Sauvegarde légère et rapide du système

- Objet : du tuto Sauvegarde légère et rapide du système
- Niveau requis :
  avisé
- Suivi :
  en-chantier
- Commentaires : *Tout est dans le titre.*
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.
- Création :philou92 le 4/07/17
- Mise à jour : le 8/12/17
- Testé par :philou92 le 4/07/17
- Commentaires sur le forum : https://debian-facile.org/viewtopic.php?id=18405

Introduction

Ce tuto donne une méthode pour sauvegarder son système d'exploitation debian et le répertoire /home

Motivation principale de ce tuto :

Hormis la sauvegarde du répertoire /home sur lequel on trouve pléthore d'excellentes solutions sur le net, ce tuto essaye d'apporter une façon de sauvegarder le système de façon légère et rapide. Légère en taille d'enregistrement en ne gardant que les fichiers utiles et rapide en ne sauvegardant que ce qui a changé entre deux sauvegardes.

Le système est sauvegardé alternativement dans deux conteneurs BANK1 et BANK2. dans le cas ou une sauvegarde serait corrompue la deuxième sert de roue de secours.

Pour le répertoire /home j'ai opté pour une sauvegarde incrémentale qui garde l'historique des fichiers modifiés ou supprimés. En cas de besoin il sera possible de revenir chercher des fichiers perdus à la demande des utilisateurs du PC.

Par défaut les sauvegardes sont programmées à intervalle minimale de 7 jours. Les utilisateurs du groupe “sauvegarde” pourront à tous moment déclencher ponctuellement une demande de sauvegarde.

Les sauvegardes ont lieu à l'arrêt du PC.

Bien sûr une sauvegarde sans méthode de restauration n'a aucun intérêt. C'est l'objet du deuxième tuto : lien https://debian-facile.org/utilisateurs:philou92:tutos:restauration-systeme?&#introduction.

Prérequis systeme : Votre systeme d'initialisation est **systemd**.

Prérequis Matériel : Vous aurez besoin d'un support de sauvegarde dédié. Personnellement j'ai fait le choix d'un disque dur USB du commerce.

Le Disque dur est formaté dans un format équivalent à votre système (exemple ext4)

du moins qui conserve les attribut des fichiers Linux.

Si besoin voir la commande mkfs https://debian-facile.org/doc:systeme:mkfs?s[]=ext4.

Dans ce que va suivre je pars du principe que c'est ce type de support qui est utilisé.

Maintenant en avant toute...

Installation

l'outil debconf-utils

Il permet de récupérer les configurations des paquets du système. Bien que je n'ai pas eu à m'en servir, ce serait bête de ne pas les avoirs sous la main le jour ou votre barbu local vous dira c'est c.. de ne pas les avoirs sauvegarder...

En matière de sauvegarde le mieux est l'ami du bien.

Pour l'installer

```
apt-get update && apt-get install debconf-utils
```

l'outil de chiffrage “encfs”

Afin de ne pas laisser les données personnelles à la merci des Rapetout en tout genre j'ai ajouté une petite fantaisie en utilisant l'outil “encfs” pour chiffrer mon /home.

Pour l'installer

```
apt-get update && apt-get install encfs
```

Mise en place du système de sauvegarde

Créer le répertoire sauvegarde sur /mnt qui servira de point de montage :

```
cd /mnt
mkdir sauvegarde
```

Préparation du disque dur

Brancher votre disque dur externe.

Récupérer l'UUID du disque de sauvegarde :

```
blkid
```

Repérez votre disque comme ici (Note : Sauvegarde c'est le petit nom que j'ai donné à mon disque)

retour de la commande

```
LABEL="Sauvegarde" UUID="8498c62b-ad6f-4966-990e-822f729a01c3"
TYPE="ext4" PARTUUID="f239042f-01"
```

Éditez votre fichier /etc/fstab avec votre éditeur favori (nano, vim) et ajoutez la ligne suivante

extrait/etc/fstab

```
#UUID=8498c62b-ad6f-4966-990e-822f729a01c3 /mnt/sauvegarde  ext4
defaults,nofail,noauto    0  2
```

La ligne est pour l'instant commentée (On enlèvera le “#” à la fin dès que l'on sera assuré que tout est OK).

