

Antoine DECABOOTER HOLLEMAN

Guillaume SANCHEZ

TP Sauvegarde à distance

1^{ère} étape crée 3 dossiers nommées respectivement « rep_dest, rep_src, sauvegarde) dans le dossier « ~ ».

```
user@debian:~$ ls
Desktop  Documents  Downloads  github  linuxtoolbox  Music  Pictures  programmation  Public  rep_dest  rep_src  sauvegarde  Templates  Videos
```

Aller dans ~/Documents et créer le script « save.sh »

Script save

```
#!/bin/bash
#script de sauvegarde

#dossier à sauvegarder, à modifier selon l'utilisateur
DATA_TO_BACKUP="/home/user/rep_src/"

#dossier local de la création de la sauvegarde, à modifier selon l'utilisateur
DATA_DIR="/home/user/sauvegarde/"

#utilisateur du serveur destinataire de la sauvegarde
USER_DEST="user"

#server destinataire de la sauvegarde
ADDRESS_DEST="172.20.10.5" #adresse serveur distant

#dossier du serveur destinataire de la sauvegarde
BACKUP_DIR_DEST="/home/user/rep_dest/"

SEPARATOR="-----"

#definition de la date du jour afin de pouvoir récupérer (à adapter)
#l'année (3$), le mois (1$) et le jour (2$) pour nommer le fichier de sauvegarde
IFS=/
set $(date +%D)
annee=$3
mois=$1
jour=$2

unset IFS

#debut du script de sauvegarde de /home
echo $SEPARATOR
echo "debut du script de sauvegarde de /home"
```

```

#sauvegarde des données
echo $SEPARATOR
echo "sauvegarde de $DATA_TO_BACKUP dans $DATA_DIR/$BACKUP_FILE"
tar cvjf $BACKUP_DIR/svg._$annee-$mois-$jour.tar.bz2 --newer-mtime "1 day ago"
$DATA_TO_BACKUP

#envoi vers le serveur de sauvegarde
echo $SEPARATOR
echo "envoi de svg._$annee-$mois-$jour.tar.bz2 vers le serveur
$USER_DEST@$ADDRESS_DEST:$BACKUP_DIR_DEST"
scp $BACKUP_DIR/svg._$annee-$mois-$jour.tar.bz2 $USER_DEST@$ADDRESS_DEST:$BACKUP_DIR_DEST

#verification de la bonne réception
echo $SEPARATOR
echo "ls -l de $USER_DEST@$ADDRESS_DEST:$BACKUP_DIR_DEST pour confirmer la bonne reception"
ssh $USER_DEST@$ADDRESS_DEST ls -l $BACKUP_DIR_DEST

#nettoyage de la sauvegarde locale
echo $SEPARATOR
echo "nettoyage de la sauvegarde locale presente dans $BACKUP_DIR"
rm -f $BACKUP_DIR/svg._$annee-$mois-$jour.tar.bz2

#fin du script de sauvegarde de /home
echo $SEPARATOR
echo "fin du script de sauvegarde de /home"
echo $SEPARATOR

```

Ce script permet d'effectuer une sauvegarde locale des fichiers modifiés au cours des 24 dernières heures.

Étapes principales :

1. Définition des variables :

- DATA_TO_BACKUP : Chemin du répertoire source à sauvegarder.
- BACKUP_DIR : Chemin du répertoire où la sauvegarde sera stockée.
- IFS et date : Permettent de formater la date actuelle pour nommer l'archive.

2. Création de l'archive :

- Utilisation de la commande **tar** avec l'option **--newer-mtime "1 day ago"** pour ne sauvegarder que les fichiers modifiés dans les dernières 24 heures.
- Compression avec **bzip2** pour réduire la taille de l'archive.

3. Envoi vers un serveur distant (facultatif) :

- Commande **scp** utilisée pour transférer l'archive au serveur spécifié.

4. Nettoyage local :

- Suppression de l'archive après transfert pour éviter l'encombrement.

