

# Synchronisation et sauvegarde de données

Grâce à rsync on peut synchroniser des fichiers/dossiers entre différents postes. L'intérêt est de répliquer des fichiers/dossiers de configuration, ou pour effectuer des sauvegardes par exemple.

Le serveur Rsync devra avoir beaucoup de mémoire, puisque nous stockerons tous les fichiers/dossier synchronisé dessus. Se sera le serveur de « sauvegarde rsync »...

Voici la configuration requise :

- un serveur rsync: récupère les fichiers en se synchronisant avec le client.
- au moins 1 client rsync.
- Les acteurs doivent pouvoir communiquer entre eux (ping icmp, ssh...).
- Les machines doivent être à jour et connectées à internet.



*Objectif général : Le but est de synchroniser les fichiers/dossiers d'un client sur un serveur rsync à l'aide d'un script lancé automatiquement chaque minute/heure/jour/mois... Puis nous verrons comment automatiser le stockage de ces données.*

# **Sommaire**

## **I- SSH**

## **II- Script de synchronisation**

## **III- Script de sauvegarde**

## **IV- Automatisation CRON**

## **V- Restauration**

## **VI- Bonus**

## **VII- Sources**

## I- SSH

*Objectif : Echange de clé pour connexion du serveur vers le client sans demande de mot de passe.*

Sur le client et le serveur :

- Installer le paquet openssh-server :
- ```
apt-get install openssh-server
```

Sur le client :

- Se loguer avec l'utilisateur propriétaire des fichiers/dossiers de configuration.
- Se connecter en ssh sur le serveur rsync.
- Une fois que l'échange des clés est établi **deconnexion**.

Sur le serveur :

- Se loguer en root.

- Génération de la clé :

```
ssh-keygen -t rsa
```

→ *Entrée, Entrée, Entrée*

- Copie de la nouvelle clé sur le client :

```
scp /root/.ssh/id_rsa.pub [user]@[IP_destination]:/[user]/.ssh/authorized_keys2
```

(N.B : Si le propriétaire est le compte « root » il faut activer la permission de se connecter en ssh sur la machine cliente, pour cela il faut modifier le fichier /etc/ssh/sshd\_config → « PermitRootLogin yes », puis « service ssh reload » sur le client)

- Se connecter en ssh depuis le serveur sur le client pour vérifier, aucun mot de passe ne devrait être demandé. Si un mot de passe est demandé c'est qu'il y a une erreur.

## II- Script de synchronisation

*Objectif : Le script lancera les commandes qu'il contient. Ces commandes feront la synchronisation entre les fichiers/dossiers du client et les fichiers/dossiers du serveur.*

Sur le client et le serveur :

- Installer le paquet rsync :
- ```
#apt-get install rsync
```

Sur le serveur :

- Créer un répertoire où recevoir les fichiers/dossiers qui seront synchronisés. (Bien organiser les noms de dossiers...)  
→ **/synchro/[nom\_de\_machine]/**
- Créer un répertoire où contenir les scripts (Bien organiser les noms de dossiers...)  
→ **/script/[nom\_de\_machine]/synchro.sh**

- Création du script :

```
#touch [chemin_synchro.sh]
#chmod +x [chemin_synchro.sh]
```

- Remplir le script :

```
#!/bin/bash
rsync -atz -e ssh [user]@[IP_client]:[chemin_destination_client]
[chemin_destination_serveur]
```

→ Le [chemin\_destination\_serveur] correspond à « synchro/[nom\_de\_machine]/

### III- Script de sauvegarde

*Objectif : Le script lancera les commandes qu'il contient. Ces commandes feront la sauvegarde des fichiers précédemment synchronisés. Ces fichiers seront stockés dans un dossier au jour de la sauvegarde effectuées.*

- Verifier que la machine serveur est à l'heure :  
#date
- Créer un repertoire où recevoir les fichiers/dossiers de sauvegardes. (Bien organiser les noms de dossiers...)  
→ /sauvegardes/[nom\_de\_machine]/

- Création du script :

```
#touch [chemin_sauv.sh]
#chmod +x [chemin_sauv.sh]
```

