

# Backup-Dokumentation

## Zielsetzung

Diese Dokumentation beschreibt die Einrichtung eines **automatisierten Backup-Systems**, das Daten von mehreren Raspberry Pis auf einen zentralen Backup-Server sichert. Die Backups erfolgen mit `rsync` über SSH, sodass keine Passworteingabe erforderlich ist. Zusätzlich wird ein **Restore-Skript** verwendet, um die gesicherten Daten bei Bedarf wiederherzustellen.

## 1. Einrichtung des Backup-Systems

### ✂ 1.1 Backup-Benutzer anlegen

Um Sicherheit und Zugriffskontrolle zu gewährleisten, wird ein dedizierter Benutzer **backupuser** auf allen Systemen (Backup-Server und Raspberry Pis) erstellt:

- **Vermeidet Root-Zugriff für Backups.**
- **Ermöglicht SSH-Zugriff ohne Passwort.**
- **Hat nur Zugriff auf relevante Backup-Ordner.**

Befehle zur Einrichtung:

```
bash
KopierenBearbeiten
sudo useradd -m -s /bin/bash backupuser
sudo passwd backupuser
```

Nach der Erstellung muss **backupuser** Schreibrechte auf das Backup-Zielverzeichnis erhalten.

### 🔑 1.2 SSH-Zugang für **backupuser** einrichten

Damit `rsync` **automatisch** ohne Passwortabfrage funktioniert, wird **SSH-Schlüsselauthentifizierung** konfiguriert:

1. **SSH-Schlüssel generieren** (auf dem Backup-Server):

```
bash
KopierenBearbeiten
su - backupuser
ssh-keygen -t rsa -b 4096
```

2. **Öffentlichen Schlüssel auf die Raspberry Pis kopieren:**

```
bash
KopierenBearbeiten
ssh-copy-id backupuser@192.168.88.101
ssh-copy-id backupuser@192.168.88.102
```

### 3. SSH-Zugang testen:

```
bash
KopierenBearbeiten
ssh backupuser@192.168.88.101 "ls /home/backupuser"
```

Wenn die Verbindung **ohne Passwortabfrage** funktioniert, ist die Einrichtung korrekt.

## 1.3 Verzeichnisstruktur für das Backup

Das Backup wird auf einem **USB-Stick** oder einem **zentralen Backup-Speicher** gesichert.  
Die Struktur:

```
bash
KopierenBearbeiten
/mnt/backupusb/
├─ raspberrypi_backup/
│   ├─ 192.168.88.101/
│   ├─ 192.168.88.102/
│   └─ 192.168.88.100/ (lokales System)
```

Jede IP-Adresse steht für ein Gerät, von dem Daten gesichert werden.

## 2. Automatisiertes Backup mit `rsync`

### 2.1 Ablauf des Backup-Prozesses

1. **Prüfen, ob der USB-Stick gemountet ist.**
2. **Daten von den definierten Pfaden der Raspberry Pis mit `rsync` sichern.**
3. **Backups in getrennten Ordnern für jede IP speichern.**
4. **Erfolgreiches Backup in eine Log-Datei schreiben.**
5. **Cronjob einrichten, um Backups regelmäßig auszuführen.**

Das Backup-Skript ist als separate Datei in GitHub abgelegt.

### 2.2 Automatisierung mit Cronjob

Um das Backup **täglich um 3 Uhr nachts** auszuführen, wird ein **Cronjob für `backupuser`** eingerichtet:

```
bash
KopierenBearbeiten
```

```
crontab -e
```

Einfügen der Zeile:

```
bash
KopierenBearbeiten
0 3 * * * /pfad/zum/backup_script.sh
```

Damit läuft das Backup **vollautomatisch**.

## 3. Wiederherstellung der Backups

### 3.1 Ablauf des Restore-Prozesses

1. **Prüfen, ob der USB-Stick gemountet ist.**
2. **Falls nötig, Zielverzeichnisse erstellen.**
3. **Gesicherte Daten mit `rsync` auf die Raspberry Pis zurückspielen.**
4. **Erfolgreiche Wiederherstellung ins Log schreiben.**

Das Restore-Skript ist ebenfalls als separate Datei in GitHub abgelegt.

### ✂ 3.2 Manuelle Wiederherstellung testen

Falls ein bestimmtes Backup manuell zurückgespielt werden soll, kann dies mit `rsync` erfolgen:

```
bash
KopierenBearbeiten
rsync -avz /mnt/backupusb/raspberrypi_backup/192.168.88.101/
/home/backupuser/
```

Falls der Restore per Skript fehlschlägt, hilft die Log-Datei zur Fehleranalyse:

```
bash
KopierenBearbeiten
cat /var/log/restore/restore_backup.log
```

## 4. Sicherheit & Überwachung

### 4.1 Schutz des Backup-Systems

- `backupuser` hat **keine Root-Rechte**, um unbefugte Änderungen zu vermeiden.
- Das Backup-Verzeichnis `/mnt/backupusb/raspberrypi_backup/` hat eingeschränkte Zugriffsrechte:

```
bash
KopierenBearbeiten
```

```
sudo chmod -R 770 /mnt/backupusb/raspberrypi_backup/  
sudo chown -R backupuser:backupuser  
/mnt/backupusb/raspberrypi_backup/
```

- Der SSH-Zugang ist auf backupuser **beschränkt** (/etc/ssh/sshd\_config):

```
nginx  
KopierenBearbeiten  
AllowUsers backupuser
```

## 4.2 Logs & Monitoring

Um Backups und Restores zu überwachen:

### 1. **Log-Dateien regelmäßig prüfen:**

```
bash  
KopierenBearbeiten  
tail -f /var/log/backup/backup.log  
tail -f /var/log/restore/restore_backup.log
```

2. **E-Mail-Benachrichtigungen für erfolgreiches Backup einrichten.**
3. **Optional: Log-Rotation aktivieren, um alte Log-Dateien zu archivieren.**