

Livre Blanc

*Script (Bash) de sauvegarde d'une archive
sur un serveur distant.*



Sommaire

- **Mise en oeuvre**
 1. Prérequis
 2. Cron
 3. Script d'install

- **Description du script**
 1. Variables utilisés
 2. Création et envoi de l'archive
 3. Vérification de la sauvegarde et complétion du fichier de log
 4. Suppression des sauvegardes antérieur à 30 jours
 5. Script intégrale

- **Recette**

Cahier des charges.

- le script est écrit en bash ;
- les sauvegardes sont placées dans un répertoire prévu à cet effet ;
- les sauvegardes sont nommées ainsi : <horodatage>_www_<login>.tar.gz ;
- le fichier de log associé doit être complété à chaque création d'une sauvegarde ;
- le script s'exécute automatiquement une fois par semaine ;
- il faut éliminer automatiquement les sauvegardes de plus de 30 jours ;

Mise en œuvre.

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.

2. 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](#).

3. Script d'install

```
#Création de la connection privilégié
ssh-keygen -t rsa
ssh-copy-id -i ~/.ssh/id_rsa.pub `whoami`@172.28.100.2

#ajout du crontab
echo "0 22 * * fri /home/`whoami`/save.bash" >> ~/save/save.temp
crontab ~/save/save.temp
rm ~/save/save.temp
```

Description du script:

1. 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.

2. 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
```

3. 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).

4. 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`.

5. 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 {} \;'
```

Recette

Création de l'archive

Commande:

```
#On crée une variable contenant le nom de l'archive
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/
```

Résultat:

```
mscaffidi@webServerSI02022:~/save$ ls
20210129-080846_www_mscaffidi.tar.gz
```

Envoi de l'archive

Commande:

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

Résultat:

```
mscaffidi@webServerSI02022:~/save$ scp ~/save/$nomArchive 172.28.100.2:
~/save_www/
20210129-080846_www_mscaffidi.tar. 100% 37MB 109.6MB/s 00:00
mscaffidi@backupServerSI02022:~/save_www$ find -ctime -1
./20210129-080846_www_mscaffidi.tar.gz
```

complétion du fichier de log

Commande:

```
#On vérifie que la sauvegarde a fonctionnée 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;
```

Résultat:

```
mscaffidi@webServerSI02022:~/save$ tail -n 1 .save.log
29/01/2021 08:23:24 mscaffidi@webServerSI02022 Sauvegarde effectuée
```

Suppression des sauvegardes antérieur à 30 jours

Commande:

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

exemple avec fichier de plus de 5 jours:

Fichiers:

```
mscaffidi@backupServerSI02022:~/save_www$ find ~/save_www/ -type f  
-ctime +5  
/home/mscaffidi/save_www/20210122-102901_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-102701_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-220001_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-103501_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-103201_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-103001_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-103101_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-103401_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-102601_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-102801_www_mscaffidi.tar.gz  
/home/mscaffidi/save_www/20210122-103301_www_mscaffidi.tar.gz
```

```
mscaffidi@backupServerSI02022:~/save_www$ ls -l  
total 489840  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 déc. 18 22:00  
20210122-102501_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:26  
20210122-102601_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:27  
20210122-102701_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:28  
20210122-102801_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:29  
20210122-102901_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:30  
20210122-103001_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:31  
20210122-103101_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:32  
20210122-103201_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:33  
20210122-103301_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:34  
20210122-103401_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 10:35  
20210122-103501_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38582796 janv. 22 22:00  
20210122-220001_www_mscaffidi.tar.gz  
-rw-r--r-- 1 mscaffidi mscaffidi 38583756 janv. 29 08:14  
20210129-080846_www_mscaffidi.tar.gz
```


Commande:

```
ssh -q 172.28.100.2 'find ~/save_www/ -type f -ctime +5 -exec rm {} \;'
```

Résultat:

```
mcaffidi@backupServerSI02022:~/save_www$ ls -l
total 75360
-rw-r--r-- 1 mcaffidi mcaffidi 38582796 déc. 18 22:00
20210122-102501_www_mcaffidi.tar.gz
-rw-r--r-- 1 mcaffidi mcaffidi 38583756 janv. 29 08:14
20210129-080846_www_mcaffidi.tar.gz
```

Cron

La configuration du crontab permet une exécution du script tous les vendredis à 22h.

```
mcaffidi@backupServerSI02022:~/save_www$ ls -l
total 150720
-rw-r--r-- 1 mcaffidi mcaffidi 38582796 janv. 8 22:00
20210108-220001_www_mcaffidi.tar.gz
-rw-r--r-- 1 mcaffidi mcaffidi 38583756 janv. 15 22:00
20210115-220001_www_mcaffidi.tar.gz
-rw-r--r-- 1 mcaffidi mcaffidi 38583756 janv. 22 22:00
20210122-220001_www_mcaffidi.tar.gz
```