SSH-Dienst laufen und je nach verwendeter Sicherungslösung ***rdiff-backup*** oder ***borg*** installiert sein. Letzteres wird bei Verwendung des neueren **invis-bbu** beim Initialisieren der Borg-Repositories automatisch erledigt. Kommt **invis-rdbu** zum Einsatz muss ***rdiff-backup*** händisch auf dem Zielserver installiert werden.

SSH vorbereiten

Damit die Sicherung über SSH erfolgen kann, muss sich der Benutzer „root“ des Quell-Servers ohne Passwort per SSH am Sicherungsserver anmelden können. Es ist nicht notwendig, dass auch auf dem Sicherungsserver das Konto des Benutzers „root“ verwendet wird. Es genügt dort ein Benutzerkonto, welches Schreibrecht im Zielverzeichnis der Datensicherung hat.

Die passwortlose Anmeldung gelingt, wenn die Authentifizierung am Sicherungsserver mit einem Schlüssel erfolgt. Dieser Schlüssel kann auf dem Quellserver (angemeldet als Benutzer „root“) einfach generiert werden:

```
invis:~ # ssh-keygen
...
Generating public/private rsa key pair.
Enter file in which to save the key (/root.sshid_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
fb:8b:20:3d:97:25:09:e7:ef:fe:63:d1:b3:49:1f:e0 [MD5] root@server
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|      . .        |
|       + .   .   |
|        S . ...  |
|     .   *  .E+. |
|    . + + .  o =.|
|     . + +  o o .|
|        ..=+..   |
+--[MD5]----------+
```

Kopieren Sie dann (auf sicherem Weg) den öffentlichen Teil (public key) des erzeugten Schlüsselpaares in des Home-Verzeichnis des Backup-Benutzerkontos auf dem Sicherungsserver:

```
linux:~ # scp .ssh/id_rsa.pub backup@backupserver:~
```

Melden Sie sich jetzt am Backupserver mit diesem Konto an und hängen Sie den Schlüssel an die Datei „authorized_keys“ in dessen Homeverzeichnis an:

```
backup@backupserver:~ # cat id_sa.pub >> .ssh/authorized_keys
```

Damit der SSH-Dienst des Backup-Servers die Anmeldung per Schlüssel akzeptiert müssen in der Konfigurationsdatei des SSH-Servers

```
/etc/ssh/sshd_config
```

folgende Einträge vorhanden sein:

```
...
PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and
.ssh/authorized_keys2
# but this is overridden so installations will only check
```

```
.ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys
...
```

Ist das der Fall, können sie einen Login-Versuch unternehmen:

```
linux:~ ssh backup@backupserver
```

Gelingt dies ohne Passwortabfrage, ist die Vorbereitung abgeschlossen.

Sicherung einrichten

Das Einrichten einer Sicherung auf einen externen Server läuft analog zur Sicherung auf externe Festplatten ab. Es existieren mit ***rdbunetconf*** und ***bbunetconf*** auch hier ein Scripts, welche die Konfigurationsdatei zur Sicherung generiert. Es ist einfach ohne weitere Optionen aufzurufen:

```
invis:~ # bbunetconf
```

Im Verlauf des Scripts wird gefragt, ob die Sicherung via SSH oder auf eine einzuhängende SMB-Freigabe erfolgen soll. Wir empfehlen ganz eindeutig die SSH-Methode zu bevorzugen. Die SMB-Methode ist eher eine Notlösung, beispielsweise zur Sicherung auf ein NAS System welches lediglich via SMB nutzbar ist. Dabei kommt es allerdings immer wieder zu Problemen, beispielsweise wegen maximaler Pfadlängen, unterschiedlichen Zeichensätzen oder auch der Konsistenz von Zugriffs- und Besitzrechten.

Kommt **invis-bbu** zum Einsatz müssen die Sicherungsrepositories noch vorbereitet werden. Anders als bei ***rdiff-backup*** sind dies nicht einfach leere Verzeichnisse. Wie bereits erwähnt unterstützt ***borg*** auch nativ verschlüsselte Repositories. Teil des Backup-Pakets ist das Tool ***bbcrr***, es installiert ***borg*** auf dem Zielserver, legt die Repositories an, verschlüsselt sie und speichert das zugehörige Passwort in:

```
/etc/invis/invis-pws.conf
```

Darüber hinaus extrahiert es die Schlüssel der Repositories und legt diese in

```
/etc/invis/private/
```

ab.

