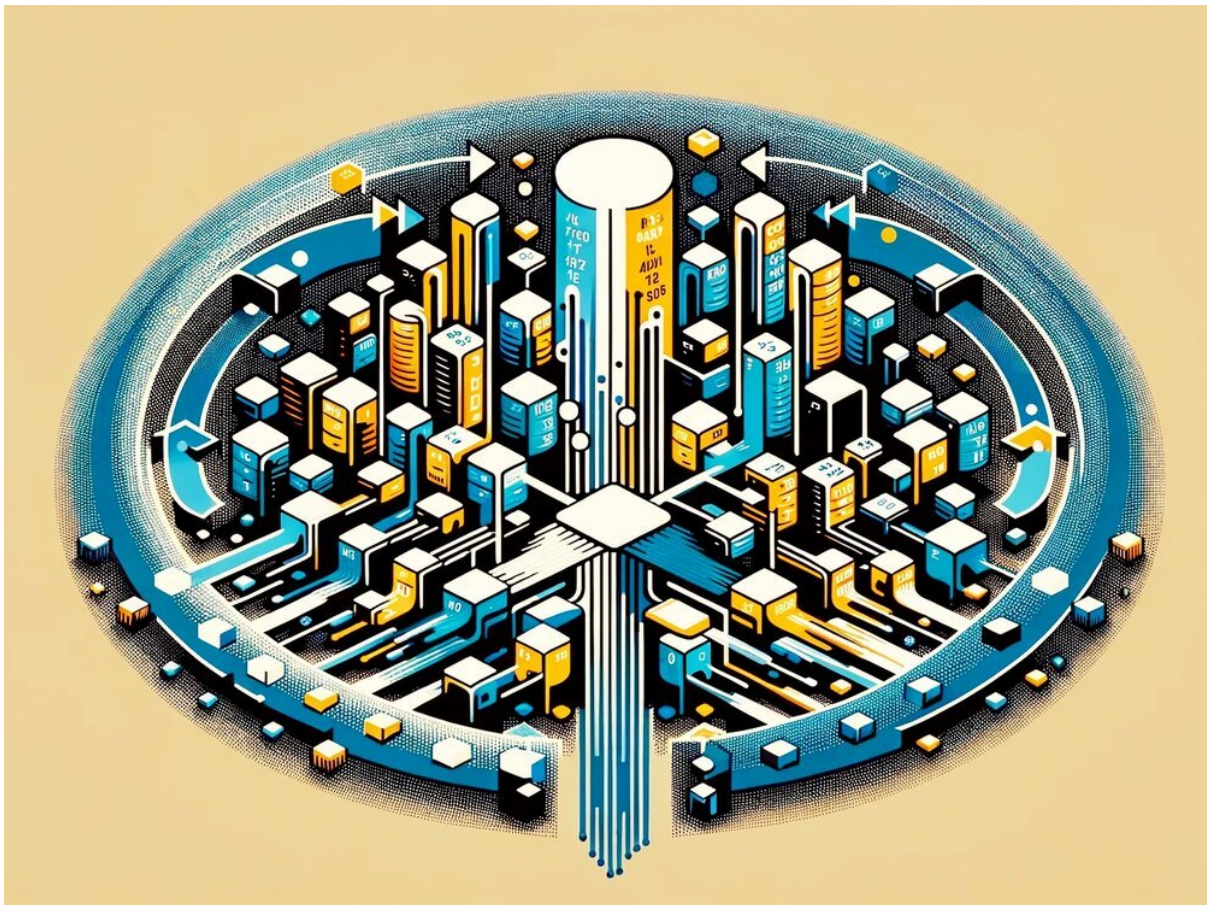


Esquema de rotación de copias de seguridad

Rota tus copias de seguridad con 'rsync-backup-rotator'

