Mes script

```
#!/bin/bash

# Dossier des enregistrements Guacamole
recordings_dir="/var/lib/guacamole/recordings"
# Dossier du NAS monté
nas_dir="/mnt/nas/guacamole_recordings"

# Étape 1 : Transfert et conversion des fichiers
transfer_and_convert() {
    local dir="$1"
    local file="$2"
    # Extraire le nom du fichier
    filename=$(basename "$file")

    # Étape 1.1 : Conversion du fichier avec guacenc en .m4v
    echo "Étape 1.1 : Conversion du fichier $filename en .m4v"
    convert_with_guacenc "$file"

    # Vérifier si la conversion a réussi
    if [ $? -eq 0 ]; then
        echo "Fichier $filename converti en .m4v avec succès
============================."

        # Étape 1.2 : Transfert du fichier converti sur le NAS
        echo "==================== Étape 1.2 : Transfert du fichier $filename sur le
NAS==================="
        rsync -av "$file.m4v" "$nas_dir/"

        # Vérifier si le fichier a été transféré correctement
        if [ $? -eq 0 ]; then
            echo "Fichier $filename transféré avec succès vers le
NAS.============================"

            # Étape 1.3 : Suppression du fichier local après conversion et transfert
            echo "Étape 1.3 : Suppression du fichier local $file et fichier converti"
            rm -f "$file" "$file.m4v"
            if [ $? -eq 0 ]; then
                echo "Fichier local $filename supprimé avec succès."
            else
                echo "Erreur lors de la suppression du fichier local $filename."
            fi

            # Étape 1.4 : Vérification et suppression du répertoire si vide
            echo "Étape 1.4 : Vérification et suppression du répertoire $dir si vide"
            if [ ! "$(ls -A "$dir")" ]; then
                rmdir "$dir"
```

```
            echo "Répertoire $dir supprimé car il est vide."
        else
            echo "Répertoire $dir non vide, il n'a pas été supprimé."
        fi
    else
        echo "Erreur lors du transfert du fichier $filename vers le NAS."
    fi
  else
    echo "Erreur lors de la conversion du fichier $filename."
  fi
}

# Étape 2 : Conversion avec guacenc pour créer un fichier .m4v
convert_with_guacenc() {
  local input_file="$1"
  # Appeler guacenc pour convertir en .m4v (résolution 1280x720)
  sudo guacenc -s 1280x720 -f "$input_file"

  # Vérifier si la conversion s'est bien déroulée
  if [ $? -eq 0 ]; then
    echo "Conversion réussie avec guacenc : $input_file -> $input_file.m4v"
    return 0
  else
    echo "Erreur lors de la conversion avec guacenc pour le fichier $input_file"
    return 1
  fi
}

# Étape 3 : Vérification du répertoire des enregistrements
echo "Étape 3 : Vérification du répertoire des enregistrements..."
if [ -d "$recordings_dir" ]; then
  # Parcourir tous les sous-répertoires dans /recordings
  for dir in "$recordings_dir"/*; do
    if [ -d "$dir" ]; then
      echo "Répertoire trouvé : $dir"

      # Parcourir les fichiers à l'intérieur de chaque sous-répertoire
      for file in "$dir"/*; do
        if [ -f "$file" ]; then
          echo "Fichier trouvé : $file"
          # Appeler la fonction pour transférer ce fichier vers le NAS et le convertir
          transfer_and_convert "$dir" "$file"
        fi
      done
    fi
  done
else
  echo "Le répertoire des enregistrements Guacamole n'existe pas."
```

```
fi

# Fin du script, sans notification par email
echo "Le script Guacamole a été exécuté avec succès."
=====================================================================

#!/bin/bash

USER_NAME="$1"
CONNECTION_ID="$2"
DATE_CONNEXION=$(date '+%Y-%m-%d %H:%M:%S')

SUBJECT="Connexion Guacamole - Utilisateur: $USER_NAME"
BODY="Salut,\n\nL'utilisateur $USER_NAME s'est connecté à Guacamole.\nDate et heure:
$DATE_CONNEXION\nConnexion ID: $CONNECTION_ID\n\nÀ plus !"