Vérifier à l'aide de la commande suivante si votre disque est déjà monté. Taper la commande suivante :

```
mount
```

Si vous trouvez une ligne du genre ci dessous, démontez le disque dur (sans le déconnecter du port USB) (voir https://debian-facile.org/doc:systeme:umount) :

retour de la commande

```
/dev/sdb1 on /media/<USER>/Sauvegarde
```

Montage du disque dur dans le répertoire /mnt/sauvegarde :

```
fdisk -l
```

repérer le répertoire en /dev/sdbx de votre disque dur externe.

Supposons qu'il s'appelle /dev/sdb1. Monter le disque avec

```
mount /dev/sdb1 /mnt/sauvegarde
```

Placez-vous à la racine du disque :

```
cd /mnt/sauvegarde
```

Créer les répertoires suivants :

```
mkdir BANK1
mkdir BANK2
```

Créer le fichier suivant :

```
touch BANK.status
```

Editez-le et ajouter dedans le texte :

nom

```
BANK2
```

Créer les trois fichiers suivants :

```
touch restaurestart
chmod 744 restaurestart
touch restaurepaquets
chmod 744 restaurepaquets
touch restauredebian
chmod 744 restauredebian
```

Editez les et y ajouter les scripts qui sont là : https://debian-facile.org/utilisateurs:philou92:tutos:sauvegarde-systeme-leger?&#scripts-de-restauration-systeme

Taper la commande :

```
encfs /mnt/sauvegarde/.backuphome /mnt/sauvegarde/backuphome
```

Créer les répertoires automatiquement et appliquer le mode normal (mode qui conserve les liens durs)

Répondre **y** puis **y** puis **entrée**

A la demande entrer votre “Monmotdepassesupersecret”

Ajoutez le fichier .superpointeur.txt et créer le répertoire “Actuel” dans backuphome :

```
cd backuphome
touch .superpointeur.txt
mkdir Actuel
```

Démonter le répertoire chiffré :

```
cd /mnt/sauvegarde
```

```
fusermount -u /mnt/sauvegarde/backuphome
```

Création du service de sauvegarde

Créer le fichier suivant : sauvegarde.service dans le répertoire /etc/systemd/system/

Précisions sur TimeoutStopSec.

Dès l'arrêt du système si une sauvegarde est déclenchée, elle doit se terminer avant le temps fixé par “TimeoutStopSec”, temps au delà duquel le système mettra fin de façon rédhibitoire au script de sauvegarde. (un compteur de temps s'affiche à ce moment).

Par défaut ce paramètre est fixé à 90s soit 1mn30. Personnellement Mes sauvegardes n'ont jamais dépassées 1mn. Néanmoins par précaution j'ai fixé le temps à 360s (6mn). A vous de voir s'il faut régler plus de temps selon vos usages et délais entre deux sauvegardes.

Le délai TimeoutStopSec est un délai maximum d'attente (un garde fou). Dès que le script de sauvegarde a terminé sont travail le système embraye immédiatement l'arrêt complet du système.

Vous pourriez également fixer ce paramètre à “Infinity”, cela revient à inhiber le garde fou). Là pas de risque d'arrêt du script de sauvegarde, mais en contrepartie, si le script vient à planter alors le système ne pourra pas s'arrêter correctement. Je déconseille cette dernière option.

sauvegarde.service

```
[Unit]
Description=Sauvegarde systeme et home
Before=shutdown.target reboot.target halt.target

[Service]
Type=oneshot
TimeoutStopSec=360
ExecStart=/bin/true
ExecStop=/root/backup/rsync-backup.sh
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

Création d'un groupe utilisateurs "sauvegarde"

Ce groupe va permettre à des utilisateurs de confiance (vous à priori) à déclencher une sauvegarde ponctuelle.

Créer le groupe sauvegarde

```
addgroup sauvegarde
```

retour de la commande

```
Ajout du groupe « sauvegarde » (GID 1003)...
Fait.
```

Ajouter un ou plusieurs <USER> habilité(s) à déclencher une sauvegarde ponctuelle au groupe “sauvegarde”.