Willem L. Middelkoop

Dec. 15, 2023

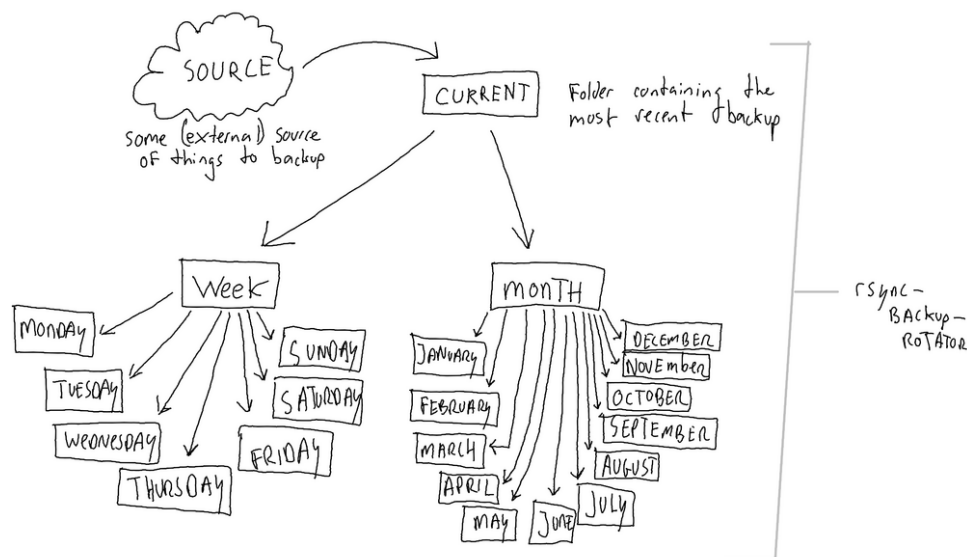


En la era digital actual, proteger sus datos es primordial. Simplemente crear una copia de sus archivos puede no ser suficiente, ya que pueden corromperse, sobrescribirse o bloquearse por ransomware. Tener múltiples copias de seguridad, rotadas en el tiempo (e idealmente, fuera del sitio) es una defensa más sólida. He creado una nueva herramienta, `rsync-backup-rotator`, para ayudarlo con esto.

Esquema de rotación de copias de seguridad

La rotación de copias de seguridad es una estrategia utilizada en la gestión de datos donde se crean y almacenan múltiples copias de seguridad a diferentes intervalos, en lugar de

depender de una sola copia. Este método es particularmente útil porque mitiga varios riesgos asociados con la pérdida de datos. Por ejemplo, si una sola copia de seguridad se corrompe, se sobrescribe o se ve comprometida (por ejemplo, por ransomware), se podrían perder todos los datos desde la última copia de seguridad. Al rotar las copias de seguridad, se crean múltiples puntos de recuperación, lo que permite restaurar los datos desde diferentes momentos en el tiempo. Este enfoque proporciona una red de seguridad más completa, ya que protege contra la pérdida de datos reciente y los problemas a más largo plazo.



Rotación de copias de seguridad para permitir la recuperación de datos desde múltiples puntos en el tiempo - ¡permitiendo viajar en el tiempo para los datos!

La herramienta '**rsync-backup-rotator**' adopta este concepto de rotación de copias de seguridad. Automatiza el proceso de creación y gestión de estas copias de seguridad rotadas. Específicamente, la herramienta utiliza una carpeta central 'actual' que contiene los datos de la copia de seguridad más reciente. En función de la configuración definida por el usuario, rota estos datos a diferentes carpetas. Para las copias de seguridad diarias, crea y almacena copias en subcarpetas con el nombre de cada día de la semana, dentro de una carpeta 'semana'. Esto significa que hay una copia de seguridad separada para cada día, como 'Lunes', 'Martes', etc. Para las copias de seguridad mensuales, el primer día de cada mes, crea una instantánea en una subcarpeta mensual correspondiente dentro de una carpeta 'mes', como 'Enero', 'Febrero', etc. Este sistema garantiza que los usuarios tengan una serie de copias de seguridad con marca de tiempo, lo que proporciona flexibilidad y seguridad en sus opciones de restauración de datos.

¿Por qué rsync?

