

# Automatisez vos actions récurrentes avec des scripts Bash

## Exemple de corrigé pour l'activité "À vous de jouer"

### 1. Création du script de sauvegarde

None

```
nano backup.sh
```

Ajout du contenu

None

```
#!/bin/bash

FICHIER_LOG="historique_sauvegardes.log"

DATE=$(date +%Y-%m-%d_%H-%M)

SOURCE="./dossier_a_sauvegarder" # Répertoire à sauvegarder

ARCHIVE="sauvegarde_$(DATE).tar.gz"

tar -czf "$ARCHIVE" "$SOURCE" >> "$FICHIER_LOG" 2>> "$FICHIER_LOG"

if [ $? -eq 0 ]; then

echo "[SUCCÈS] L'archive $ARCHIVE a été créée."

else

echo "[ERREUR] La sauvegarde a échoué. Consultez $FICHIER_LOG."
```

```
fi
```

→ Le script définit dans des variables le répertoire à sauvegarder, le nom de l'archive et le fichier `log`, puis lance la sauvegarde compressée. Ces valeurs peuvent être adaptées facilement selon le contexte.

## 2. Compression du répertoire avec `tar`

```
None
```

```
tar -czf "$ARCHIVE" "$SOURCE"
```

Résultat attendu (exemple)

Une archive compressée est créée dans le dossier courant, avec un nom du type :

```
None
```

```
sauvegarde_2026-04-13_14-30.tar.gz
```

→ La commande `tar` permet ici de regrouper et compresser le répertoire fourni en une seule archive.

## 3. Redirection vers un fichier de log

```
None
```

```
>> "$FICHIER_LOG" 2>> "$FICHIER_LOG"
```

Résultat attendu (exemple)

Un fichier `historique_sauvegardes.log` est créé ou complété dans le dossier courant.

→ Les redirections permettent de conserver une trace de l'exécution du script dans un fichier log, y compris en cas d'erreur.

## 4. Vérification du résultat avec le code de retour

```
None
```

```
if [ $? -eq 0 ]; then
```

```
    echo "[SUCCÈS] L'archive $ARCHIVE a été créée."
else
    echo "[ERREUR] La sauvegarde a échoué. Consultez
    $FICHIER_LOG."
fi
```

Résultat attendu (exemple)

None

```
[SUCCÈS] L'archive sauvegarde_2026-04-13_14-30.tar.gz a
été créée.
```

ou

None

```
[ERREUR] La sauvegarde a échoué. Consultez
historique_sauvegardes.log.
```

→ Le script vérifie si la commande de sauvegarde s'est bien déroulée, puis affiche un message clair à l'utilisateur. Cette logique suit directement l'exemple du cours avec `$?` et `if`.

## 5. Exécution du script

None

```
chmod +x backup.sh
./backup.sh
```

Résultat attendu (exemple)

None

```
[SUCCÈS] L'archive sauvegarde_2026-04-13_14-30.tar.gz a été
créée.
```

Puis, dans le dossier de travail, on retrouve :

- l'archive compressée,

- le fichier `historique_sauvegardes.log`.

→ Le script produit une sauvegarde réutilisable, laisse une trace de son exécution et informe clairement l'utilisateur du résultat obtenu.