

Réalisation du Script de Sauvegarde

Réalisation du Script :

J'ai créé le script de sauvegarde avec tous les points demandés :

Tout d'abord nous avons utilisé des fonctions pour simplifier le script et que cela soit plus facile à exécuté. De plus, nous devons installer **mailutils** avec **apt-get install mailutils**

Nous avons aussi attribué des noms spécifiques à nos variables :

```
#!/bin/bash
#On définit nos variables, et nos lieux où se trouvent nos répertoires
BACKUP_AP=/etc/apache2
BACKUP_WWW=/var/www
BACKUP_DOSS=/home/adminserveur/Desktop/Backup
DATE=`date +%d-%m-%y_%HH%M`
LOG_FILE=/var/log.txt
```

- BACKUP-AP = Dossier d'apache2
- BACKUP_WWW = Dossier du www
- BACKUP_DOSS = Dossier de stockage de sauvegarde
- DATE = On fixe la variable qu'énoncera la date
- LOG_FILE = Fichier texte où se trouve les logs (ce qui se passe)

Mon script permet de faire une sauvegarde complète tous les six mois :

```
save_comp()
{
echo -e "--- SAUVEGARDE COMPLETE ---" >> $LOG_FILE
#On regarde dans nos répertoires s'il y a des dossiers ou répertoires à sauvegarder, si il ny en à pas on ne sauvegarde pas
NBRE_FILES=`find $BACKUP_AF -type f 2>/dev/null | wc -l`
if [[ $NBRE_FILES == 0 ]]
then
echo -e "Il n'y a aucun fichier à sauvegarde dans $BACKUP_AF, donc pas de sauvegarde. \n" >> $LOG_FILE
else
#On regarde si le dossier où l'on met nos sauvegardes existe
if [ -e $BACKUP_DOSS ]
then
echo -e "Le dossier de sauvegarde existe" >> $LOG_FILE
#La commande tar permet de copier et de compresser le fichier puis le coller dans le répertoire destination
tar -czf $BACKUP_DOSS/SAVE_COMP_$DATE.tgz $BACKUP_AP $BACKUP_WWW 2>/dev/null
echo -e "Sauvegarde du $DATE bien créé. \n" >> $LOG_FILE
else
echo -e "Le dossier de sauvegarde n'existe pas. Création en cours... Dossier créé. \n" >> $LOG_FILE
#On crée le repertoire savecomp dans le lieux où se trouve le disque dur externe
mkdir "$BACKUP_DOSS"
tar -czf $BACKUP_DOSS/SAVE_COMP_$DATE.tgz $BACKUP_AP $BACKUP_WWW 2>/dev/null
echo -e "Sauvegarde du $DATE bien créé. \n" >> $LOG_FILE
fi
fi
}
```

De faire des sauvegardes incrémentales tous les jours avec le **crontab** :

```
save_incr()
{
echo -e "--- SAUVEGARDE INCREMENTALE ---" >> $LOG_FILE
#On regarde dans nos dossiers de configuration si il y a eu des modifications durant les dernières 24h
NBRE_FILES=`find $BACKUP_AP -type f -mtime -1 2>/dev/null | wc -l`
NBRE_FILES2=`find $BACKUP_WWW -type f -mtime -1 2>/dev/null | wc -l`
if [ $NBRE_FILES == 0 ] && [ $NBRE_FILES2 == 0 ]
then
echo -e "Il n'y a pas eu de modifications dans les dossiers. PAS DE SAUVEGARDE." >> $LOG_FILE
else
if [ $NBRE_FILES != 0 ] && [ $NBRE_FILES2 != 0 ]
then
#Où il y a des "tar" on copie et compresse seulement les changements depuis les dernières 24h
echo -e "Les deux dossiers ont été modifié. SAUVEGARDE EN COURS..." >> $LOG_FILE
tar -czf $BACKUP_DOSS/SAVE_INC_$DATE.tgz `find $BACKUP_AP $BACKUP_WWW -type f -mtime -1` 2>/dev/null
else
if [[ $NBRE_FILES == 0 ]]
then
echo -e "Le fichier WWW a été modifié. SAUVEGARDE EN COURS..." >> $LOG_FILE
tar -czf $BACKUP_DOSS/SAVE_INC_$DATE.tgz `find $BACKUP_WWW -type f -mtime -1` 2>/dev/null
else
echo -e "Le fichier Apache2 a été modifié. SAUVEGARDE EN COURS..." >> $LOG_FILE
tar -czf $BACKUP_DOSS/SAVE_INC_$DATE.tgz `find $BACKUP_AP -type f -mtime -1` 2>/dev/null
fi
fi
fi
}
```

Ensuite on devait faire en sorte de d'envoyer un mail lorsqu'il y a une erreur de sauvegarde :

Il nous fallait aussi vérifier l'espace disque pour savoir si on peut refaire une sauvegarde :

```
espace_disque()
{
#On affiche l'espace disque, la ligne qu'on veut ainsi que la colonne pour récupérer le % d'espace pris
ESPACE=`df -h / | sed -n 2p | cut -c40-41`
if [[ $ESPACE -ge 90 ]]
then
echo -e "L'espace disque est presque plein. Attention !" >> $LOG_FILE
fi
}

limit_backup()
{
#Ici, on crée une variable qui va afficher le nombre de ligne contenant les caractères SAVE_INC
LIMITE=`ls -lt $BACKUP_DOSS | grep SAVE_INC | wc -l`
#Si le nombre de ligne est supérieur à 3, on rentre dans le dossier où on a nos backup et on supprime le fichier le plus vieux
if (( $LIMITE >= 3 ))
then
cd /var/run/savecomp
rm -f $(ls -lt | tail -1)
echo "Suppression de la sauvegarde la plus vieille" >> $LOG_FILE
fi
}
```

Pour finir nous devons faire en sorte que nous ne garderons que les 6 derniers mois de sauvegarde et effacerons les plus anciennes à chaque fois :

Ici nous n'avons pas fait avec 183 jours donc 6 mois, mais avec seulement l'équivalent de 3 jours pour pouvoir bien montrer le résultat. (Nous pouvons facilement le changer en modifiant la Limite dans notre IF.

Toutes ces parties sont des fonctions que j'appelle dans mon script avec :

```
echo -e " --- Backup du `date +%D` à `date +%H:%M` ---" >> $LOG_FILE
#On appelle nos fonctions
limit_backup
espace_disque
#On regarde si une sauvegarde complète existe
NBRE_FULL=`ls $BACKUP_DOSS/SAVE_COMP_* 2>/dev/null | wc -l`
if [[ $NBRE_FULL > 0 ]];
then
    echo -e "Il y a une sauvegarde complète on peut donc faire des incrémentales" >> $LOG_FILE
    save_incr
    mailsave
else
    save_comp
fi
```

Cependant, pour que notre script fonctionne il nous faut donner les permissions à notre script en faisant **chmod 777 backup.sh**, ensuite on a plus qu'à exécuter le script avec **./backup.sh** (ou seulement **backup.sh** s'il est dans /usr/bin) ou sinon on peut le faire automatiquement tous les jours comme il le faut en mettant la commande :

0 0 * * * /etc/backup.sh dans le **crontab -e**.