

Script de synchronisation Rclone

```
#!/bin/bash

# DÉFINITION DES VARIABLES
# Fichier de log principal (toutes les opérations)
LOG_FILE="/var/log/rclone-gdrive.log"
# Fichier de log dédié aux erreurs
LOG_ERR="/var/log/rclone-gdrive-erreurs.log"
# Point de montage NFS vers le serveur TrueNAS
DEST="/mnt/<mount_point_name>"

# OPTIONS RCLONE
RCLONE_OPTS="--bwlimit 90M \           # Limite la bande passante à 90 Mo/s
             --progress \           # Affiche la progression en temps
             réel
             --update \           # Ne copie que les fichiers plus
             récents
             --checksum \         # Vérifie l'intégrité par somme de
             contrôle
             --transfers 10 \     # Nombre de transferts parallèles
             --checkers 10 \     # Nombre de vérifications parallèles
             --drive-skip-shortcuts \ # Ignore les raccourcis Google Drive
             --drive-acknowledge-abuse \ # Accepte le téléchargement des
             fichiers signalés
             --log-file=$LOG_FILE \ # Chemin du fichier de log
             --log-level DEBUG"     # Niveau de verbosité maximum

# TABLEAU DES SOURCES ET DESTINATIONS
# Format : "NOM_REMOTE_RCLONE: DOSSIER_DESTINATION"
# Chaque remote correspond à un Google Drive configuré dans rclone
DRIVES=(
    "<GDrive_1>: <dossier_destination_1>"
    "<GDrive_2>: <dossier_destination_2>"
    # Ajouter autant de remotes que nécessaire
)

# VÉRIFICATIONS DE SÉCURITÉ
# Vérification 1 : point de montage NFS actif ?
# Arrêt du script en cas de non-montage du NAS => Anti-saturation répertoire local
if ! mountpoint -q "$DEST"; then
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] ERREUR : point de montage absent ($DEST)" >> "$LOG_ERR"
```

```

    exit 1
fi
# Vérification 2 : espace disque disponible suffisant ?
# Si utilisation > 90 % alors arrêt de la synchronisation => Anti-saturation
stockage distant
DISK_USAGE=$(df "$DEST" | awk 'NR==2 {gsub(/%/,""); print $5}')
if [ "$DISK_USAGE" -gt 90 ]; then
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] ERREUR : disque presque plein
($DISK_USAGE%), arrêt du script" >> "$LOG_ERR"
    exit 1
fi

# BOUCLE DE SAUVEGARDE
# Pour chaque entrée du tableau, extraction de la source et de la
destination,
# puis exécution synchronisation rclone copy.
for ENTRY in "${DRIVES[@]}; do
    # Extraction de la source et de la destination
    SRC=$(echo "$ENTRY" | awk '{print $1}')
    DST=$(echo "$ENTRY" | awk '{print $2}')
    # Journalisation du début de la copie
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] Début de la copie : $SRC ->
$DEST/$DST" >> "$LOG_ERR"
    # Lancement copie avec options définies
    # Redirection des erreurs vers le fichier de log erreurs
    rclone copy "$SRC" "$DEST/$DST" $RCLONE_OPTS 2>> "$LOG_ERR"
done

```