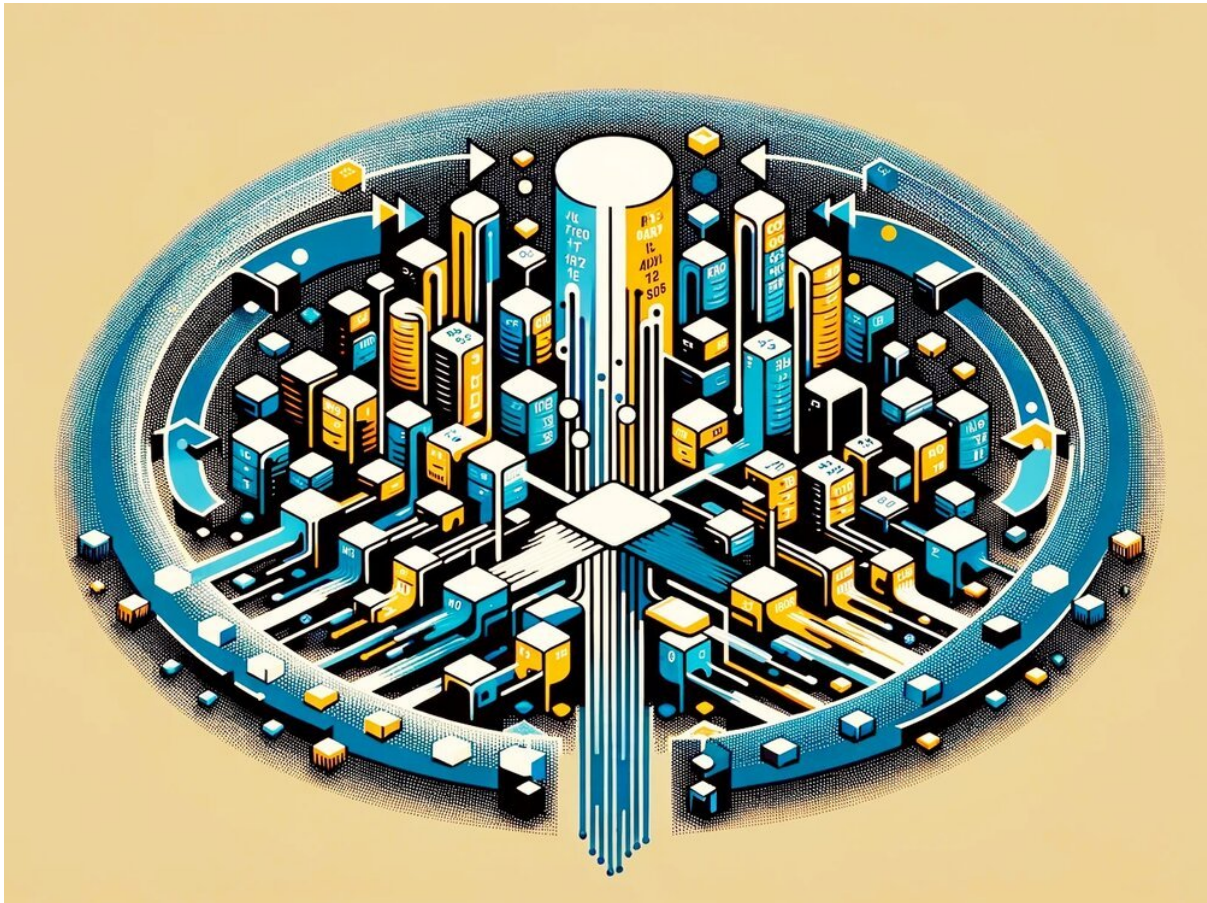


# Sicherungsschema

## *Rotation Ihrer Backups mit 'rsync-backup-rotator'*

Willem L. Middelkoop

Dec. 15, 2023

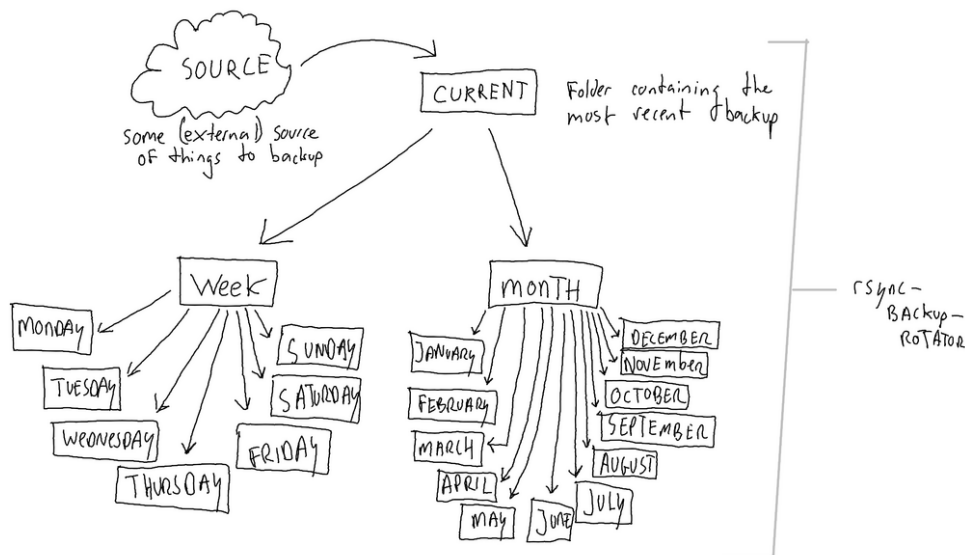


Im heutigen digitalen Zeitalter ist der Schutz Ihrer Daten von größter Bedeutung. Das einfache Erstellen einer Kopie Ihrer Dateien reicht möglicherweise nicht aus, da diese beschädigt, überschrieben oder durch Ransomware blockiert werden können. Mehrere, zeitlich rotierte (und idealerweise externe) Backups bieten einen stärkeren Schutz. Ich habe ein neues Tool namens `rsync-backup-rotator` entwickelt, um Sie dabei zu unterstützen.

### Backup-Rotationsschema

Backup-Rotation ist eine Strategie in der Datenverwaltung, bei der mehrere Sicherungskopien in unterschiedlichen Intervallen erstellt und gespeichert werden, anstatt sich auf eine einzige Sicherungskopie zu verlassen. Diese Methode ist besonders nützlich, da sie verschiedene Risiken im Zusammenhang mit Datenverlust mindert. Wenn beispielsweise

eine einzelne Sicherungskopie beschädigt, überschrieben oder kompromittiert wird (z. B. durch Ransomware), könnten alle Daten seit der letzten Sicherung verloren gehen. Durch die Rotation von Backups erstellen Sie mehrere Wiederherstellungspunkte, die es Ihnen ermöglichen, Daten von