Das Tool wird ohne weitere Optionen auf dem Quell-Server aufgerufen:

```
invis:~ # bbcrr
```

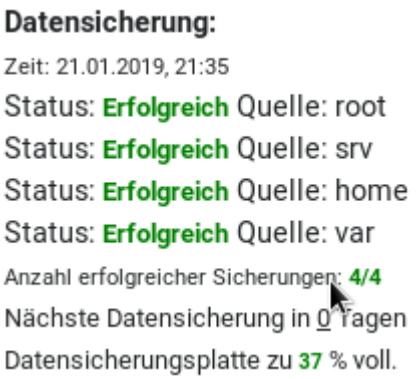
Durchführung und Kontrolle

Sicherung auf Backupserver

Wird auf einen Server gesichert, müssen Sie sich um die Durchführung der Datensicherung keine Gedanken machen, sie erfolgt automatisch, zeitgesteuert.

Sicherung auf externe Festplatten

Wird auf externe Festplatten, ist das Verbinden der Platte mit dem Server das auslösende Ereignis. Das bedeutet auch, dass Sie die Sicherungsfestplatte nach erfolgter Sicherung wieder vom Server trennen müssen.



***Achtung**: Lassen Sie eine Sicherungsfestplatte am Server angeschlossen, werden **nicht** automatisch regelmäßige Datensicherungen durchgeführt.*

Die ideale Vorgehensweise ist eine abwechselnde Datensicherung auf zwei Festplatten. D.h. Eine der Festplatten ist mit dem Server verbunden, während die zweite an einem „sicheren“ Ort aufbewahrt wird. Tauschen Sie idealerweise zu einem Zeitpunkt an dem wenig oder nicht auf dem Server gearbeitet wird die Festplatten aus.

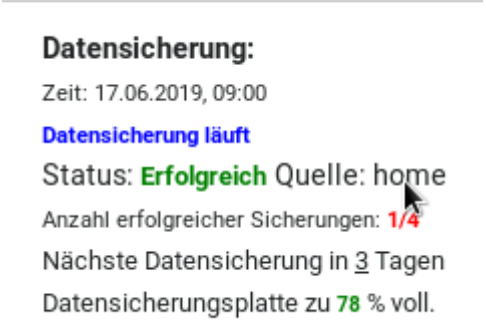
Sie können die verbundene Festplatte dabei einfach ohne weitere Vorbereitung vom Server trennen, die logische Trennung wird nach erfolgter Sicherung bereits vom Sicherungsprogramm vorgenommen.

Kontrolle

Zur Kontrolle der Datensicherung versendet das Sicherungssystem im Fehlerfall oder wahlweise auch immer Status-Mails an einen in der Konfiguration festzulegenden Administrator. Einen kleinen Haken hat die Sache jedoch, läuft die Datensicherung aufgrund eines Problems gar nicht erst an, wird leider auch keine Mail versendet.

Sie können jederzeit den Status der Datensicherung auf der Status-Seite Ihres invis-Portals überprüfen. Eine Anmeldung am Portal ist dafür nicht notwendig. Die Statusdaten werden Nutzern des

lokalen Netzes immer gezeigt.



Nebenstehende Abbildungen zeigen den Idealfall - alle 4 Datensicherungsaufgaben wurden erfolgreich durchgeführt - sowie eine gerade aktive Datensicherung. In beiden Fällen ist auf der Datensicherungplatte noch ausreichend Platz vorhanden. Weiterhin wird angezeigt, wie viel Zeit bis zur nächsten Datensicherung verbleibt. Dieser Zeitraum kann in der Konfiguration des invis-Portals individuell eingestellt werden. Ist eine Datensicherung überfällig wird dies ebenfalls angezeigt. Dabei ändert sich die Farbe der angezeigten Tage selbstverständlich in rot.

Daten wiederherstellen

Die Wiederherstellung von Daten unterscheidet sich geringfügig, je nach dem, ob das Sicherungsziel externe Festplatten oder ein Sicherungsserver ist und deutlich, je nachdem, ob **invis-rdbu** oder **invis-bbu** genutzt wird.

