

# Mini-Projet : Utilitaire d'archive


## Guide d'installation & d'utilisation

### I - Configuration

- Machine virtuel : Linux Debian 11 (bullseye)
- Language : Python3 et Bash
- Serveur web : Apache
- Méthode de transfert : SFTP avec openSSH
- Automatisation : Crontab
- Méthode email : SMTPS avec gmail
- Fichier de configuration : format json

```
mohamed@debian:~$ neofetch
_,met$$$$$gg.
,g$$$$$$$$$$$$$P.
,g$$$P"        ""Y$.
,$$P'          $$$
',$$P          ,ggs.  $$$b:
`d$$'          ,P"'    $$$
$$P            d$'     $$P
$$:            $$.    -  ,d$$'
$$;            Y$b._   ,d$P'
Y$$.          '  "Y$$$$P"'
`$$b          "-._
`Y$$
`Y$$
`$$b.
`Y$$b.
`"Y$b._
  ""

mohamed@debian
-----
OS: Debian GNU/Linux 11 (bullseye) x86_64
Host: VirtualBox 1.2
Kernel: 5.10.0-18-amd64
Uptime: 2 mins
Packages: 1682 (dpkg)
Shell: bash 5.1.4
Resolution: preferred
DE: GNOME 3.38.6
WM: Mutter
WM Theme: Adwaita
Theme: Adwaita [GTK2/3]
Icons: Adwaita [GTK2/3]
Terminal: gnome-terminal
CPU: Intel i5-10300H (1) @ 2.495GHz
GPU: 00:02.0 VMware SVGA II Adapter
Memory: 512MiB / 976MiB
```



### II - Prérequis :

0. Une machine sous Linux avec une distribution Debian ou basé sur Debian.
1. Les adresses IP des serveurs doivent être statique.
2. Exécuter en *sudo* le fichier *dependances.sh* afin de télécharger automatiquement toutes les dépendances nécessaires au bon fonction de l'utilitaire.

### III - Configuration du script :

Pour paramétrer l'exécution du script, il faut modifier le fichier *configuration.json* qui doit rester dans le répertoire *Main* qui fait tourner le script principal. Nous allons expliciter les champs un à un.

- `url_fichier` : lien vers le fichier sur le serveur web
- `ip_machine` : Ip de la machine où l'on stocks les fichiers en SFTP
- `user_sftp` : utilisateur de la machine SFTP
- `mdp_sftp` : mot de passe de la machine SFTP
- `chemin_Programme` : Chemin dans lequel se trouve le programme principal
- `chemin_sftp` : Chemin où seront stockés les fichiers à conserver sur la machine en SFTP
- `periode` : période à laquelle doit s'exécuter le programme principal. **Il faut exécuter le fichier *automatisation\_crontab.sh* après chaque modification de la période ou du chemin du programme.**

Arguments possibles pour le champs *période* voir [Crontab](#):

- `@reboot`
- `@yearly`
- `@annually`
- `@monthly`
- `@weekly`
- `@daily`
- `@midnight`
- `@hourly`
- Mail :
  - `email` : adresse mail depuis laquelle le mail pourra être envoyé
  - `key` : clé ou mot de passe de l'adresse mail
  - `Serveur_smtp` : serveur smtp utilisé pour envoyer des mails (avec une adresse google, on utilise les serveurs smtp de google)
  - `port_smtp` : ports associé au serveur smtp (pour google c'est 465)
  - `envoi_mail` : booléen qui indique si oui ou non il faut envoyer un mail après l'exécution du script
  - `logs_mail`: booléen qui indique si oui ou non on envoie en pièce jointe du mail les logs
  - `Objet_mail_reussi` : l'objet du mail en cas de succès du script
  - `Objet_mail_echec` : l'objet du mail en cas d'échec du script
  - `destinataires_mail` : le/les destinataire(s) du mail
- `historisation` : booléen qui indique si oui ou non la sauvegarde de plusieurs versions sont permises.
- `periode_suppression` : en jours, indique la durée de vie des fichiers une fois qu'ils sont archivés sur la machine en sftp

## IV - Mise en place du serveur web avec chiffrement SSL

### Choix retenu pour le serveur : Apache2

Mise en place du serveur Apache:

1. Installer les paquets/dépendances nécessaire en Executant en *sudo* le fichier *dependance.sh* sur chacune des machines.
2. Executer en *sudo* le fichier *publish\_WebServer.sh*, le fichier 'test100.sql.zip' à upload doit être dans le répertoire */var/www/html* de la machine qui hébergera le serveur.
3. Mettre les bon paramètres dans le fichier de configuration *configuration.json* et le placer avec le script principal.

Mise en place du chiffrement SSL :

1. Créer un répertoire pour les certificats avec *sudo mkdir /etc/apache2/certs* puis se placer dedans *cd /etc/apache2/certs*.
2. Lancer la génération du certificat avec *sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out apache.crt -keyout apache.key*.
3. Les fichiers *apache.crt* et *apache.key* ont normalement été crée dans le repertoire *certs*.
4. Changer la configuration d'Apache avec *sudo nano /etc/apache2/sites-enabled/000-default.conf* et ajouter :

```
<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/apache2/certs/apache.crt
    SSLCertificateKeyFile /etc/apache2/certs/apache.key
</VirtualHost>
```

5. Pour automatiser la redirection en HTTPS ajouter dans la règle du port 80 :

```
<VirtualHost *:80>
...
    RewriteEngine on
    RewriteCond %{HTTPS} !=on
    RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R=301,L]
</VirtualHost>
```

6. Relancer le serveur apache avec *sudo systemctl restart apache2*.

Le serveur web est opérationnel, on y accède avec [https://\[IP\]](https://[IP]).

## V - Mise en place du serveur de destination

**Choix retenu : Protocole SFTP**

**Sur la machine qui servira de stockage :**

1. Installer OpenSSH avec la commande `sudo apt install openssh-server`
2. Créer un répertoire et ajouter son chemin dans le fichier de configuration
3. Se connecter une première fois à la machine distante en sftp depuis la machine qui exécutera le script principal en **root** en entrant la commande `sftp [user]@[ip]` puis entrer le mot de passe de l'utilisateur puis autoriser la connexion. Vous êtes maintenant connecté en SFTP à la machine qui hébergera vos archives, vous pouvez fermer le terminal.

**NB :** *user* et *ip* ainsi que le mot de passe sont ceux de la machine distante vers laquelle on se connecte en SFTP

## VI - Exécution du script

Une fois toutes les étapes précédentes réalisées on peut passer à la partie exécution du code.

1. Choisir les bons paramètres dans le fichier *configuration.json*
2. Exécuter le fichier *automatisation\_crontab.sh*

**L'utilitaire est maintenant opérationnel et s'exécutera selon la période indiqué dans *configuration.json***

**NB 1 :** Si vous changez la période dans le fichier configuration, il faut réexécuter *automatisation\_crontab.sh*.

**NB 2 :** L'exécution du fichier *automatisation\_crontab.sh* supprime toutes les tâches mise en place avec crontab. Si vous avez d'autres tâches à automatiser, il faut exécuter *automatisation\_crontab.sh* en premier.