verschiedenen Zeitpunkten wiederherzustellen. Dieser Ansatz bietet ein umfassenderes Sicherheitsnetz, da er sowohl vor kürzlich aufgetretenen Datenverlusten als auch vor längerfristigen Problemen schützt.



*Rotation von Backups zur Ermöglichung der Datenwiederherstellung von mehreren Zeitpunkten - Zeitreisen für Daten!*

Das Tool **'rsync-backup-rotator'** verwendet dieses Konzept der Backup-Rotation. Es automatisiert den Prozess der Erstellung und Verwaltung dieser rotierten Backups. Konkret verwendet das Tool einen zentralen Ordner „current“, der die neuesten Backup-Daten enthält. Basierend auf benutzerdefinierten Einstellungen rotiert es diese Daten dann in verschiedene Ordner. Für tägliche Backups erstellt und speichert es Kopien in Unterordnern, die nach jedem Wochentag benannt sind, innerhalb eines Ordners „week“. Dies bedeutet, dass es für jeden Tag eine separate Sicherung gibt, z. B. „Monday“, „Tuesday“ usw. Für monatliche Backups erstellt es am ersten Tag jedes Monats einen Snapshot in einem entsprechenden monatlichen Unterordner innerhalb eines Ordners „month“, z. B. „January“, „February“ usw. Dieses System stellt sicher, dass Benutzer über eine Reihe von zeitgestempelten Backups verfügen, die Flexibilität und Sicherheit bei ihren Datenwiederherstellungsoptionen bieten.