Wiederherstellung von Daten vorbereiten

Sicherungsfestplatten

Im Falle externer Sicherungsplatten ist beim Verbinden der Platte mit dem Server zu verhindern, dass das Verbinden unmittelbar eine neue Sicherung auslöst. Für diesen Zweck existiert ein eigenes Script, welches **vor** dem Verbinden der Festplatte ohne weiteren Parameter auf der Kommandozeile des Servers aufgerufen wird:

```
invis: # udbrestore
```

Dies gilt gleichermaßen für **invis-rdbu** und **invis-bbu**.

Jetzt kann die Sicherungsplatte verbunden und eingehängt werden:

```
invis:~ # mount /dev/backup /mnt/udevsync
```

Datensicherungsserver

In diesem Fall melden Sie sich zunächst per SSH am Sicherungsserver an:

```
invis:~ # ssh root@sicherungsserver
```

```
sicherungsserver:~ #
```

Die Abfrage eines Passworts erfolgt nicht, da die Authentifikation via Public-Key Verfahren erfolgt.

Datenwiederherstellung mit rdiff-backup

Für die Wiederherstellung von Daten ist es entscheidend, wie schnell der Verlust einer Datei oder eines Verzeichnisses auffällt. Da die Basis der Sicherung das Programm ***rdiff-backup*** ist, ist jeweils der letzte Sicherungsstand direkt verfügbar, ältere Sicherungsstände liegen nur noch als sogenannte Inkremente in der Sicherung vor und müssen unter Verwendung von ***rdiff-backup*** **berechnet** werden. In jedem Fall benötigen Sie Erfahrung auf der Linux-Kommandozeile.

In der Praxis bedeutet das, dass eine Datei die nach der letzten Datensicherung auf dem Server gelöscht wurde, auf dem Sicherungsmedium vollwertig vorhanden ist und einfach zurück kopiert werden kann. Erfolgt nach dem Löschen eines Originals eine zweite Sicherung auf das gleiche Sicherungsmedium muss die Datei aus den Inkrementen wiederhergestellt werden. Letzteres ist deutlich aufwändiger.

Daten aus der aktuellsten Sicherung wiederherstellen

Um Daten wiederherzustellen müssen Sie auf der Kommandozeile in das Verzeichnis der Sicherung wechseln. Je nach Sicherungsziel müssen Sie unterschiedliche Wege gehen.

Ist die Festplatte angeschlossen und gemountet, kann ins Verzeichnis der Sicherungen gewechselt werden:

```
invis:~ # cd /mnt/udevsync/rdbackups
```

Unterhalb dieses Verzeichnisses befinden sich für jede Sicherungsquelle ein Unterverzeichnis:

```
invis:~ # ls -l
drwxr-xr-x 19 root root 4096  4. Okt 16:06 home
drwxr-xr-x 20 root root 4096  1. Dez 12:30 root
drwxr-xr-x 12 root root 4096  1. Dez 13:38 srv
drwxr-xr-x 13 root root 4096  1. Dez 12:05 var
```

Die Nutzdaten Ihres Unternehmens finden Sie unter „/home“ bzw. „/srv“. In diesen Verzeichnissen finden Sie 1:1 die Datenstruktur vor wie auf den Quellverzeichnissen des Servers. Um Daten der letzten Sicherung wiederherzustellen müssen Sie lediglich die entsprechende Datei oder das Verzeichnis aus dem Sicherungsverzeichnis an einen Ort Ihrer Wahl kopieren. Am einfachsten können Sie dass mit Hilfe des auf jedem invis-Server installierten Dateimanagers „Midnight Commander“ (***mc***) erledigen.

Nach dem Wiederherstellen der Daten ist die Sicherungsfestplatte wieder auszuhängen und vom Server zu trennen:

```
invis:~ # umount /mnt/udevsync
```

Wiederherstellen älterer Dateiversionen mittels rdiff-backup

Gehen wir im folgenden Beispiel von einer Sicherung auf eine Datensichrungsfestplatte aus und entsprechend der obigen Anleitung vor:

```
invis:~ # udbrestore
invis:~ # mount /dev/backup /mnt/udevsync
```

Wechseln Sie jetzt ins Verzeichnis der Sicherungen:

```
invis:~ # cd /mnt/udevsync/rdbackups
invis:/mnt/udevsync/rdbackups #
```

Auf einem Backupserver gehen wir vom folgenden Pfad aus. Wechseln sie jetzt ins Sicherungsverzeichnis:

```
sicherungsserver:~ # cd /srv/backup
```

