

Sauvegarde

Borgbackup

Pour effectuer des sauvegardes incrémentielles, j'utilise BorgBackup qui a l'avantage de pouvoir garder seulement X sauvegardes, de restaurer uniquement un fichier ou dossier en particulier et de monter la sauvegarde dans le système de fichier.

En prérequis, il faut bien sûr avoir installé le paquet `borgbackup`.

Voici mon script :

```
#!/bin/bash

# Définir le répertoire de sauvegarde
backup_dir="/mnt/disque1/borgbackup"

# Définir le nom de l'archive
archive_name=borg_backup

# Définir le nombre de sauvegardes à conserver
num_backups_to_keep=5

# Obtenir la date et l'heure actuelles
current_date_time=$(date +%d-%m-%Y_%H-%M-%S)

# Supprimer les sauvegardes excédentaires
borg prune --keep-last $num_backups_to_keep "${backup_dir}"

# Créer la sauvegarde incrémentielle en excluant les dossiers indiqués
borg create -v --stats --compression none \
  --exclude /swapfile \
  --exclude '/dev/*' \
  --exclude '/proc/*' \
  --exclude '/sys/*' \
```

```

--exclude '/tmp/*' \
--exclude '/run/*' \
--exclude '/mnt/*' \
--exclude '/media/*' \
--exclude '/lost+found' \
"${backup_dir}::${archive_name}_${current_date_time}" /

# https://doc.ubuntu-fr.org/borgbackup
# restauration
# borg extract --progress /chemin/vers/repertoire/de/sauvegarde::borg_backup_01-01-2021_12-00-00

# Montage de la sauvegarde
# borg mount /backup/borg::20230119 /tmp/backup
# Démontage
# borg umount /tmp/backup

```

Ne pas oublier de personnaliser selon les besoins et la configuration du système de fichier.
 Notez qu'à la fin du script, je me suis laissé des notes.
 Ne pas oublier d'initialiser le dossier de sauvegarde avec la commande suivante :

```
borg init -e none /chemin/vers/dossier
```

DD et Dump

`dd` va me permettre de faire une image de la partition EFI et `dump` pour faire une sauvegarde de la partition EXT4.

L'avantage d'utiliser `dump` est qu'il fait partie des outils pour l'EXT1-4 pour faire des sauvegardes et ne sauvegarde pas l'espace vide.

Le script comporte une copie sur un disque local supplémentaire, puis l'envoi sur un serveur FTP.

Voici le script simplifié :

```

#!/bin/bash

# serveur ftp
FTP_USER='user'
FTP_PASSWORD='password'
FTP_SERVER='serveur'
FTP_MOUNTPOINT='/point/de/montage/ftp'

```

```
FTP_BACKUP='/dossier/seur/serveur/ftp'

# Définir la source EFI
src_efi=/dev/sda1
dst_dir=${FTP_MOUNTPOINT}${FTP_BACKUP}
dst_image=${TRANSIT_D}/efi_image_$(date +%d-%m-%Y).img
error_log=${TRANSIT_D}/ERROR_image_$(date +%d-%m-%Y)

# Zone Transit avant l'envoi en ftp
TRANSIT_D=/mnt/usb1/backup

# création du dossier de montage

fusermount -u $FTP_MOUNTPOINT ;
rm -r $FTP_MOUNTPOINT ;
mkdir $FTP_MOUNTPOINT

# Montage FTP

curlftps $FTP_SERVER $FTP_MOUNTPOINT -o user=$FTP_USER:$FTP_PASSWORD

# Copie avec dd pour l'EFI et dump pour l'ext4
echo "Démarrage de la création de l'image disque..."
dd if=$src_efi of=$dst_image bs=4096 conv=noerror,sync
dump -0uf $TRANSIT_D/dump-$(date +%d-%m-%Y) /

# Vérifier si la copie s'est terminée avec succès
if [ $? -eq 0 ]; then
    echo "Image disque créée avec succès dans $dst_image."
else
    echo "Erreur lors de la création de l'image disque."
    touch $error_log
fi

# Spécifique Freebox

chmod 600 $TRANSIT_D/*

# transfère en FTP
```

```
sync
sleep 2
echo "Transfert FTP de efi_image_$(date +%d-%m-%Y).img"
mv $dst_image $dst_dir
echo "Transfert FTP de dump-$(date +%d-%m-%Y)"
mv $TRANSIT_D/dump-$(date +%d-%m-%Y) $dst_dir
mv $error_log $dst_dir ;

# suppression des fichier de plus de 10 jours

echo "Suppression des ancien fichiers"
find $dst_dir -type f -mtime +10 -delete

# Démontage du dossier ftp

echo "Démontage du dossier ftp"
sync
fusermount -u $FTP_MOUNTPOINT

# Suppression du point de montage

sleep 2
rm -r $FTP_MOUNTPOINT
```

Bien sûr, il ne faut pas oublier de l'adapter et de créer une tâche CRON.

Revision #8

Created 28 January 2023 09:12:09 by Julien

Updated 8 February 2023 04:59:32 by Julien