## Warum rsync?

Rsync ist ein Tool, das Backups sowohl effizient als auch effektiv macht. Es aktualisiert nur die Teile von Dateien, die sich seit der letzten Sicherung geändert haben, anstatt alles erneut zu kopieren. Dieser Ansatz reduziert die Menge der übertragenen Daten erheblich, schont Festplatten und SSDs und senkt die Kosten für Cloud-Speicher, da weniger Datenänderungen weniger I/O-Operationen bedeuten. Im Wesentlichen stellt rsync sicher, dass Backups schnell und ressourcenschonend sind, was es zu einer intelligenten Wahl für regelmäßige Datensicherungen und -rotationen macht.

**Rsync** ist ein standardmäßiges, nicht-kommerzielles Tool, das in viele Betriebssysteme wie GNU/Linux, BSD-Varianten und macOS integriert ist, wodurch es ohne zusätzliche Installationen oder Käufe leicht zugänglich ist. Seine robusten und effizienten Datensicherungsfunktionen, denen Millionen weltweit vertrauen, haben es zu einem zuverlässigen und weit verbreiteten Industriestandard gemacht.

## Erstellen von Backups mit Rsync

Um eine einzelne Kopie mit `rsync` zu erstellen, können Sie den Befehl **`rsync -arvz SOURCE TARGET`** verwenden. In einem früheren Beitrag habe ich erklärt, wie man `rsync` für Backups verwendet, den Sie hier finden: [How to Use Rsync to Make Backups](#). Es ist wichtig, sorgfältig zu entscheiden, was Sie sichern möchten, ob es sich um einen einfachen Dokumentenordner oder eine gesamte Installation handelt. Das Tool **`rsync-backup-rotator`**, wie in diesem Blogbeitrag erläutert, geht davon aus, dass Sie über einen funktionierenden Kopiermechanismus verfügen. Dieser Mechanismus sollte eine aktuelle Kopie Ihrer Dateien in einem Ordner namens „current“ verwalten, der sich in einem für das Tool zugänglichen Pfad befindet, wodurch eine effiziente Rotation ermöglicht wird.

## Das Tool: `rsync-backup-rotator`

Wie bei meinen Uhren, [hier](#) und [hier](#), bevorzuge ich zeitlose, selbsterklärende, einfache und zweckmäßige Tools. Der **`rsync-backup-rotator`** wurde mit demselben Design-Ethos erstellt.

Das Tool ist ein Bash-Skript, das ohne seltsame Abhängigkeiten oder Bloatware auf so ziemlich allen GNU/Linux-, BSD- und macOS-Systemen läuft. Wahrscheinlich können Sie es sogar unter Windows mit dem WSL zum Laufen bringen. Es ist so konzipiert, dass es seine Aufgabe jetzt und in Zukunft ohne Wartung oder obligatorische Updates erfüllt.

Sie können das Tool hier herunterladen: <https://source.willem.com/rsync-backup-rotator/>. Dort finden Sie auch [Anweisungen](#) und ein [Changelog](#). Das Tool wird als freie Software unter der [GPLv3-Lizenz](#) vertrieben, um Ihre Freiheit zu gewährleisten, alle Versionen dieses Tools zu verwenden, zu teilen und zu verändern.

## Installation

- Laden Sie das Skript von <https://source.willem.com/rsync-backup-rotator/> herunter
- Machen Sie das Skript ausführbar: **`chmod +x rsync-backup-rotator.sh`**

## Benutzung

Um `rsync-backup-rotator` zu verwenden, führen Sie das Skript mit den erforderlichen Argumenten aus: **`./rsync-backup-rotator.sh [options]`**

## Optionen:

- `-w`: Aktiviert Backups für jeden Wochentag.
- `-m`: Aktiviert Backups für den ersten Tag jedes Monats.
- `--delete`: Fügt die Option `--delete` in `rsync` ein, um Dateien im Ziel zu entfernen, die sich nicht mehr in der Quelle befinden.
- `--help`: Zeigt Hilfeinformationen an.

