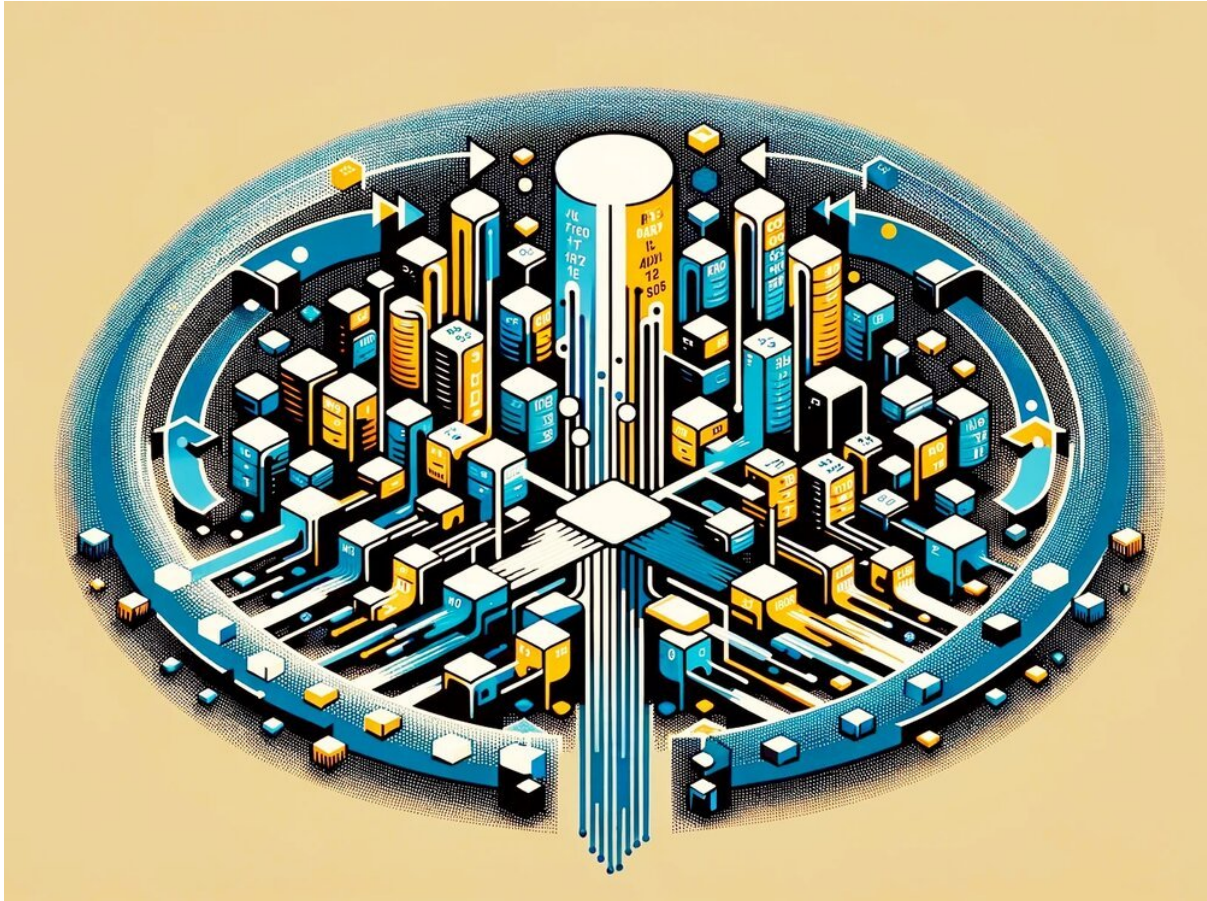


Back-uprotatieschema

Roteer je backups met 'rsync-backup-rotator'