```
adduser <USER1> sauvegarde
```

retour de la commande

```
Ajout de l'utilisateur « <USER1> » au groupe « sauvegarde »...
Ajout de l'utilisateur <USER1> au groupe sauvegarde
Fait.
```

Créer le répertoire /usr/local/sauvegarde/

```
cd /usr/local
mkdir sauvegarde
cd sauvegarde
```

A l'intérieur du répertoire faire un

```
touch rsync-backup-user.date
touch rsync-backup-user.sh
```

Ajouter le script suivant au fichier rsync-backup-user.sh avec votre éditeur préféré :

rsync-backup-user.sh

```
#! /bin/sh
touch /usr/local/sauvegarde/rsync-backup-user.date
exit
```

Ajouter les deux fichier au groupe sauvegarde

```
chgrp sauvegarde rsync-backup-user.date
```

```
chgrp sauvegarde rsync-backup-user.sh
```

Modifiez les droits des deux fichiers comme suit:

```
chmod 750 rsync-backup-user.sh
chmod 660 rsync-backup-user.date
```

Vérifiez les droits :

```
ls -al
```

retour de la commande

```
drwxr-sr-x  2 root staff      4096 juin  29 22:05 .
drwxrwsr-x 11 root staff      4096 juin  29 22:03 ..
-rw-rw----  1 root sauvegarde    0 juin  29 22:03 rsync-backup-
user.date
-rwxr-x---  1 root sauvegarde   68 juin  29 22:05 rsync-backup-user.sh
```

Le script de sauvegarde

Créer le répertoire /root/backup/

```
cd /root
mkdir backup
cd backup
```

A l'intérieur créer les trois fichiers suivants :

```
touch rsync-backup.log
touch rsync-backup.date
touch rsync-backup.sh
```

Comme son extension le souligne, le fichier rsync-backup.log enregistre les événements d'erreur ou de réussite de la sauvegarde.

Editer le fichier rsync-backup.sh et ajouter le script suivant:

Les lignes encadrées en haut et en bas par les caractères “#A-CORRIGER” sont à modifier :

- Modifier la ligne 26 avec l'UUID de votre disque.

- Modifier <USER> à la ligne 102 en le remplaçant par le nom du répertoire utilisateur dans lequel vous avez stocké le fichier “.clefsauvegarde”.

rsync-backup.sh

```
#! /bin/sh
SOURCE="/"
MONTAGE="/mnt/sauvegarde"
DESTINATION1="/mnt/sauvegarde/BANK1"
DESTINATION2="/mnt/sauvegarde/BANK2"
BANKSTATUS="/mnt/sauvegarde/BANK.status"
JOURNAL="/root/backup/rsync-backup.log"
LASTDATE="/root/backup/rsync-backup.date"
NBJOUR=$(($(date -u +%s) - $(date -ur $LASTDATE  +%s))/86400))
APPELUSER="/usr/local/sauvegarde/rsync-backup-user.date"
NBJOURUSER=$(($(date -u +%s) - $(date -ur $APPELUSER +%s))/86400))

# test si la date d'échéance est arrivée
if [ $NBJOUR -lt 7 ];
then
    echo "`date`    Pas le moment" >> $JOURNAL
    if [ $NBJOURUSER -lt 7 ];
    then
        touch -t 201701010000 $APPELUSER
    else
        exit
    fi
fi
# Essaye de monter le disque dur et recherche dans quel conteneur
enregistrer la sauvegarde
#A-CORRIGER
mount UUID=8498c62b-ad6f-4966-990e-822f729a01c3 $MONTAGE # A modifier
avec l'UUID de votre de disque de sauvegarde
#A-CORRIGER
if [ -f $BANKSTATUS ]; then
    echo "`date`    Montage disque OK" >> $JOURNAL
    VALSAUVEGARDE=`cat $BANKSTATUS`
    if [ "$VALSAUVEGARDE" = "BANK1"  ]; then
        DESTINATION=$DESTINATION2
        VALSAUVEGARDE="BANK2"
    elif [ "$VALSAUVEGARDE" = "BANK2"  ]; then
        DESTINATION=$DESTINATION1
        VALSAUVEGARDE="BANK1"
    else
        echo "`date`    Echec sauvegarde SYSTEME:    Contenu
$BANKSTATUS non conforme" >> $JOURNAL
        umount $MONTAGE
        exit
    fi
else
    echo "`date`    Echec sauvegarde SYSTEME:    fichier $BANKSTATUS
introuvable" >> $JOURNAL
```