- Remplir le script

```
#!/bin/bash
#Se placer dans le dossier ou les fichiers rsync sont receptionne
cd /synchro/[nom_de_machine]/

#Initialisation de la variable 'da' avec la date des fichiers
da=`date +%m_%d_%G`
echo "La date est : $da"

#Création du dossier à la date reçue des fichiers synchro dans la
#variable 'da'
mkdir /sauvegardes/[nom_de_machine]/$da

cd /synchro/[nom_de_machine]/

#Couper/coller des fichiers synchro
mv * /sauvegardes/[nom_de_machine]/$da/
```

## IV- Automatisation CRON

*Objectif : Automatiser le lancement du script qui synchronise les données entre clients/serveur.*

### Sur le serveur

- Accéder à la table CRON :

```
#crontab -e
```

Choisir « 1 » pour l'éditeur de texte « nano ».

- Exemples de configuration pour choisir la fréquence des sauvegardes :

```
# m h dom mon dow command
```

- m est pour minutes [0-59];
- h est pour heure [0-23];
- dom est pour journée du mois [1-31];
- mon est pour mois, de janvier à décembre [1-12];
- dow est pour journée du dimanche au samedi [0-6];
- command est le chemin d'accès *absolu* de votre script

Placez votre curseur sous la première colonne et entrez la règle que vous souhaitez. Notez qu'il ne doit pas y avoir plus d'un espace entre chaque colonne.

## Exemples de règles *cron*

### 1. Exécuter un script à tous les jours à minuit

```
# m h dom mon dow command
0 0 * * * /home/itx/scripts/script_a_executer.sh
```

### 2. Exécuter un script le lundi à 15h00

```
# m h dom mon dow command
0 15 * * 1 /home/itx/scripts/script_a_executer.sh
```

### 3. Exécuter un script à toutes les heures

```
# m h dom mon dow command
0 */1 * * * /home/itx/scripts/script_a_executer.sh
```

### 4. Exécuter un script à toutes les 15 minutes

```
# m h dom mon dow command
*/15 * * * * /home/itx/scripts/script_a_executer.sh
```

- Dans notre configuration deux scripts sont utilisés : d'abord [synch.sh] puis [sauv.sh]. La syntaxe utilisée dans crontab pour que deux scripts s'exécutent l'un après l'autre est :  
\*/ \* \* \* \* \* [chemin\_synch.sh] ; [chemin\_sauv.sh]

## V- Restauration

*Objectif : Restaurer un fichier synchronisé depuis le serveur rsync sur le client.*

- Commande :

```
#scp [chemin_destination_serveur] [user]@[IP_destination_client]:  
[chemin_destination_fichier]
```

→ **scp** /sauvegardes/[nom\_de\_machine]/[date\_dossier]/[fichier\_souhaité]  
[user]@[ip\_destination\_client]

## VI- Bonus

Une autre solution de sauvegarde : « diff-backup » ...

Voir : <https://doc.ubuntu-fr.org/rdiff-backup>

Plus général: <https://doc.ubuntu-fr.org/sauvegarde>

## VII- Sources

[http://lea-linux.org/documentations/Trucs:Connection\\_SSH\\_sans\\_mot\\_de\\_passe](http://lea-linux.org/documentations/Trucs:Connection_SSH_sans_mot_de_passe)

<https://wiki.debian-fr.xyz/Rsync>

<https://doc.ubuntu-fr.org/rsync>

<http://www.thegeekstuff.com/2010/09/rsync-command-examples/>

<https://openclassrooms.com/courses/reprenez-le-controle-a-l-aide-de-linux/executer-un-programme-a-une-heure-differee>

<http://www.progmaticque.fr/article-122-Batch-date-heure.html>

<http://hardware-libre.fr/2014/03/8-exemples-pour-maitriser-linux-cron/>

<https://itx-technologies.com/blog/2259-executer-un-script-recurrent-avec-cron-exemples-sous-linux>

[http://forum.hardware.fr/hfr/OSAlternatifs/Codes-scripts/avec-date-bash-sujet\\_33358\\_1.htm](http://forum.hardware.fr/hfr/OSAlternatifs/Codes-scripts/avec-date-bash-sujet_33358_1.htm)

<http://www.progmaticque.fr/article-122-Batch-date-heure.html>