Dort befinden sich die Sicherungen. In der Regel sind dies die Verzeichnisse `home`, `srv`, `var` und `root`. Nehmen wir an Sie wollen das persönliche Verzeichnis eines Benutzer in einem älteren Sicherungsstand wiederherstellen. Dazu sollten Sie zunächst nachschauen, welche Sicherungen auf dem Datenträger vorhanden sind:

```
invis:/mnt/udevsync/rdbackups # rdiff-backup --list-increments home/hmohr/
Found 4 increments:
    hmohr.2018-11-30T19:43:23+01:00.dir   Fri Nov 30 19:43:23 2018
    hmohr.2018-12-04T19:58:57+01:00.dir   Tue Dec  4 19:58:57 2018
    hmohr.2019-01-11T20:49:56+01:00.dir   Fri Jan 11 20:49:56 2019
    hmohr.2019-01-15T19:20:28+01:00.dir   Tue Jan 15 19:20:28 2019
Current mirror: Thu Jun 13 21:20:53 2019
```

Das Beispiel zeigt insgesamt neben der letzten aktuellen Sicherung insgesamt 4 ältere Sicherungsstände. (Nebenbei bemerkt ist dies ein trauriges Beispiel von Datensicherungseifer; ganze 4 Sicherungen in 8 Monaten und die letzte ist bereits 5 Monate alt...)

Um jetzt die älteste Sicherung wiederherzustellen gehen Sie wie folgt vor. Legen Sie zunächst ein temporäres Wiederherstellungsverzeichnis an:

```
invis:/mnt/udevsync/rdbackups # mkdir /srv/restore/hmohr
```

Jetzt starten Sie die Wiederherstellung:

```
invis:/mnt/udevsync/rdbackups # rdiff-backup --restore-as-of 2018-11-30
home/hmohr/ /srv/restore/hmohr/
```

Dabei erwartet die Option `restore-as-of` die Angabe des Datums des gewünschten Sicherungsstandes. Wählen Sie ein Datum welches zuvor mit der Option `list-increments` aufgezeigt wurde.

... work in progress ...

Datenwiederherstellung mit borg Backup

Anders als bei ***rdiff-backup*** legt ***borg*** keine gesicherten Daten in Ihrer Normalform ab. D.h. eine Rücksicherung mit Copy & Paste ist zunächst nicht möglich. Dennoch ist die Lösung eleganter als beim Konkurrenten. Die Datensicherungsrepositories lassen sich als Dateisysteme in den Verzeichnisbaum des Servers einhängen. (Zugegeben, das ist auch mit ***rdiff-backup*** möglich, hat uns aber weit weniger gefallen.)

Je nach dem, ob Sie als Sicherungsziel externe Festplatten oder einen Backup-Server nutzen müssen Sie entweder auf den Backup-Server wechseln oder die Sicherungsplatte mounten. Siehe Abschnitt „Vorbereitung“ weiter oben im Text.

Ist dies geschehen, können Sie die Sicherungsrepositories mit dem Kommando ***borg*** mounten. Sie benötigen dazu das Passwort der Repository Verschlüsselung, zu finden in:

```
/etc/invis/invis-pws.conf
```

auf dem invis-Server.

Hängen Sie das gewünschte Repository ein:

```
invis:~ # borg mount /mnt/udevsync/bbackups/srv /mnt/restore
Enter passphrase for key /srv/backup/srv:
```

Im Zielverzeichnis finden Sie jetzt für jede erfolgte Datensicherung in dieses Repo ein Verzeichnis, benannt mit dem Datum der Datensicherung. Wechselns Sie ins gewünschte Sicherungsverzeichnis und tauchen Sie dort in den Verzeichnisbaum bis an die Stelle der wiederherzustellenden Daten ein. Jetzt können Sie die Daten einfach aus dem Repository heraus kopieren.

Clever ist auch eine Wiederherstellung mittels ***rsync*** direkt an den Ursprungsort oder ein gesondertes Wiederherstellungsverzeichnis. Das geht natürlich auch Remote vom Backupserver aus.

Beispiel:

```
backupserver:~ # rsync -raHAXv -e "ssh -i /home/user/.ssh/sshkey"
/mnt/restore/2020-06-15/mnt/backup/srv/shares/gruppen/technik/zeichnungen
root@invis.example-net.loc:/srv/shares/gruppen/technik/zeichnungen
```

Die im Beispiel gezeigten ***rsync*** Optionen „aHAX“ sorgen dafür, dass Zugriffs- und Besitzrechte, erweiterte Attribute und POSIX-ACLs erhalten bleiben.