echo -e "Subject: $SUBJECT\n\n$BODY" | /usr/bin/msmtp --file=/root/.msmtprc -a default
stagiaire-it@daudruy.fr

=================================================================


zafar@apache-guaca:/opt/scripts$ cat monitor_guacamole_ssh.py
import smtplib
import time
import re
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Configuration SMTP
SMTP_SERVER = "smtp-mibc-fr-07.mailinblack.com"
SMTP_PORT = 25
MAIL_TO = "stagiaire-it@daudruy.fr"
MAIL_FROM = "stagiaire-it@daudruy.fr"  # À remplacer par un mail valide

# Fonction d'envoi d'alerte
def send_alert(subject, message):
    try:
        msg = MIMEMultipart()
        msg['From'] = MAIL_FROM
        msg['To'] = MAIL_TO
        msg['Subject'] = subject

        msg.attach(MIMEText(message, 'plain'))

        with smtplib.SMTP(SMTP_SERVER, SMTP_PORT) as server:
            server.sendmail(MAIL_FROM, MAIL_TO, msg.as_string())
        print("[+] Notification envoyée avec succès !")
```

```
    except Exception as e:
        print(f"[-] Erreur d'envoi de l'email: {e}")

# Surveillance des logs SSH et Guacamole
def monitor_logs():
    auth_log = "/var/log/auth.log"
    guac_log = "/var/log/tomcat9/catalina.out"

    with open(auth_log, "r") as ssh_log, open(guac_log, "r") as guac:
        ssh_log.seek(0, 2)
        guac.seek(0, 2)

        while True:
            ssh_line = ssh_log.readline()
            guac_line = guac.readline()

            # Détection des connexions SSH réussies
            if ssh_line and "Accepted password" in ssh_line:
                user = re.search(r'Accepted password for (\w+)', ssh_line)
                ip = re.search(r'from ([\d\.]+)', ssh_line)
                if user and ip:
                    msg = f"Connexion SSH détectée sur le serveur guacamole :\nUtilisateur :
{user.group(1)}\nIP : {ip.group(1)}"
                    send_alert("🔐 Alerte Connexion SSH sur le serveur Guacamole", msg)

            # Détection des échecs SSH
            if ssh_line and "Failed password" in ssh_line:
                ip = re.search(r'from ([\d\.]+)', ssh_line)
                if ip:
                    msg = f"Tentative de connexion SSH échouée depuis {ip.group(1)}"
                    send_alert("⚠️ Tentative SSH échouée", msg)

            # Détection des connexions Guacamole
            if guac_line and "User \" " in guac_line and "connected from" in guac_line:
                user = re.search(r'User \"(.*?)\"', guac_line)
                ip = re.search(r'from ([\d\.]+)', guac_line)
                if user and ip:
                    msg = f"Connexion Guacamole détectée en ssh  :\nUtilisateur :
{user.group(1)}\nIP : {ip.group(1)}"
                    send_alert("🖥️ Connexion Guacamole", msg)

            time.sleep(1)

if __name__ == "__main__":
    monitor_logs()
```

```
==============================================================
zafar@apache-guaca:/opt/scripts$ cat nas_supprime.sh
#!/bin/bash

# Répertoire cible
nas_dir="/mnt/nas/guacamole_recordings/"

# Vérifier si le répertoire existe
if [ -d "$nas_dir" ]; then
    echo "Suppression de tous les fichiers dans le répertoire $nas_dir"

    # Supprimer tous les fichiers dans le répertoire
    rm -rf "$nas_dir"/*

    echo "Tous les fichiers ont été supprimés avec succès."
else
    echo "Le répertoire $nas_dir n'existe pas."
fi

================================================
zafar@apache-guaca:/opt/scripts$ cat watch_guac_log.sh
#!/bin/bash
#
# Script qui surveille les logs de Guacamole et envoie une notification lors d'une connexion.

LOGFILE="/var/log/tomcat9/catalina.out"
KEYWORD="User .* connected to connection"

# Suivi en temps réel des logs
tail -n0 -F "$LOGFILE" | while read -r LINE; do
    # Vérifie si la ligne contient le mot-clé indiquant une connexion
    echo "$LINE" | grep -E "$KEYWORD" > /dev/null
    if [ $? -eq 0 ]; then
        # Extraction du nom de l'utilisateur et de la connexion
        USER_NAME=$(echo "$LINE" | awk -F"" '{print $2}')
        CONNECTION_ID=$(echo "$LINE" | awk -F'connected to connection "' '{print $2}' | awk
-F"" '{print $1}')
        DATE_CONNEXION=$(date '+%Y-%m-%d %H:%M:%S')

        # Log local (pour debug)
        echo "$(date '+%Y-%m-%d %H:%M:%S') - Connexion détectée : $USER_NAME sur
connexion $CONNECTION_ID" >> /tmp/guac_notify_watch.log

        # Appel du script de notification
        /opt/scripts/guac_notify.sh "$USER_NAME" "$CONNECTION_ID"
    fi
done
```