```
    exit
fi

# Sauvegarde du système
if [ -d $DESTINATION ] ;
then
    rsync_backup_include='/root/backup/rsync-backup-include'
    dpkg --get-selections "*" > $DESTINATION/dpkg-selections.list
    debconf-get-selections > $DESTINATION/debconf-selections
    /usr/bin/rsync -a --delete-after $SOURCE $DESTINATION --exclude-
from=$rsync_backup_include --exclude='**'
    touch $LASTDATE
    echo "`date`    Sauvegarde SYTEME OK:    sur $DESTINATION" >>
$JOURNAL
    echo $VALSAUVEGARDE > $BANKSTATUS
# umount $MONTAGE
else
    echo "`date`    Echec sauvegarde SYSTEME:    répertoire
$DESTINATION introuvable" >> $JOURNAL
    umount $MONTAGE
    exit
fi

# Variables d'environnement
SOURCE="/home"
DESTINATION="/mnt/sauvegarde"
ACTUEL="/mnt/sauvegarde/backuphome/Actuel"
BACKUP="/mnt/sauvegarde/backuphome/$(date +'%d-%m-%Y')"
T="$DESTINATION/backuphome/.superpointeur.txt"
LISTEEXCLUSIONS="/root/backup/rsync-backup-include2"

# Codes d'erreur
E_REPABSENT=65
E_POINTEURABSENT=66

# test 1
if [ -d $DESTINATION/backuphome ] && [ -d $DESTINATION/.backuphome ]  ;
    then # les répertoires existent
        echo "répertoires trouvés"

    else

        echo "`date`    Echec sauvegarde HOME:         répertoires de
sauvegarde introuvables." >> $JOURNAL
        # Vérifier si le support de sauvegarde est monté
        umount $MONTAGE
        exit $E_REPABSENT
fi # fin test 1

# Test 2
if [ -e "$T" ] ;
```

```
    then # le répertoire d'archivage sécurisé est déjà monté
        echo "`date`    Echec sauvegarde HOME    Le dossier sécurisé
est déjà monté." >> $JOURNAL
        # Démonter le répertoire sécurisé avant l'utilisation du
script."
        umount $MONTAGE
        exit $E_POINTEURABSENT

    else # Il faut monter le répertoire d'archivage sécurisé
        chmod 750 $DESTINATION/backuphome
        #Déchiffrage du répertoire
                #Remplacer <USER> par l'utilisateur de confiance où est
stocké le mot de passe
                #A-CORRIGER
        cat /home/<USER>/Documents/.clefsauvegarde | encfs --stdinpass
-v "$DESTINATION/.backuphome" "$DESTINATION/backuphome"
                #A-CORRIGER
        # Coffre ouvert la sauvegarde sécurisée va démarrer!
        # crée le répertoire Backup du jour
        mkdir -p "$BACKUP"
        # crée la sauvegarde
        /usr/bin/rsync -a --delete --exclude-from $LISTEEXCLUSIONS --
backup --backup-dir="$BACKUP" "$SOURCE" "$ACTUEL" --exclude='**'
        # supprime le répertoire backup s'il est vide
        rmdir --ignore-fail-on-non-empty "$BACKUP"
        # démonte le répertoire en clair
        fusermount -u "$DESTINATION/backuphome"
        chmod 440 $DESTINATION/backuphome
        echo "`date`    Sauvegarde HOME OK" >> $JOURNAL
        umount $MONTAGE
fi # fin test 2

exit 0
```

Modifier les droits du fichier rsync-backup.sh

```
chmod 700 rsync-backup.sh
```

Fichiers de configuration rsync

faire un

```
touch rsync-backup-include
```

et ajouter les lignes suivantes :

rsync-backup-include