Achtung: *Passen Sie beim Wiederherstellen auf jeden Fall darauf auf, dass sie sich dabei keine wichtigen Daten mit älteren Dateiversionen überschreiben!*

Der Quellpfad im gezeigten Beispiel sieht auf den ersten Blick ein wenig merkwürdig aus. Das liegt daran, dass ***borg*** den Quellpfad der erhält. Da wir durch einen LV-Snapshot sichern und dieser während der Sicherung nach „/mnt/backup“ gemountet ist, taucht dieser Pfad natürlich genau so in der Sicherung auf.

Ist die Wiederherstellung abgeschlossen muss das Sicherungsrepository wieder ausgehangen werden:

```
backupserver:~ # umount /mnt/restore
```

Wurde von einer Sicherungsfestplatte wiederhergestellt, muss diese natürlich auch wieder ausgehangen werden:

```
invis:~ # umount /mnt/udevsync
```

Sicherung Überwachen und ggf. Aufräumen

Global

Wir haben festgestellt, dass gerade bei der Verwendung von externen Sicherungsfestplatten immer wieder Probleme auftreten, beispielsweise, weil Festplatten versehentlich getrennt werden noch während eine Sicherung läuft. Um derartiges zu vermeiden zeigt das invis-Portal inzwischen wesentlich deutlicher an, dass eine Sicherung aktiv ist. Dies setzt natürlich die Nutzung des invis-Portals voraus....

Um nach einem Abbruch einer Sicherung verwaiste Mounts oder Snapshot-Volumes wieder los zu werden läuft per Cornjob alle 3 Minuten das Script ***dasimonitor***. Das Script ist auch die Quelle der Informationen die im invis-Portal angezeigt werden.

Borg Backup

Etwa durch einen unterbrochenen Backup-Vorgang kann es sein, dass ein Borg-Sicherungs-Repository dauerhaft geblockt ist. Eine solche Blockade muss manuell aufgehoben werden. Zuvor ist sicherzustellen, dass aktuell niemand auf das Repository zugreift. Ist dies geschehen gehen Sie wie folgt vor:

Disk-Backup

Backup-Server

```
invis:~ # borg break-lock
ssh://root@backupserver.example.loc:22/pfad/zum/repo
```

Sicherungsverlauf im invis-Portal anzeigen

Das Web-Portal des invis-Servers ist ebenfalls in der Lage über Erfolg oder Misserfolg der Datensicherung zu informieren. Diese Funktion muss zunächst in der Konfiguration des invis-Portals frei geschaltet werden. Öffnen Sie für diesen Zweck die Datei:

```
/etc/invis/portal/config.php
```

und befreien Sie die Zeile:

```
// Bitte folgende Zeile von den Kommentarzeichen befreien, wenn udevsync zur
Datensicherung verwendet wird.
$STATUS_BACKUP_TIMER = 3;
```

von den führenden Kommentarzeichen (Doppel-Slash '/'). In dieser Zeile können Sie auch den gewünschten Sicherungszyklus vorgeben. Das Portal wird dann deutlich darauf hinweisen, wenn eine Sicherung nicht wie geplant durchgeführt wird.

Die Anzeige erfolgt im Portal auf der Seite „Status“.

Kopano Datensicherung

Die Groupware Kopano bringt ein eigenes Brick-Level Backup-System mit. invis-Server sind so eingerichtet, dass sie täglich abwechselnd einmal die komplette Kopano-Datenbank in Form eines Dumps sichern und einmal das Bricklevel-Backup durchführen. Dabei dient die gesicherte Datenbank als Disaster-Recovery, also dem Wiederherstellen der gesamten Datenbank im Falle eines Crashs und das Bricklevel-Backup dem Wiederherstellen einzelner Elemente, beispielsweise wenn diese versehentlich gelöscht wurden.

Ziel beider Sicherungen ist das Verzeichnis:

```
/srv/shares/sicherungen
```

Wiederherstellung einzelner Elemente aus dem Brick-Level Backup

Das benötigte Werkzeug ist ***kopan-backup***, es bietet verschiedene Möglichkeiten Elemente eines geicherten Kopano-Stores wiederherzustellen. Beginnen wir damit uns den Inhalt der Sicherung anzeigen zu lassen. Wechseln Sie dazu ins Verzeichnis in dem die gesicherten Stores liegen:

```
/srv/shares/archiv/sicherungen/kopanostores
```

Der folgende Befehl gibt den gesamten Inhalt eines gesicherten Benutzer-Stores aus:

```
invis:/srv/shares/archiv/sicherungen/kopanostores # kopano-backup --index
username
```

Im Beispiel steht „username“ sowohl für den Namen des Sicherungverzeichnisses eines Benutzers, als auch für den aktiven Benutzer in dessen Store ein oder mehrere Elemente wiederhergestellt werden sollen.

