

créer et automatiser un script de sauvegarde bash

- [Phase préparatoire](#)
  - [Ligne de commande](#)
  - [Capture d'écran](#)
- [Script](#)
  - [ligne de commande](#)
  - [Contenu du Script](#)
  - [Capture d'écran](#)
- [Automatisation avec cron](#)
  - [ligne de commande](#)
  - [capture d'écran](#)

## Phase préparatoire

### Ligne de commande

```
#crée un dossier backup
mkdir backup
#crée des fichiers de test
touch Documents/fichier.txt
echo "Contenu du fichier 2" > documents/fichier2.log
```

### Capture d'écran

```
root@debian:~# touch Documents/fichier.txt
root@debian:~# echo "Contenu du fichier 2" > Documents/fichier2.log
root@debian:~#
```

## Script

### ligne de commande

```
#crée un fichier .sh avec le contenu du Script
nano sauvegarde.sh
#rendre exécutable le script
chmod +x sauvegarde.sh
#exécuter le script
./sauvegarde.sh
```

# Contenu du Script

```
#!/bin/bash

# =====
# Script de sauvegarde simple pour copier des fichiers
# Auteur : [Votre Nom]
# Date : 06/08/2025
# =====

# --- VARIABLES DE CONFIGURATION ---
# Utilisez toujours des chemins absolus pour éviter les erreurs, surtout avec
Cron.
SOURCE_DIRECTORY="/home/votre_user/documents"
BACKUP_DIRECTORY="/home/votre_user/backup"

# --- DÉBUT DU SCRIPT ---

echo "Démarrage du script de sauvegarde le $(date)"

# 1. Vérifier que le répertoire source existe
if [ ! -d "$SOURCE_DIRECTORY" ]; then
    echo "Erreur : Le répertoire source $SOURCE_DIRECTORY n'existe pas."
    exit 1 # Quitte le script avec un code d'erreur
fi

# 2. Si le répertoire de sauvegarde n'existe pas, le créer
if [ ! -d "$BACKUP_DIRECTORY" ]; then
    echo "Le répertoire de sauvegarde $BACKUP_DIRECTORY n'existe pas.
Création..."
    mkdir -p "$BACKUP_DIRECTORY"
fi

# 3. Copier les fichiers : deux méthodes

# Méthode A : Simple avec 'cp' (celle du script de base)
# cp -R "$SOURCE_DIRECTORY"/* "$BACKUP_DIRECTORY"

# Méthode B : Professionnelle avec 'rsync' (fortement recommandée !)
rsync -avh --delete "$SOURCE_DIRECTORY/" "$BACKUP_DIRECTORY"

# 4. Vérifier si la dernière commande (la copie) a réussi
if [ $? -eq 0 ]; then
    echo "Sauvegarde effectuée avec succès."
else
    echo "Erreur lors de la sauvegarde avec rsync."
```

```
    exit 1 # Quitte le script avec un code d'erreur
fi

echo "Fin du script de sauvegarde."
```

## Capture d'écran

```
GNU nano 7.2                               sauvegarde.sh *
#!/bin/bash

# =====
# Script de sauvegarde simple pour copier des fichiers
# Auteur : [Votre Nom]
# Date : 06/08/2025
# =====

# --- VARIABLES DE CONFIGURATION ---
# Utilisez toujours des chemins absolus pour éviter les erreurs, surtout avec Cron.
SOURCE_DIRECTORY="/home/votre_user/documents"
BACKUP_DIRECTORY="/home/votre_user/backup"

# --- DÉBUT DU SCRIPT ---

echo "Démarrage du script de sauvegarde le $(date)"

# 1. Vérifier que le répertoire source existe
if [ ! -d "$SOURCE_DIRECTORY" ]; then
    echo "Erreur : Le répertoire source $SOURCE_DIRECTORY n'existe pas."
    exit 1 # Quitte le script avec un code d'erreur
fi

# 2. Si le répertoire de sauvegarde n'existe pas, le créer
if [ ! -d "$BACKUP_DIRECTORY" ]; then
    echo "Le répertoire de sauvegarde $BACKUP_DIRECTORY n'existe pas. Création..."
    mkdir -p "$BACKUP_DIRECTORY"
fi

# 3. Copier les fichiers : deux méthodes

root@debian:~# ./sauvegarde.sh
Démarrage du script de sauvegarde le ven. 26 sept. 2025 17:09:10 CEST
sending incremental file list
./
fichier.txt
fichier1.txt
fichier2.log

sent 286 bytes  received 76 bytes  724,00 bytes/sec
total size is 35  speedup is 0,10
Sauvegarde effectuée avec succès.
Fin du script de sauvegarde.
```

# Automatisation avec cron

## ligne de commande

```
#ouvrir crontab et choisir celui conseiller (ici c'est le 1)  
crontab -e  
#mod le fichier voir la capture d'écran
```

## capture d'écran

```
root@debian:~# crontab -e  
no crontab for root - using an empty one  
  
Select an editor. To change later, run 'select-editor'.  
  1. /bin/nano          <---- easiest  
  2. /usr/bin/vim.tiny  
  
Choose 1-2 [1]: █
```

```
Terminal -
Fichier  Édition  Affichage  Terminal  Onglets  Aide
GNU nano 7.2 /tmp/crontab.xS0ZY8/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
00 23 * * * /root/sauvegarde.sh >> /root/backup.log 2>&1

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^V Remplacer  ^U Coller    ^J Justifier  ^_ Aller ligne
```

quitter et le cron est actif