Commande clé :

```
tar cvjf $BACKUP_DIR/svg._$annee-$mois-$jour.tar.bz2 --newer-mtime "1 day ago"
$DATA_TO_BACKUP
```

Script restore

```
#!/bin/bash
#script de restauration

#utilisateur du serveur destinataire de la sauvegarde
USER_DEST="user"

#server destinataire de la sauvegarde
ADDRESS_DEST="172.20.10.5" #adresse serveur distant

#dossier du serveur destinataire de la sauvegarde
BACKUP_DIR_DEST="/home/user/rep_dest/"

#dossier local de la réception de la sauvegarde, à créer s'il n'existe pas, à
modifier selon l'utilisateur
BACKUP_DIR="/home/user/sauvegarde/"

SEPARATOR="-----"

#debut du script de restauration de /home
echo $SEPARATOR
echo "debut du script de restauration de /home"
echo $SEPARATOR

#tableau de la liste des fichiers de sauvegardes présentes sur le serveur dans
/home/user/rep_dest/
fileslist=$(ssh $USER_DEST@$ADDRESS_DEST ls $BACKUP_DIR_DEST))

echo "voici la liste des sauvegardes disponibles sur $ADDRESS_DEST dans le
repertoire $BACKUP_DIR_DEST"
echo

#boucle for afin d'afficher les fichiers avec leur position dans le tableau devant
for (( i=0; i<${#fileslist[@]}; i++ ))
do
    echo "$[i] ${fileslist[$i]}" #premier fichier
done

echo
echo $SEPARATOR

#un read pour que l'utilisateur puissent entrer le numéro de la sauvegarde qu'il
veut
read -r -p "choisir l'archive a restaurer : " numSauv
echo $SEPARATOR
```

```

#place le nom de la sauvegarde dans une variable OKZOU
NAME_SAVE=${fileslist[$numSauv]}
echo "vous avez choisi : $NAME_SAVE"
echo $SEPARATOR

#je rentre dans une boucle "infini" mais contrôlé par une action utilisateur
while true; do

    #un read pour que l'utilisateur puisse confirmer si oui ou non il veut
    restaurer
    read -r -p "lancer la restauration ? (o/n) : " response
    case $response in
        [oO]|[yY]|[oui]|[yes])
            #récupération avec le protocole ssh (commande scp) de la sauvegarde
            présente sur le server, vers notre pc
            echo "récupération de $NAME_SAVE dans $BACKUP_DIR"
            scp $USER_DEST@$ADDRESS_DEST:$BACKUP_DIR_DEST$NAME_SAVE $BACKUP_DIR
            echo $SEPARATOR

            #extraction de l'archive à la racine (donc /) pour respecter la
            structure des répertoires d'un chemin
            echo "extraction de la sauvegarde"
            tar xvjf $BACKUP_DIR$NAME_SAVE -C /
            echo $SEPARATOR

            #petit nettoyage de la sauvegarde présente sur notre pc
            echo "nettoyage de $NAME_SAVE"
            rm $BACKUP_DIR$NAME_SAVE

            #fin du script de sauvegarde de /home
            echo $SEPARATOR
            echo "fin du script de restauration de /home"
            echo $SEPARATOR
            exit 0
            ;;
        [nN]|[non]|[no])
            #fin du script de sauvegarde de /home
            echo $SEPARATOR
            echo "fin du script de restauration de /home"
            echo $SEPARATOR
            exit 0
            ;;
        *)
            echo "erreur, merci d'entrer o ou n"
            ;;
    esac
done

```

Ce script facilite la restauration d'une sauvegarde en proposant une liste interactive des archives disponibles.

Étapes principales :

1. Connexion au serveur distant :

- Utilisation de **ssh** pour lister les archives disponibles dans le répertoire distant.

2. Liste interactive :

- Affichage des fichiers disponibles avec leur index pour une sélection facile.

3. Récupération et extraction :

- Téléchargement de l'archive sélectionnée avec **scp**.
- Extraction dans la structure originale à l'aide de **tar**.

4. Confirmation utilisateur :

- Le script demande une confirmation avant de lancer la restauration.

Commande clé :

tar xvjf \$BACKUP_DIR\$NAME_SAVE -C /

Points Importants

- Variables à modifier :

- Les chemins pour DATA_TO_BACKUP, BACKUP_DIR, et les informations du serveur distant (adresse IP, utilisateur).

- Sécurité :

- Assurez-vous que les permissions des répertoires sont correctement configurées pour éviter des accès non autorisés.

- Logs :

- Ajouter des fichiers journaux (log) pour tracer les opérations peut être une amélioration.

Scénarios d'Utilisation

1. Création d'une sauvegarde :

- Modifier le script **save.sh** avec les bons chemins et exécuter.
- Exemple :
./save.sh

2. Restauration d'une sauvegarde :

- Exécuter le script **restore.sh** et choisir l'archive.
- Exemple :
./restore.sh

Améliorations Futures

- Ajouter une vérification de l'intégrité des sauvegardes avec **sha256sum**.
- Gérer des sauvegardes incrémentales ou différentielles.
- Permettre un déploiement multi-utilisateur avec gestion des permissions.

