

Description des scripts de sauvegarde

Linux

```
-rwX----- 1 root root 439 févr. 11 14:03 sauvegarde_complete.sh
-rwx----- 1 root root 527 févr. 11 14:04 Sauvegardes.sh
root@debian:/etc/cron.daily#
```

Pour commencer nous avons créé les fichiers qui vont contenir les scripts avec la commande : `touch "nomdefichier".sh`

Nous avons ensuite donné les droits de lecture, d'écriture et d'exécution au fichier pour son propriétaire (ici le propriétaire est le super utilisateur "root").

```
GNU nano 2.7.4 Fichier : sauvegarde complete.sh
#!/bin/bash

backupdate=$(date +%Y-%m-%d)

#répertoire de backup
#Variable qui contient le chemin
Variable=/home/jules/Test/Projet/Saves/$backupdate
#On crée un répertoire à chaque fois que le script s execute donc à chaque fois que l'on sauvegarde les données

/bin/mkdir $Variable
#tar permet de créer un fichier compressé donc une archive

/bin/tar -cjf $Variable/$backupdate.tar.bz2 /home/jules/Test/Projet/Fichier_Sauvegardes
```

Ensuite nous avons écrit le script de sauvegarde complète :

`#!/bin/bash :`

Cette ligne est l'indication que le script devra être exécuté avec le shell bash.

`Backupdate=$(date +%Y-%m-%d) :`

Ici on crée la variable "backupdate" et on lui assigne la date du jour où le script est exécuté sous la forme Année/Mois/Jour.

Variable=/home/jules/Projet/Saves/\$backupdate :

Nous créons la variable “**Variable**” à laquelle nous assignons le **chemin du répertoire dans lequel les fichiers seront sauvegardés**. A la fin du chemin nous rajoutons /\$backupdate, de sorte que le script crée un répertoire dont le nom sera la date de la sauvegarde (\$backupdate étant la valeur de la variable backupdate) ce qui permettra une meilleure organisation des fichiers dans le répertoire

/bin/mkdir \$Variable:

Cette commande va permettre de créer le répertoire contenu dans la variable « Variable », or les répertoires /home/jules/Test/Saves existent déjà, il sera alors uniquement créé le répertoire \$backupdate que le script nommera par rapport à la date de sa création dans le répertoire Saves.

/bin/tar -cjf \$Variable/\$backupdate.tar.bz2 :

```
/bin/tar -cjf $Variable/$backupdate.tar.bz2 /home/jules/Test/Projet/Fichier_Sauvegardes
```

Cette ligne est la commande en charge de **la création d’une archive**, la commande va créer une archive dans le répertoire ciblé par la variable « Variable », le nom de l’archive sera la date contenue dans la variable backupdate avec l’extension .tar.bz2 correspondant à l’extension des archives sur Linux. Ensuite il suffit d’entrer le chemin des fichiers à sauvegarder à la suite de cette commande comme suit dans notre cas :

/home/jules/Test/Projet/Fichier_Sauvegardes

```
GNU nano 2.7.4                                Fichier : Sauvegardes.sh
#!/bin/bash
#là ou on enregistre:
CIBLE=/home/jules/Documents/Sauvegardes1
#variable contenant la date
DATE=$(date +%Hh%M-%d-%B-%Y)

cd
cd /home/jules/Test/Projet/Fichier_Sauvegardes #endroit où le fichier à sauver se trouve
#on trouve f et doss ,executer la commande cp à l endroit cible avec la date
find -mtime -1 -exec cp -r {} $CIBLE/$DATE \;
#ensuite on doit supprimer les sauvegardes de plus de 6 mois
cd
cd $CIBLE

#6 mois ->6*30= 180 jours, print0=argument
find -type d -mtime +180 -print0 | xargs -0 /bin/rm -rf
```

Ici, on voit comment on va sauvegarder de façon incrémentielle.

On initie d'abord la variable CIBLE contenant le chemin du répertoire où l'on enregistrera les modifications.

