Backups mit Linux

ITS-Net-Lin

Sebastian Meisel

20. Dezember 2024

1 Einführung

Backups sind ein essenzieller Bestandteil der IT-Sicherheit und Datenverwaltung. Sie gewährleisten, dass Daten nach einem Hardwareausfall, versehentlichem Löschen oder einer Malware-Infektion wiederhergestellt werden können. In dieser Einführung betrachten wir die verschiedenen Arten von Backups und wie diese mit dem Tool `rsync` in Linux umgesetzt werden können.

1.1 Funktionsweise von rsync

Das Tool `rsync` ist ein Programm zur Synchronisation und Sicherung von Dateien, das standardmäßig auf Linux-Systemen installiert ist. Es arbeitet effizient, indem es nur die geänderten Daten zwischen Quelle und Ziel überträgt. `rsync` nutzt dazu sogenannte delta-Transfers, bei denen nur die tatsächlich veränderten Teile einer Datei kopiert werden.

1.2 Unterschiedliche Backup-Strategien

Es gibt drei grundlegende Arten von Backups, die sich hinsichtlich Speicherbedarf, Zeitaufwand und Wiederherstellungszeit unterscheiden:

1.2.1 Vollbackup

Ein Vollbackup umfasst alle Daten in einem definierten Verzeichnis oder auf einem gesamten Datenträger. Es ist die vollständigste Form des Backups, benötigt aber am meisten Speicherplatz und Zeit.

- **Vorteil**: Einfaches Wiederherstellen, da alle Daten in einem einzigen Backup enthalten sind.
- **Nachteil**: Hoher Speicher- und Zeitaufwand.

Beispiel mit `rsync`:

```
1 rsync -av --progress /quelle/ /backup/vollbackup/
```

‘-a‘ (archive) Aktiviert den Archivmodus, wodurch Dateien rekursiv kopiert und die wichtigsten Attribute (z. B. Rechte, Besitzer, Zeitstempel) beibehalten werden.

‘-v‘ (verbose) Gibt detaillierte Informationen über den Fortschritt aus.

‘–progress‘ Zeigt den Fortschritt für jede Datei an.

1.2.2 Inkrementelles Backup

Ein inkrementelles Backup speichert nur die Änderungen, die seit dem letzten Backup (egal welcher Art) vorgenommen wurden. Es benötigt weniger Speicherplatz und Zeit als ein Vollbackup.

- **Vorteil**: Spart Speicherplatz und Zeit.
- **Nachteil**: Wiederherstellung ist komplexer, da alle inkrementellen Backups seit dem letzten Vollbackup benötigt werden.

Beispiel mit `rsync` und Nutzung eines Zeitstempels:

```
rsync -av --progress --link-dest=/backup/vollbackup/ /quelle/ /backup/inkrementell
   /$(date +%Y%m%d)/
```

1.2.3 Differenzielles Backup

Ein differenzielles Backup speichert alle Änderungen seit dem letzten Vollbackup. Es bietet eine Kompromisslösung zwischen Voll- und inkrementellem Backup.

- **Vorteil**: Schneller als ein Vollbackup, aber weniger aufwändig als inkrementelle Backups.
- **Nachteil**: Kann mit der Zeit speicherintensiv werden, da alle Änderungen seit dem letzten Vollbackup enthalten sind.

Beispiel mit `rsync`:

```
rsync -av --progress --link-dest=/backup/vollbackup/ /quelle/ /backup/
   differenziell/$(date +%Y%m%d)/
```

- '–link-dest=<Pfad>': Verwendet eine Referenz auf ein vorheriges Backup, um Hardlinks zu erstellen. Diese Methode spart Speicherplatz, da unveränderte Dateien nicht erneut kopiert werden.

1.3 Praktische Hinweise

- Vor einem Backup sollte geprüft werden, ob ausreichend Speicherplatz zur Verfügung steht.
- Automatisierungen mit `cron` oder `systemd` Timer können regelmäßige Backups sicherstellen.
- Es empfiehlt sich, die Backups regelmäßig zu testen, um sicherzustellen, dass die Wiederherstellung im Ernstfall funktioniert.