Mit der Option **`-u username2`** können Elemente auch in einen anderen aktiven Store wiederhergestellt werden.

Um die Ausgabe einzugrenzen, ist ***grep*** ein geeignetes Werkzeug:

```
invis:/srv/shares/archiv/sicherungen/kopanostores # kopano-backup --index
username | grep Posteingang
```

Die Ausgabe der Informationen erfolgt tabellarisch, Spaltentrenner ist das Komma. Nachfolgend der Aufbau der Zeilen:

```
Eindeutige ID,Ordner,Datum Uhrzeit,Betreff oder Property
```

Anhand der ID können einzelne Elemente wiederhergestellt werden. Suchen Sie zunächst wie oben gezeigt das wiederherzustellende Element, beispielsweise anhand des Betreffs oder des Datums eines Elementes. Nutzen Sie jetzt die ID des gesuchten Elements um es wiederherzustellen:

Wiederherstellen eines einzelnen Elementes

```
/srv/shares/archiv/sicherungen/kopanostores # kopano-backup --restore
username --sourcekey DFD123BE9FD84F7AB08E00DD959F2730080100000000
```

Wiederherstellen aller Elemente eines Ordners

```
/srv/shares/archiv/sicherungen/kopanostores # kopano-backup --restore
username -f Posteingang
```

Dabei wird im Store des Benutzers nichts überschrieben. Bereits vorhandene Elemente werden automatisch übersprungen.

Wiederherstellung nach Datum

Auch eine zeitliche Eingrenzung der Wiederherstellung auf ein Datum oder einen Datumsbereich ist möglich.

```
/srv/shares/archiv/sicherungen/kopanostores # kopano-backup --restore
username -b 2014-01-01 -e 2015-01-01
```

Die Optionen **-b** und **-e** stehen wie kaum anders zu vermuten für „begin“ und „end“.

Komplexeres Beispiel

Die hier gezeigten Beispiele lassen sich selbstverständlich auch kombinieren.

Wenn Sie beispielsweise alle Mails aus dem Posteingang eines Users aus einem bestimmten Zeitraum in einen Unterordner eines anderen Users wiederherstellen wollen sähe das wie folgt aus:

```
/srv/shares/archiv/sicherungen/kopanostores # kopano-backup --restore
username1 -f Posteingang -b 2014-01-01 -e 2015-01-01 -u username2 --restore-
root from_username1
```

Zu erwähnen wäre noch die Option **- -recursive**, sie stellt alle Elemente eines Ordners inklusive Unterordner wieder her.

Wiederherstellung der vollständigen Datenbank

***Achtung**: Die Wiederherstellung der Kopano-Datenbank sollten Sie nur durchführen, wenn die vorhandene Datenbank irreparabel beschädigt ist. Die vorhandene Datenbank wird bei der Wiederherstellung vollständig überschrieben.*

Es gibt zwei mögliche Vorgehensweisen. Die Sicherung liegt gezippt vor. Sie können die Datei, wenn gewünscht zunächst entzippen und dann wiederherstellen, oder Sie führen beide Schritte in einem durch.

2 Schritte

```
invis:~ # gunzip gunzip
/srv/shares/archiv/sicherungen/vollsicherungen/datenbanksicherungen/20171003
/kopano.invis.20171003.gz
invis:~ # mysql -u root -p kopano <
/srv/shares/archiv/sicherungen/vollsicherungen/datenbanksicherungen/20171003
/kopano.invis.20171003
```

1 Schritt

```
invis:~ # gunzip <
/srv/shares/archiv/sicherungen/vollsicherungen/datenbanksicherungen/20171003
/kopano.invis.20171003.gz | mysql -u root -p kopano
```

From:
https://wiki.invis-server.org/ - **invis-server.org**

Permanent link:
https://wiki.invis-server.org/doku.php?id=invis_server_wiki:dasi&rev=1591791594

Last update: **2020/06/10 12:19**