## Beispiele:

- Führen Sie ein Wochentags-Backup durch: `‘./rsync-backup-rotator.sh /path/to/parent -w‘`
- Führen Sie sowohl Wochentags- als auch Monats-Backups mit Löschen entfernter Dateien durch: `‘./rsync-backup-rotator.sh /path/to/parent -w -m --delete‘`

## Fazit

In Zukunft beabsichtige ich, weitere Tools und Anwendungen unter der GPLv3-Lizenz als freie Software zu veröffentlichen. Obwohl dies ein kleiner Anfang ist, ist es mein bescheidener Versuch, der Welt etwas zurückzugeben. Bleiben Sie sicher, bleiben Sie gesichert!

```

1 #!/bin/bash
2 #
3 # rsync-backup-rotator
4 # Version: 1.0.0
5 #
6 # Copyright (C) 2023 Willem L. Middelkoop (mail@willem.com)
7 #
8 # This program is free software: you can redistribute it and/or modify
9 # it under the terms of the GNU General Public License as published by
10 # the Free Software Foundation, either version 3 of the License, or
11 # (at your option) any later version.
12 #
13 # This program is distributed in the hope that it will be useful,
14 # but WITHOUT ANY WARRANTY; without even the implied warranty of
15 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16 # GNU General Public License for more details.
17 #
18 # You should have received a copy of the GNU General Public License
19 # along with this program. If not, see <https://www.gnu.org/licenses/>.
20 #
21 # For more information, visit: https://source.willem.com/rsync-backup-rotator/
22
23 # Function to show help/usage
24 show_help() {
25     echo "Usage: $0 parent_directory [-w] [-m] [--delete] [--help]"
26     echo
27     echo "Options:"
28     echo "  parent_directory  Mandatory. The parent directory for 'current' backups."
29     echo "  -w                Optional. Enable backups for each weekday."
30     echo "  -m                Optional. Enable backups for the first day of each month."
31     echo "  --delete          Optional. Include the --delete option in rsync to remove files in t
32
33     echo "  --help            Show this help message and exit."
34     echo
35     echo "Example:"
36     echo "  $0 /path/to/parent -wm --delete"
37 }
38
39 # Function to perform rsync
40 do_rsync() {
41     local source=$1
42     local destination=$2
43     # Ensure the destination directory exists
44     mkdir -p "$destination"
45     local rsync_opts="-arvz"
46     if [ "$DELETE_ENABLED" = true ]; then
47         rsync_opts="$rsync_opts --delete"
48     fi
49     rsync $rsync_opts "$source" "$destination"
50 }
51
52 # Function to handle daily backups
53 daily_backup() {
54     if [ "$ENABLE_WEEKDAY_BACKUP" = true ]; then
55         local weekday=$(date +%A)
56         local dest_week="$PARENT_DIR/week/$weekday/"
57         do_rsync "$SOURCE_PATH" "$dest_week"
58     fi
59 }
60
61 # Function to handle monthly backups
62 monthly_backup() {
63     if [ "$ENABLE_MONTHLY_BACKUP" = true ]; then
64         local month=$(date +%B)
65         local day=$(date +%d)
66         if [ "$day" -eq 1 ]; then
67             local dest_month="$PARENT_DIR/month/$month/"
68             do_rsync "$SOURCE_PATH" "$dest_month"
69         fi
70     fi
71 }
72
73 # Check for --help option first
74 for arg in "$@"; do
75     if [ "$arg" = "--help" ]; then
76         show_help
77         exit 0
78     fi
79 done
80
81 # Check if a parent directory is provided
82 if [ $# -lt 1 ]; then
83     echo "Error: No parent directory provided."
84     show_help
85     exit 1
86 fi
87
88 # Parsing command-line arguments
89 PARENT_DIR=$1
90 shift # Removing the first argument as it's the parent directory
91 ENABLE_WEEKDAY_BACKUP=false
92 ENABLE_MONTHLY_BACKUP=false
93 DELETE_ENABLED=false
94 SOURCE_PATH="$PARENT_DIR/current"
95
96 while (( $# )); do
97     case $1 in
98         -w) ENABLE_WEEKDAY_BACKUP=true;;
99         -m) ENABLE_MONTHLY_BACKUP=true;;
100         --delete) DELETE_ENABLED=true;;
101         *) echo "Invalid option $1" >&2; exit 1;;
102     esac

```