

Sauvegarde d'un dossier sur un serveur distant

Le script intégrale est à la fin de ce document.

1. Prérequis

- Créer un dossier save dans le répertoire personnel de l'utilisateur sur le serveur web (172.28.100.1).
- Créer un dossier save_www dans le répertoire personnel de l'utilisateur sur le serveur backup (172.28.100.2).
- Placer le script save.bash dans le répertoire save du serveur web.
- Créer une relation privilégiée entre le serveur web et le serveur backup:

Commande à exécuter sur le serveur web.

```
ssh-keygen -t rsa  
ssh-copy-id -i ~/.ssh/id_rsa.pub `whoami`@172.28.100.2
```

Ceci permettra d'exécuter le script automatiquement sans avoir à entrer le mot de passe de la connexion ssh.

Cron

Pour automatiser l'exécution du script on utilise le planificateur de tâches cron.
on ajoute la ligne suivante au crontab de l'utilisateur:

```
0 22 * * fri /home/`whoami`/save.bash
```

Cette ligne permet l'exécution du script tous les vendredis à 22h00.

Pour plus d'informations sur le service cron, consulter la [fiche technique](#).

2. Variables utilisé

Variables d'environnement:

\$USER

nom de l'utilisateur courant (peut être remplacé par la commande ``whoami``)

\$HOSTNAME

nom de l'ordinateur (peut être remplacé par la commande ``uname -n`` ou par ``hostname``)

Pour récupérer l'heure et la date on utilise la commande `date`:

```
$date +%Y%m%d-%H%M%S
```

Cette commande retourne: AAAAMMJJ-hhmmss

Pour plus de simplicité je crée une variable contenant le nom de l'archive à créer.

```
nomArchive=`date +%Y%m%d-%H%M%S`_www_`whoami`.tar.gz
```

Le caractère (```) permet d'intégrer le résultat d'une commande.

3. Création et envoi de l'archive

On crée une archive du dossier à sauvegarder au format `.tar.gz` :

```
tar cvzf ~/save/$nomArchive -C ~/ ./www/
```

l'option `-C` permet de spécifier l'emplacement à partir duquel exécuter la commande, ce qui revient à exécuter la commande `cd ~/` avant la commande `tar`.

Puis on l'envoie par ssh via la commande `scp` sur le serveur de backup dans le dossier `~/save_www/` :

```
scp ~/save/$nomArchive 172.28.100.2:~/save_www/
```

Et on supprime l'archive en local:

```
rm ~/save/$nomArchive
```

4. Vérification de la sauvegarde et complétion du fichier de log

On vérifie l'existence de l'archive sur le serveur distant:

```
ssh -q 172.28.100.2 [[ -e ~/save_www/$nomArchive ]] &&
echo "`date +%d/%m/%Y`      `date +%H:%M:%S`      `whoami`@$HOSTNAME
Sauvegarde effectuée" >> ~/save/.save.log ||
echo "`date +%d/%m/%Y`      `date +%H:%M:%S`      `whoami`@$HOSTNAME
Une Erreur est survenue." >> ~/save/.save.log;
```

description de la commande:

```
ssh -q 172.28.100.2 [[ -e ~/save_www/$nomArchive ]] ...///
```

On envoie la commande `[[-e ~/save_www/$nomArchive]]` sur le serveur de backup à travers ssh, l'option -q de ssh permet de ne pas renvoyer les messages d'avertissement et de diagnostics.

Cette commande est équivalente à la commande `test -e ~/save_www/$nomArchive` (les crochets `[]` remplace test), elle permet de vérifier l'existence d'un fichier grâce à l'option -e.

Les doubles crochets `[[]]` permettent l'utilisation de `&&` et `||` en tant que if, then, else.

Si le fichier existe, la commande `[[-e ~/save_www/$nomArchive]]` renvoi le code succès (le code succès est contenu dans la variable `$?` qui contient la valeur 0 si succès.),

La commande située à droite de `&&` est exécutée si condition vraie.

La commande située à droite de `||` est exécutée si autre (faux).

5. Suppression des sauvegardes antérieur à 30 jours

```
ssh -q 172.28.100.2 'find ~/save_www/ -type f -ctime +30 -exec rm {} \;'
```

On envoie la commande sur le serveur de backup à travers ssh.

On exécute la commande find en spécifiant le type, ici fichier (`-type f`) et que l'on cherche les fichiers dont la dernière modification date de plus de 30 jours (`-ctime +30`).

Sur le retour de cette commande on exécute la commande rm et on supprime tous les fichiers renvoyés par la commande find.

6. Script intégral

```
#!/bin/bash
#nom : save.bash
#Sauvegarde dossier www user vers serveur backup

nomArchive=`date +%Y%m%d-%H%M%S`_www_`whoami`.tar.gz

#On crée une archive du dossier www
tar cvzf ~/save/$nomArchive -C ~/ ./www/

#On copie l'archive vers le serveur de backup
scp ~/save/$nomArchive 172.28.100.2:~/save_www/

#On supprime l'archive en local
rm ~/save/$nomArchive

#On vérifie que la sauvegarde a fonctionné et on complète le fichier de
log
ssh -q 172.28.100.2 [[ -e ~/save_www/$nomArchive ]] && echo "`date
+%d/%m/%Y`      `date +%H:%M:%S`      `whoami`@$HOSTNAME      Sauvegarde
effectuée" >> ~/save/.save.log || echo "`date +%d/%m/%Y`      `date
+%H:%M:%S`      `whoami`@$HOSTNAME      Une Erreur est survenue." >>
~/save/.save.log;

#On supprime les sauvegardes du serveur backup antérieur à 30 jours
ssh -q 172.28.100.2 'find ~/save_www/ -type f -ctime +30 -exec rm {} \;'
```

6. Rsync

Une autre méthode pour une sauvegarde et l'utilisation de rsync, ce logiciel permet une synchronisation d'un dossier, donc ne transfère pas les éléments identiques.

rsync utilise SSH. Utilisation :

```
$rsync -az <dossierSource> <login>@<serveur>:<dossierDestination>
```

option :

- `-a` ou `--archive` : est un moyen rapide de dire que vous voulez la récursivité et préserver pratiquement tout.
- `-z` ou `--compress` : compresse les données lors du transfert.