```
# La commande rsync utilisée inclue / (root) et exclue tout. Cela ne
sauvegarde rien. Le fichier rsync-backup-include permet d'inclure ce
que l'on veur sauvegarder.
# (note : pour plus d'information voir la page de manuel de rsync dans
les sections : FILTER RULES, INCLUDE/EXCLUDE PATTTERN RULES, MERGE-FILE
FILTviER RULES)
# à la visite d'un dossier, penser à exclure ce que l'on ne veut pas
sauvegarder avant de mettre la règle contenant les *
# + [pattern] inclu un motif
# - [pattern] exclu un motif
# '*' remplace tout et s'arrête au premier slash.
# '**' remplace tout, incluant les slashs.
# + /dossier/ visite un dossier mais ne sauvegarde rien
# + /dossier/ sauvegarde tous les fichiers et dossiers mais de manière
non récursive (les dossiers seront vides)
# + /dossier/** sauvegarde tous le contenu du dossier récursivement
# exclure les fichiers
- /initrd.img
- /vmlinuz

# sauvegarder le contenu du dossier /etc
+ /etc/
+ /etc/**

# sauvegarder le contenu du dossier /var
+ /var/
+ /var/spool/
+ /vat/spool/**
+ /var/backups/
+ /var/backups/**
+ /var/lib/
+ /var/lib/**
+ /var/games/
+ /var/games/**
+ /var/lock/
+ /var/lock/**
+ /var/log/
+ /var/log/**
+ /var/mail/
+ /var/mail/**
+ /var/opt/
+ /var/opt/**
+ /var/cache/
+ /var/cache/apt/
+ /var/cache/apt/**

# sauvegarder le contenu du dossier /usr
+ /usr/
+ /usr/local/
```

```
+ /usr/local/**

# sauvegarder le contenu de /root
+ /root/
+ /root/**

# exclure tout (comme ça on est sûr)
- **
```

faire un :

```
touch rsync-backup-include2
```

et ajouter les lignes suivantes (à adapter selon les besoins) :

rsync-backup-include2

```
# La commande rsync utilisée inclue / (root) et exclue tout. Cela ne
sauvegarde rien. Le fichier rsync-backup-include permet d'inclure ce
que l'on veur sauvegarder.
# (note : pour plus d'information voir la page de manuel de rsync dans
les sections : FILTER RULES, INCLUDE/EXCLUDE PATTTERN RULES, MERGE-FILE
FILTER RULES)
# à la visite d'un dossier, penser à exclure ce que l'on ne veut pas
sauvegarder avant de mettre la règle contenant les *
# + [pattern] inclu un motif
# - [pattern] exclu un motif
# '*' remplace tout et s'arrête au premier slash.
# '**' remplace tout, incluant les slashs.
# + /dossier/ visite un dossier mais ne sauvegarde rien
# + /dossier/ sauvegarde tous les fichiers et dossiers mais de manière
non récursive (les dossiers seront vides)
# + /dossier/** sauvegarde tous le contenu du dossier récursivement
# exclure les fichiers
- /initrd.img
- /vmlinuz

# A titre d'exemple, les deux lignes ci-dessous <USER1> et <USER2> sont
deux répertoires utilisateurs pour lesquels
# on ne souhaite pas sauvegarder le contenu du répertoire .cache.

- /home/<USER1>/.cache
- /home/<USER2>/.cache


# sauvegarder le contenu de /home
+ /home/
+ /home/**
```

```
# exclure tout (comme ça on est sûr)
- **
```

Créer un fichier .clefsauvegarde dans le répertoire <USER> de confiance (le même que celui déclaré à la ligne 102 du script “rsync-backup.sh”) :

```
touch /home/<USER>/Documents/.clefsauvegarde
```

Editer le fichier et y écrire votre mot de passe super secret :

```
nano /home/<USER>/Documents/.clefsauvegarde
```

```
Monmotdepassesupersecret
```

Affecter les droits unique de lecture pour root.