Rsync es una herramienta que hace que las copias de seguridad sean eficientes y efectivas. Funciona actualizando solo las partes de los archivos que han cambiado desde la última copia de seguridad, en lugar de copiar todo de nuevo. Este enfoque reduce significativamente la cantidad de datos que se transfieren, lo que ahorra desgaste en los discos duros y SSD, y reduce los costos del almacenamiento en la nube, ya que menos cambios de datos significan menos operaciones de E/S. Esencialmente, rsync garantiza que las copias de

seguridad sean rápidas y ligeras en recursos, lo que la convierte en una opción inteligente para la copia de seguridad y rotación de datos regulares.

Rsync es una herramienta estándar, no comercial, integrada en muchos sistemas operativos como GNU/Linux, variantes de BSD y macOS, lo que la hace fácilmente accesible sin instalaciones ni compras adicionales. Con la confianza de millones de personas en todo el mundo, sus sólidas y eficientes capacidades de copia de seguridad de datos la han establecido como un estándar de la industria confiable y ampliamente utilizado.

Creación de copias de seguridad con Rsync

Para crear una sola copia usando rsync, puede usar el comando **‘rsync -arvz ORIGEN DESTINO’**. En una publicación anterior, expliqué cómo usar rsync para hacer copias de seguridad, que puede encontrar aquí: [How to Use Rsync to Make Backups](#). Es importante decidir cuidadosamente qué desea respaldar, ya sea una simple carpeta de documentos o una instalación completa. La herramienta **‘rsync-backup-rotator’**, como se describe en esta publicación de blog, opera bajo el supuesto de que tiene un mecanismo de copia funcional en su lugar. Este mecanismo debe mantener una copia actualizada de sus archivos en una carpeta llamada "actual", ubicada en una ruta accesible por la herramienta, lo que permite una rotación eficiente.

La herramienta: rsync-backup-rotator

Al igual que mis relojes, [aquí](#) y [aquí](#), prefiero que mis herramientas sean atemporales, autoexplicativas, simples y dedicadas a su trabajo. El **rsync-backup-rotator** se creó con el mismo espíritu de diseño.

La herramienta es un script bash que se ejecuta sin dependencias extrañas ni bloatware en prácticamente todos los sistemas GNU/Linux, BSD y macOS. Incluso, probablemente pueda hacer que funcione en Windows también, usando el WSL. Está diseñado para hacer su trabajo ahora y en el futuro, sin ningún mantenimiento ni actualizaciones obligatorias.

Puede descargar la herramienta aquí: <https://source.willem.com/rsync-backup-rotator/>. Allí encontrará [instrucciones](#) y un registro de [cambios](#) también. La herramienta se distribuye como software libre bajo la licencia [GPLv3](#), con la intención de garantizar su libertad para usar, compartir y cambiar todas las versiones de esta herramienta.

Instalación

- Descargue el script desde <https://source.willem.com/rsync-backup-rotator/>
- Haga que el script sea ejecutable: **‘chmod +x rsync-backup-rotator.sh’**

Uso

Para usar rsync-backup-rotator, ejecute el script con los argumentos requeridos: **./rsync-backup-rotator.sh [opciones]**

Opciones:

- **-w**: Habilitar copias de seguridad para cada día de la semana.
- **-m**: Habilitar copias de seguridad para el primer día de cada mes.

- `--delete`: Incluir la opción `--delete` en `rsync` para eliminar los archivos en el destino que ya no están en el origen.
- `--help`: Mostrar información de ayuda.

Ejemplos:

- Realizar una copia de seguridad de un día de la semana: `‘./rsync-backup-rotator.sh /ruta/al/directorio_principal -w’`
- Realizar copias de seguridad diarias y mensuales con eliminación de archivos eliminados: `‘./rsync-backup-rotator.sh /ruta/al/directorio_principal -w -m -delete’`

Conclusión

En el futuro, tengo la intención de lanzar más herramientas y aplicaciones bajo la licencia GPLv3 como software libre. Aunque este es un pequeño comienzo, es mi humilde intento de devolver algo al mundo. ¡Manténgase seguro, mantenga sus datos respaldados!

```

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

```