Willem L. Middelkoop

15 dec. 2023

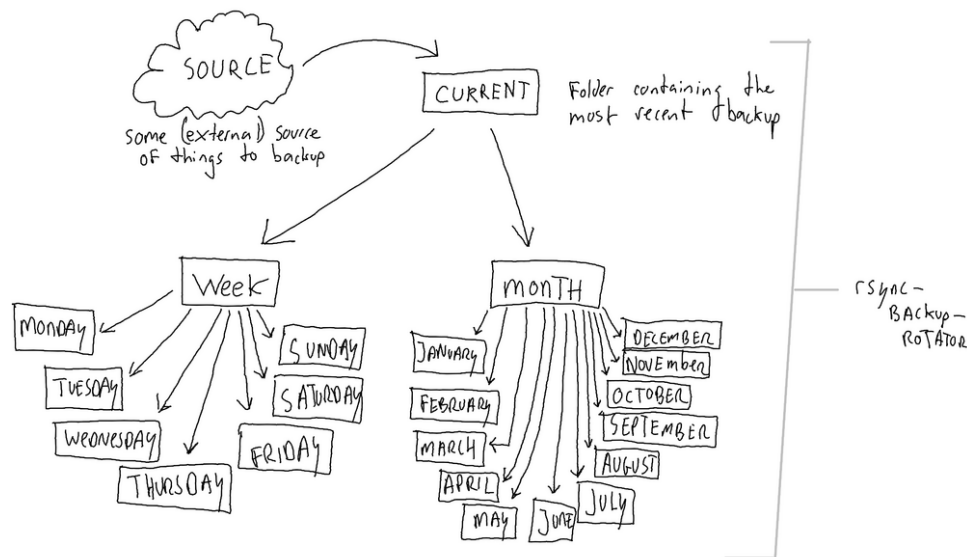


In het digitale tijdperk van vandaag is het beschermen van je gegevens van het grootste belang. Alleen een kopie van je bestanden maken is misschien niet voldoende, omdat ze beschadigd, overschreven of geblokkeerd kunnen raken door ransomware. Meerdere, tijdgeroteerde (en idealiter offsite) backups vormen een sterkere verdediging. Ik heb een nieuwe tool gemaakt, rsync-backup-rotator, om je hierbij te helpen.

Backup Rotatieschema

Backup rotatie is een strategie die gebruikt wordt in databeheer waarbij meerdere backup kopieën worden gemaakt en opgeslagen met verschillende intervallen, in plaats van te vertrouwen op een enkele backup kopie. Deze methode is bijzonder nuttig omdat het verschillende risico's in verband met dataverlies vermindert. Als bijvoorbeeld een enkele

backup kopie beschadigd, overschreven of gecompromitteerd raakt (bijv. door ransomware), kunnen alle gegevens sinds de laatste backup verloren gaan. Door backups te roteren, creëer je meerdere herstelpunten, waardoor je gegevens van verschillende momenten in de tijd kunt herstellen. Deze aanpak biedt een uitgebreider vangnet, omdat het beschermt tegen zowel recent dataverlies als problemen op langere termijn.



Rotatieback-up om gegevensherstel vanaf meerdere punten in de tijd mogelijk te maken - tijdreizen voor data!

De tool **'rsync-backup-rotator'** omarmt dit concept van backup rotatie. Het automatiseert het proces van het maken en beheren van deze geroteerde backups. De tool gebruikt specifiek een centrale 'current' map die de nieuwste backupgegevens bevat. Op basis van door de gebruiker gedefinieerde instellingen roteert het vervolgens deze gegevens naar verschillende mappen. Voor dagelijkse backups maakt en bewaart het kopieën in submappen die vernoemd zijn naar elke weekdag, binnen een 'week' map. Dit betekent dat er een aparte backup is voor elke dag, zoals 'Maandag', 'Dinsdag', enzovoort. Voor maandelijkse backups, op de eerste dag van elke maand, maakt het een snapshot in een overeenkomstige maandelijkse submap binnen een 'maand' map, zoals 'Januari', 'Februari', etc. Dit systeem zorgt ervoor dat gebruikers een reeks backups met tijdstempel hebben, wat flexibiliteit en veiligheid biedt in hun dataherstel opties.

Waarom rsync?