```
chmod 400 /home/<USER>/Documents/.clefsauvegarde
```

Premières sauvegardes avant le grand saut

Nous allons forcer les premières sauvegardes qui vont être naturellement de longues durées.

Effectuer les commandes suivantes:

```
cd /usr/local/sauvegarde
./rsync-backup-user.sh
cd /root/backup
./rsync-backup.sh
```

Le système est sauvegardé sur la BANK1 et le /home dans le répertoire sécurisé.

Lancer à nouveau les mêmes commandes pour sauvegarder le système dans la BANK2 (comme le /home est déjà fait cela devrait prendre moins de temps).

Démonter le disque dur

```
cd /
umount /mnt/sauvegarde
```

Puis démarrer le service en saisissant les commandes suivantes :

```
systemctl daemon-reload
systemctl enable sauvegarde.service
systemctl start sauvegarde.service
```

Éditez votre fichier /etc/fstab avec votre éditeur favori (nano, vim) supprimez le “#” en début de ligne

extrait/etc/fstab

```
UUID=8498c62b-ad6f-4966-990e-822f729a01c3 /mnt/sauvegarde  ext4
defaults,nofail,noauto    0   2
```

Voilà... C'est fini. 😃

Scripts de restauration système

restaurestart.sh

```
#!/bin/bash
# Variables d'environnement
SOURCE="/mnt/restauration"
# Codes d'erreur
E_REPABSENT=65
E_PASDERESEAU=66

# test 1
if [ -d $SOURCE ] ;
        then # le répertoire existe
                echo "Répertoire restauration trouvé."
        else # Le répertoire n'existe pas

                echo "Le répertoire de restauration n'a pas été
trouvé."
                echo "Vérifier si le support de restauration est monté"
                echo "dans le répertoire /mnt/restauration"
        echo" Ensuite relancer le script restaurestart"
        exit $E_REPABSENT
fi # fin test 1
# test 2
if ping -c 1 debian-facile.org &> /dev/null
    then
    echo "Réseau OK"
    else
    echo "A priori le réseau n'a pas accès à internet"
    echo "Vérifier vos paramètres réseau"
    echo
    echo "Lien d'aide pour installer la connexion réseau : "
    echo "https://debian-facile.org/doc:reseau:interfaces"
    exit $E_PASDERESEAU
fi # fin test 2
```

```
echo "Choix répertoire de restauration : 1=BANK1, 2=BANK2"
echo "Une valeur différente de 1 ou 2 = fin programme"
read VALEUR

if [ $VALEUR -eq 1 ];then
BANK="BANK1"
elif [ $VALEUR -eq 2 ];then
BANK="BANK2"
else
exit
fi
echo "Copie du fstab de la sauvegarde au cas où..."
cp -av $SOURCE/$BANK/etc/fstab $SOURCE/$BANK/etc/fstab.bak
echo "Copie du fstab du nouveau système dans la sauvegarde"
cp -av /etc/fstab $SOURCE/$BANK/etc/fstab
echo "Copie le fichier sources.list de la sauvegarde vers le nouveau
système"
cp -av $SOURCE/$BANK/etc/apt/sources.list /etc/apt/sources.list
echo "Copie du cache apt de la sauvegarde vers le nouveau système"
cp -a $SOURCE/$BANK/var/cache/apt /var/cache
apt-get update
echo "************************************"
echo "* Lancer le script restaurepaquets *"
echo "************************************"
exit 0
```

restaurepaquets.sh

```
#!/bin/bash
# Variables d'environnement
SOURCE="/mnt/restauration"
MesPaquets="dpkg-selections.list"
ConfDebian="debconf-selections"
# Codes d'erreur
E_REPABSENT=65
# test 1
if [ -d $SOURCE ] ;
    then # le répertoire existe
        echo "Répertoire restauration trouvé."
    else # Le répertoire n'existe pas

        echo "Le répertoire de restauration n'a pas été trouvé."
        echo "Vérifier si le support de restauration est monté"
        exit $E_REPABSENT
fi # fin test 1

echo "Choix répertoire de restauration : 1=BANK1, 2=BANK2"
echo "Une valeur différente de 1 ou 2 = fin programme"
read VALEUR
```