1.4 Beispiel für ein Backup-Schema mit systemd-Timer

Ein sinnvolles Backup-Schema könnte wie folgt aussehen:

- **Täglich**: Inkrementelle Backups.
- **Wöchentlich**: Differenzielle Backups.
- **Monatlich**: Vollbackups.

Dazu richten wir mit `systemd` einen Service und einen Timer ein.

1.4.1 Schritt 1: Backup-Skript erstellen

Erstellen Sie ein Skript, das den Backup-Vorgang ausführt:

```
#!/bin/bash
BACKUP_DIR="/backup"
SOURCE_DIR="/quelle"

# Datum im Format YYYYMMDD
DATE=$(date +%Y%m%d)

# Backup-Art je nach Argument
case "$1" in
  full)
    rsync -av --progress "$SOURCE_DIR/" "$BACKUP_DIR/full/"
    ;;
  differential)
    rsync -av --progress --link-dest="$BACKUP_DIR/full/" "$SOURCE_DIR/" "
       $BACKUP_DIR/differential/$DATE/"
    ;;
  incremental)
    rsync -av --progress --link-dest="$BACKUP_DIR/differential/" "$SOURCE_DIR/" "
       $BACKUP_DIR/incremental/$DATE/"
    ;;
  *)
    echo "Usage: $0 {full|differential|incremental}"
    exit 1
    ;;
esac
```

Stellen Sie sicher, dass das Skript ausführbar ist:

```
chmod +x /usr/local/bin/backup.sh
```

1.4.2 Schritt 2: systemd-Service erstellen

Erstellen Sie eine Datei `/etc/systemd/system/backup.service`:

```
[Unit]
Description=Backup Service

[Service]
Type=oneshot
ExecStart=/usr/local/bin/backup.sh %i
```

1.4.3 Schritt 3: Timer für Backups erstellen

Erstellen Sie drei Timer-Dateien für die verschiedenen Backup-Typen.

1. Täglicher inkrementeller Timer: `/etc/systemd/system/backup@incremental.timer`

```
[Unit]
Description=Daily Incremental Backup Timer

[Timer]
OnCalendar=daily
Persistent=true

```

```
[Install]
WantedBy=timers.target
```

1. Wöchentlicher differenzieller Timer: `/etc/systemd/system/backup@differential.timer`

```
[Unit]
Description=Weekly Differential Backup Timer

[Timer]
OnCalendar=weekly
Persistent=true

[Install]
WantedBy=timers.target
```

1. Monatlicher Vollbackup-Timer: `/etc/systemd/system/backup@full.timer`

```
[Unit]
Description=Monthly Full Backup Timer

[Timer]
OnCalendar=monthly
Persistent=true

[Install]
WantedBy=timers.target
```

1.4.4 Schritt 4: Timer aktivieren

Aktivieren Sie die Timer:

```
systemctl enable --now backup@incremental.timer
systemctl enable --now backup@differential.timer
systemctl enable --now backup@full.timer
```

1.5 Moderne Backup-Tools unter Linux

Neben `rsync` gibt es eine Vielzahl moderner Tools, die speziell für Backups entwickelt wurden und viele zusätzliche Funktionen bieten. Einige Beispiele:

1.5.1 BorgBackup (`borg`)

- **Beschreibung**: Ein modernes deduplizierendes Backup-Tool, das effiziente und sichere Backups ermöglicht.
- **Funktionen**: Datenkomprimierung, Verschlüsselung und effiziente Speicherung durch Deduplizierung.
- **Installation**:

```
  sudo apt install borgbackup
```

1.5.2 Restic

- **Beschreibung**: Ein sicheres, schnelles und benutzerfreundliches Backup-Tool, das auf vielen Plattformen läuft.
- **Funktionen**: Verschlüsselung, Unterstützung für mehrere Speichersysteme (lokal, Cloud), inkrementelle Backups.
- **Installation**:

```
sudo apt install restic
```

1.5.3 Duplicity

- **Beschreibung**: Ein Backup-Tool, das Verschlüsselung und inkrementelle Backups mit Unterstützung für viele Remote-Speicherarten (z. B. Amazon S3) bietet.
- **Funktionen**: Verwendet GPG zur Verschlüsselung, ideal für Cloud-Backups.
- **Installation**:

```
sudo apt install duplicity
```