Rsync is een tool die backups zowel efficiënt als effectief maakt. Het werkt door alleen de delen van bestanden bij te werken die zijn gewijzigd sinds de laatste backup, in plaats van alles opnieuw te kopiëren. Deze aanpak vermindert de hoeveelheid te verzenden gegevens aanzienlijk, wat slijtage van harde schijven en SSD's bespaart en de kosten voor cloudopslag verlaagt, aangezien minder datawijzigingen minder I/O-bewerkingen betekenen. In wezen zorgt rsync ervoor dat backups snel en licht zijn op resources, waardoor het een slimme keuze is voor regelmatige databackup en rotatie.

Rsync is een standaard, niet-commerciële tool die is ingebouwd in veel besturingssystemen zoals GNU/Linux, BSD-varianten en macOS, waardoor het gemakkelijk toegankelijk is zonder extra installaties of aankopen. Vertrouwd door miljoenen wereldwijd, hebben

de robuuste en efficiënte databackup mogelijkheden het gevestigd als een betrouwbare en veelgebruikte industriestandaard.

Backups maken met Rsync

Om een enkele kopie te maken met rsync, kun je het commando `'rsync -arvz BRON DOEL'` gebruiken. In een eerder bericht heb ik uitgelegd hoe je rsync kunt gebruiken voor het maken van backups, die je hier kunt vinden: [How to Use Rsync to Make Backups](#). Het is belangrijk om zorgvuldig te beslissen wat je wilt back-uppen, of het nu een eenvoudige documentenmap of een volledige installatie is. De `'rsync-backup-rotator'` tool, zoals besproken in deze blogpost, werkt onder de aanname dat je een functioneel kopieermechanisme hebt. Dit mechanisme moet een up-to-date kopie van je bestanden bijhouden in een map met de naam "current", die zich bevindt op een pad dat toegankelijk is voor de tool, waardoor efficiënte rotatie mogelijk wordt.

De tool: rsync-backup-rotator

Net als mijn horloges, [hier](#) en [hier](#), geef ik er de voorkeur aan dat mijn tools tijdloos, zelfverklarend, eenvoudig en toegewijd aan hun taak zijn. De `rsync-backup-rotator` is gemaakt met dezelfde ontwerpfilosofie.

De tool is een bash script dat zonder rare afhankelijkheden of bloat draait op vrijwel alle GNU/Linux, BSD en macOS systemen. Je kunt het waarschijnlijk ook op Windows aan de praat krijgen met behulp van de WSL. Het is ontworpen om zijn werk nu en in de toekomst te doen, zonder onderhoud of verplichte updates.

Je kunt de tool hier downloaden: <https://source.willem.com/rsync-backup-rotator/>. Daar vind je ook [instructies](#) en een [changelog](#). De tool wordt gedistribueerd als vrije software onder de [GPLv3](#) licentie, bedoeld om je vrijheid te garanderen om alle versies van deze tool te gebruiken, te delen en te wijzigen.

Installatie

- Download het script van <https://source.willem.com/rsync-backup-rotator/>
- Maak het script uitvoerbaar: `'chmod +x rsync-backup-rotator.sh'`

Gebruik

Om `rsync-backup-rotator` te gebruiken, voer je het script uit met de vereiste argumenten: `./rsync-backup-rotator.sh [opties]`

Opties:

- `-w`: Schakel backups in voor elke weekdag.
- `-m`: Schakel backups in voor de eerste dag van elke maand.
- `--delete`: Voeg de `--delete` optie toe aan rsync om bestanden in de bestemming te verwijderen die niet langer in de bron staan.
- `--help`: Toon helpinformatie.

Voorbeelden:

- Voer een weekdag backup uit: `./rsync-backup-rotator.sh /pad/naar/bovenliggende_map -w`
- Voer zowel weekdag- als maandelijkse backups uit met verwijdering van verwijderde bestanden: `./rsync-backup-rotator.sh /pad/naar/bovenliggende_map -w -m --delete`

Conclusie

In de toekomst ben ik van plan om meer tools en applicaties uit te brengen onder de GPLv3 licentie als vrije software. Hoewel dit een klein begin is, is het mijn bescheiden poging om iets terug te geven aan de wereld. Blijf veilig, blijf geback-upt!

```

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
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

```