```
if [ $VALEUR -eq 1 ];then
BANK="BANK1"
elif [ $VALEUR -eq 2 ];then
BANK="BANK2"
else
exit
fi
cd $SOURCE/$BANK
apt-cache dumpavail | dpkg --merge-avail
dpkg --clear-selections
dpkg --set-selections <$MesPaquets
apt-get dselect-upgrade
echo "*********************************************************"
echo "* Terminé : Si besoin exécutez les opérations 1 et 2  *"
echo "* dans l'ordre.                                         *"
echo "*                                                       *"
echo "* 1- Installer manuellement les paquets issus d'autres*"
echo "* dépôts que stable                                     *"
echo "* exemple :                                             *"
echo "* apt-get -t jessie-backports install paquet1 paquet2 *"
echo "*                                                       *"
echo "* 2 - Si vous restaurez également le répertoire /home *"
echo "* faites le maintenant.                                 *"
echo "*                                                       *"
echo "* Ensuite lancer le script restauredebian               *"
echo "*********************************************************"
exit 0
```

restauredebian.sh

```
#!/bin/bash
# Variables d'environnement
SOURCE="/mnt/restauration"

# Codes d'erreur
E_REPABSENT=65
# test 1
if [ -d $SOURCE ] ;
    then # les répertoires existent
        echo "Répertoire restauration trouvé."
    else # Les répertoires n'existent pas

        echo "Le répertoire de restauration n'a pas été trouvé."
        echo "Vérifier si le support de restauration est monté"
        exit $E_REPABSENT
fi # fin test 1

echo "Choix répertoire de restauration : 1=BANK1, 2=BANK2"
```

```
echo "Une valeur différente de 1 ou 2 = fin programme"
read VALEUR

if [ $VALEUR -eq 1 ];then
BANK="BANK1"
elif [ $VALEUR -eq 2 ];then
BANK="BANK2"
else
exit
fi

cd $SOURCE/$BANK
echo "Copie du répertoire usr/local"
cp -a usr/local/sauvegarde /usr/local
echo "Copie du répertoire var/log"
cp -a var/log /var
echo "Copie du répertoire var/mail"
cp -a var/mail /var
echo "Copie du répertoire var/spool"
cp -a var/spool /var
echo "Copie du répertoire etc"
cp -a etc /
echo "Copie du répertoire root"
cp -a root /
echo "****************************************"
echo "* restauration fichiers système terminée *"
echo "* Lancer la commande : shutdown -r       *"
echo "****************************************"
exit 0
```

Mes sources d'inspiration

Les excellents tuto dont je me suis largement inspiré :

rsync-backup https://debian-facile.org/doc:systeme:rsync:backup.

J'ai repris beaucoup de principes de ce tuto, notamment l'admirable et instructive façon d'utiliser rsync (merci à gksam).

Deux tutos expliquant comment faire une sauvegarde - restauration complète du système (donc lourde) :

- https://wiki.debian-fr.xyz/Restauration_ou_Reconstruction_de_sa_Debian_-_Racine_et_Home_s%C3%A9par%C3%A9s
- https://wiki.debian-fr.xyz/Sauvegarder_sa_Debian_en_vue_d%27une_R%C3%A9installation_ou_Reconstruction_-_Racine_et_Home_s%C3%A9par%C3%A9s

La référence debian :

https://www.debian.org/doc/manuals/debian-reference/ch10.fr.html#_backup_and_recovery

Le tuto encfs de ubuntu : https://doc.ubuntu-fr.org/encfs

From:
http://debian-facile.org/ - **Documentation - Wiki**

Permanent link:
http://debian-facile.org/utilisateurs:philou92:tutos:sauvegarde-systeme-leger

Last update: **01/05/2023 22:43**