Démonstration :

Pour commencer, nous créons un fichier “test” dans “~/rep_src” :

```
-(jeu. janv. 23 15:22:33)--(pop-os:~/rep_src)-  
[nk] $ echo "Bonjour, nous sommes le 23.01.25" >> test  
-(jeu. janv. 23 15:23:04)--(pop-os:~/rep_src)-  
[nk] $ cat test  
Bonjour, nous sommes le 23.01.25  
-(jeu. janv. 23 15:23:09)--(pop-os:~/rep_src)-  
[nk] $ ls -l  
total 4  
-rw-rw-r-- 1 nk nk 33 janv. 23 15:23 test
```

Ensuite, nous lançons le script “save.sh” pour sauvegarder le fichier “test”, il n’y a pas de demande de mot de passe car nous avons échangé nos clés publiques :

```
-(jeu. janv. 23 15:20:30)--(pop-os:~/Documents/UTEC-Lic_ME_2024_2025/UTC502/UTC502-26-11-24/TU)-  
[nk] $ ./save.sh  
-----  
Debut du script de Sauvegarde de /home  
-----  
Sauvegarde de /home/nk/rep_src/ dans /home/nk/sauvegarde/  
tar: Option --newer-mtime : date « 1 day ago » traitée comme 2025-01-22 15:24:02.15034195  
tar: Suppression de « / » au début des noms des membres  
/home/nk/rep_src/  
/home/nk/rep_src/test  
-----  
Envoie de svg._250123.tar.bz2 vers le serveur user@192.168.1.181:/home/user/rep_dest/  
svg._250123.tar.bz2 100% 198 329.3KB/s 00:00  
-----  
ls -l de user@192.168.1.181:/home/user/rep_dest/ pour confirmer bonne reception :  
total 216148  
-rw-r--r-- 1 user user 609435 Nov 27 11:45 svg._241125.tar.bz2  
-rw-r--r-- 1 user user 609435 Nov 27 11:45 svg._241126.tar.bz2  
-rw-r--r-- 1 user user 609435 Nov 27 11:37 svg._241127.tar.bz2  
-rw-r--r-- 1 user user 219493378 Dec 16 18:26 svg._241216.tar.bz2  
-rw-r--r-- 1 user user 204 Jan 6 11:46 svg._250106.tar.bz2  
-rw-r--r-- 1 user user 198 Jan 23 15:24 svg._250123.tar.bz2  
-----  
Nettoyage de la sauvegarde local présente dans /home/nk/sauvegarde/  
Fin du script de Sauvegarde de /home  
-----
```

Nous supprimons le fichier test :

```
-(jeu. janv. 23 15:23:49)--(pop-os:~/rep_src)-  
[nk] $ rm test  
-(jeu. janv. 23 15:24:45)--(pop-os:~/rep_src)-  
[nk] $ ls -l  
total 0
```

Nous lançons le script “restore.sh” pour restaurer le fichier “test”. Nous choisissons l’archive 5 qui est à la date de la sauvegarde :

```
-(jeu. janv. 23 15:25:08)--(pop-os:~/Documents/UTEC-Lic_ME_2024_2025/UTC502/UTC502-26-11-24/Td)-  
[nk] $ ./restore.sh  
-----  
Debut du script de restauration de /home  
-----  
Voici la liste des sauvegardes disponible sur 192.168.1.181 dans le repertoire /home/user/rep_dest/ :  
0-svg._241125.tar.bz2  
1-svg._241126.tar.bz2  
2-svg._241127.tar.bz2  
3-svg._241216.tar.bz2  
4-svg._250106.tar.bz2  
5-svg._250123.tar.bz2  
-----  
Choisir l'archive à restaurer : 5  
-----  
Vous avez choisi : svg._250123.tar.bz2  
-----  
Lancer la restauration ? (o/n) o  
Reception de svg._250123.tar.bz2 dans /home/nk/sauvegarde/  
svg._250123.tar.bz2 100% 198 322.0KB/s 00:00  
-----  
Extraction de la sauvegarde  
home/nk/rep_src/  
home/nk/rep_src/test  
-----  
Nettoyage de svg._250123.tar.bz2  
-----  
Fin du script de Restauration de /home  
-----
```

Enfin, nous constatons que le fichier test a été restauré avec succès.

```
-(jeu. janv. 23 15:25:40)--(pop-os:~/rep_src)-  
[nk] $ ls -l  
total 4  
-rw-rw-r-- 1 nk nk 33 janv. 23 15:23 test  
-(jeu. janv. 23 15:25:42)--(pop-os:~/rep_src)-  
[nk] $ cat test  
Bonjour, nous sommes le 23.01.25
```