On initie ensuite la variable DATE contenant la fonction date avec ses arguments %Hh(heure) %M(mois) %d (jour du mois) %B (Mois complet, cad prend en compte le fait que le mois comporte plus ou moins 30 jours) %Y (année format aaaa)

Le \$date est une commande avec laquelle le + %H... définit le format de la date à afficher dans la variable DATE

Cd /home/.../Fichier_Sauvegardes : chemin du répertoire à sauver.

```
Find -mtime -1 -exec cp -r {} $CIBLE/$DATE \; :
```

On cherche fichiers et dossiers vieux de moins d'un jour et on les copie de manière récursive à l'endroit cible avec la date

Enfin on s'occupe de la partie suppression :

```
find -type d -mtime +180 -print0 | xargs -0 /bin/rm -rf :
```

on recherche tous les répertoires de datés de plus de 6 mois puis avec l'option print0 qui affiche le nom complet du fichier sur la sortie standard, suivi d'un

caractère nul, permettant aux fichiers dont le nom contient un saut de ligne, d'être correctement interprétés par les programmes utilisant les données en sortie de **find**.

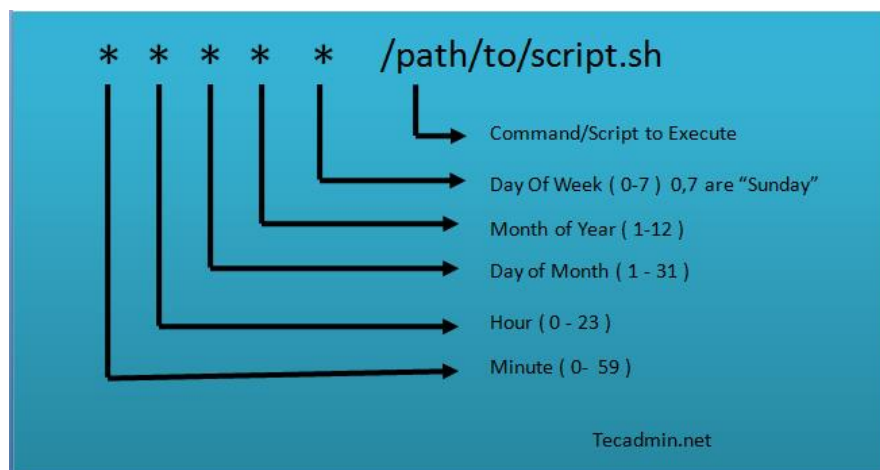
Puis comme on utilise une commande qui prends des arguments en entrée, on va utiliser **xargs**, un petit outil pratique pour chaîner des commandes qui prennent des arguments comme entrée.(**xargs** est une commande UNIX puissante qui permet de récupérer les arguments en sortie de la commande précédente et de les traiter à travers un *pipe*.) Ici on chaîne avec la commande **rm -rf** dans le dossier **bin/** qui va supprimer tous les dossiers de plus de 180 jours

Après réflexion, l'utilisation de cron pour sauvegarder toutes les 2 semaines semble compromise. En effet cron est une commande qui certes peut s'utiliser simplement en mettant le script à exécuter dans les répertoires suivants :

1. `/etc/cron.d`
2. `/etc/cron.hourly`
3. `/etc/cron.daily`
4. `/etc/cron.weekly`
5. `/etc/cron.monthly`

(La syntaxe étant assez explicite je ne développerai pas plus.)

Mais d'autre part `crontab -e` (-e précise que c'est l'utilisateur actuel qui exécute la commande) permet aussi de planifier une tâche avec néanmoins une syntaxe particulière :



Un exemple dans vim :

```
00 15 * * * php monfichier.php
```

Minute- Heure - JourDuMois - Mois – JoursDeLaSemaine commande

Le problème suivant se pose dès lors : comment sauvegarder une semaine sur deux ?

En ayant fait plusieurs recherches, il en ressort une utilisation de flags ce qui peut être considéré comme du bricolage.

Ceci étant dit, comme il est possible de faire des sauvegardes complètes plus fréquemment, il sera préférable de passer à une sauvegarde complète par semaine ce qui est très simple à faire :

```
* * * * 5 /mnt/hgfs/sauvegarde_complete.sh 2>>/mnt/hgfs/Sauvegarde erreur.log  
* 22 * * * /mnt/hgfs/Sauvegardés.sh 2>> /mnt/hgfs/Sauvegarde erreur.log
```

le /mnt/hgfs/comande indique les commandes à exécuter