

GUÍA PARA COPIAS DE SEGURIDAD

1. Requisitos previos

- Sistema de archivos raíz en **Btrfs**.
- Disco externo formateado a **Btrfs**.
- Software instalado:

- tmux
 - btrfs-progs
 - sudo

- Verificar sistema Btrfs:

```
1 $ lsblk -f | grep btrfs
2 $ mount | grep /mnt/backup
```

- Formatear el disco en btrfs:

```
1 $ mkfs.btrfs /dev/sdb1
```

- Crear un punto de montaje y montar:

```
1 $ mkdir -p /mnt/backup
2 $ mount /dev/sdb1 /mnt/backup
```

2. Backup completo en segundo plano

1. Definir *timestamp*

```
1 $ timestamp=$(date +%F_%H%M)
```

2. Crear *snapshot readonly* de la raíz

```
1 $ sudo btrfs subvolume snapshot -r / /.snapshots/backup-
    $timestamp
```

3. Iniciar *tmux*

```
1 $ tmux new -s backup
```

4. Enviar el *snapshot* al *backup* (primer *backup* completo)

```
1 $ sudo btrfs send /.snapshots/backup-$timestamp | sudo btrfs
    receive /mnt/backup
```

5. Desconectar *tmux* (sin parar): *ctrl+b* luego *d*

6. Comprobar subvolúmenes en *backup*:

```
1 $ sudo btrfs subvolume list /mnt/backup
```

7. (Opcional) Eliminar *snapshot* local para ahorrar espacio

```
1 $ sudo btrfs subvolume delete /.snapshots/backup-$timestamp
```

3. Backup Incremental

Se asume que el primer snapshot se llama backup-YYYY-MM-DD_HHMM.

1. Crear nuevo *snapshot*

```
1 $ new=$(date +%F_%H%M)
2 $ sudo btrfs subvolume snapshot -r / /.snapshots/backup-$new
```

2. Utilizar *tmux* o iniciar nuevo

```
1 $ tmux attach -t backup || tmux new -s backup
```

3. Enviar Incremental

```
1 $ sudo btrfs send -p /.snapshots/backup-$timestamp /.snapshots/
   backup-$new | sudo btrfs receive /mnt/backup
```

4. Actualizar referencia

```
1 $ timestamp=$new
```

5. (Opcional) Borrar *snapshot* local antiguo

```
1 $ sudo btrfs subvolume delete /.snapshots/backup-$timestamp
```

4. Supervisión en tiempo real

Dentro de tu sesión *tmux*:

```
1 $ watch -n 5 'sudo du -sh /mnt/backup/backup-*'
2 $ watch -n 5 'sudo btrfs filesystem df /mnt/backup'
```

5. Automatizar *backups* con *cron*

Para programar *backups* completos e incrementales todos los viernes a las 02:00 AM, hay que crear un *script* y añadirlo al *crontab*.

1. Crear el *script* de *backup*

Guardar el contenido siguiente en `/usr/local/bin/btrfs-backup.sh` y hacerlo ejecutable con `$ sudo chmod +x /usr/local/bin/btrfs-backup.sh`

```
1 #!/bin/bash
2 set -euo pipefail
3
4 # Variables
5 BACKUP_MNT="/mnt/backup"
6 DEVICE="/dev/sdf1"
7 SNAP_DIR="/.snapshots"
8 LOGFILE="/var/log/btrfs-backup.log"
9 TIMESTAMP=$(date +%F_%H%M)
10 CUR_SNAP="$SNAP_DIR/backup-$TIMESTAMP"
11
12 # Asegura montaje
13 if ! mountpoint -q "$BACKUP_MNT"; then
14     mount "$DEVICE" "$BACKUP_MNT"
15 fi
16
17 # Crear snapshot readonly
18 btrfs subvolume snapshot -r / "$CUR_SNAP"
19
20 # Encontrar ultimo backup (si existe)
21 LAST=$(ls -1 "$BACKUP_MNT" | sort | tail -n1 || true)
22
23 # Enviar backup (completo o incremental)
```

```
24 if [[ -n "$LAST" ]]; then
25     btrfs send -p "$BACKUP_MNT/$LAST" "$CUR_SNAP" | btrfs
        receive "$BACKUP_MNT"
26 else
27 btrfs send "$CUR_SNAP" | btrfs receive "$BACKUP_MNT"
28 fi
29
30 # Eliminar snapshot local antiguo (opcional)
31 btrfs subvolume delete "$CUR_SNAP"
32
33 # Log exitoso
34 echo "$(date): Backup $TIMESTAMP completed" >> "$LOGFILE"
```

2. Añadir al *crontab* de *root*

Editar el *crontab*:

```
1 $ sudo crontab -e
```

Agregar la línea siguiente para ejecutarlo todos los viernes a las 02:00:

```
1 0 2 * * 5 /usr/local/bin/btrfs-backup.sh
```

3. Monitorear el estado

Para revisar el log:

```
1 $ tail -n50 /var/log/btrfs-backup.log
```

Y para comprobar espacio en el pendrive:

```
1 btrfs filesystem df /mnt/backup
```