



Doc utilisateur

Projet Scripting system

Sommaire

Installation :	3
Pré-requis	4
Déroulement :	5
Exécution automatique :.....	6

Installation

La 1^{ère} étape consiste à installer python :

Téléchargeable ici : <https://www.python.org/downloads/>

Choisir en fonction de votre OS puis l'installer

Vérifiez que python est bien installé dans une invite de commande, par exemple avec :

`python --version`

Si la commande n'est pas reconnue, vérifiez que le chemin vers votre dossier d'installation python se trouve dans le PATH (variable d'environnement)

Placez-vous dans le dossier Scripts (`cd chemin/vers/Scripts`)

Installer les dépendances nécessaires : `pip3 install -r dependances.txt`

Pré-requis :

Dans le fichier **config.ini** :

Si vous venez de télécharger le dossier et que vous n'avez pas de dump SQL (test_export.sql) directement prêt à être comparé, veuillez configurer l'option **[premiere_fois]** à **oui**. Cela permet de sauter l'étape de comparaison vu qu'il n'y a pas de fichier de référence.

L'option passera automatiquement à **non** après l'exécution du script.

La partie **[smb]** sert à configurer la connexion au serveur. **user** et **mdp** sont le nom d'utilisateur et le mot de passe de la machine distante.

adresse_ip est l'adresse ip du serveur.

dossier_partage correspond au nom du dossier partagé par SMB.

nom_pc_local et **nom_pc_serveur** sont respectivement le nom des 2 machines.

Enfin le **port** (par défaut 139) correspond au port relatif au SMB.

Pour **[url_fichier_zip]** il s'agit de l'url ou on va chercher le fichier zip pour le télécharger.

Pour **[mattermost]** il s'agit du webhook permettant de poster des messages sur mattermost.

Pour **[mail]** : **email_source** est l'adresse email qui va envoyer le mail avec **mdp** son mot de passe, **email_destination** étant le destinataire.

Pour **[historisation]** on définit s'il est **actif** ou non, enfin **nbrJours** correspond au nombre de jours maximal acceptable pour les fichiers présents sur le serveur.

Déroulement :

Pour lancer le script : `python ./script.py`

- Chargement du fichier de configuration (et vérification de la présence des différentes sections et options) et on se prépare à écrire dans le fichier log.
- Téléchargement du fichier zip à l'url donné.
- Vérification de la présence du dump SQL dans le zip
- Comparaison du nouveau dump avec le fichier de référence (en fonction de leur date de modification). S'il s'agit de la première fois on passe cette étape.
- Extraction du dump SQL
- Compression du dump en format tar.gz
- Connexion au serveur SMB
- Si l'historisation est active on vérifie les dates des fichiers (dans leur nom) présent sur le serveur, suppression si la date est supérieure à ***nbrJours***.
- Envoi du fichier tar.gz avec un nom sous la forme AnnéeJourMois au serveur
- Envoi du message récapitulatif sur mattermost (via inform.py)
- Envoi du mail récapitulatif avec le fichier de log (recap.log) comme pièce jointe (via inform.py)

Exécution automatique :

Sur Windows :

Nous avons choisi d'utiliser le Planificateur de tâches (Task scheduler) de Windows afin de lancer automatiquement notre script de manière régulière (au choix)

Vous pouvez soit créer un script .bat (batch) qui exécutera notre script python via ce tutoriel :

<https://datatofish.com/python-script-windows-scheduler/>

Ou en utilisant le chemin absolu vers python.exe puis en passant en paramètre notre script comme le montre cette vidéo ou se site :

<https://www.youtube.com/watch?v=lsuAltPOiEw>

<https://www.icchouinard.com/python-automation-using-task-scheduler/>

Dans les deux cas, l'idée est de lancer le planificateur de tâches :

- Créer une nouvelle tâche (ou nouvelle tâche de base en fonction de votre précédent choix)
- Paramétrer la tâche en lui donnant un nom, une fréquence d'exécution et quelle action effectuer lors du déclenchement de la tâche.
- Une fois tout cela terminé la tâche s'effectuera de manière régulière en fonction de vos paramètres.

Sur Linux :

Si toute fois vous préférez utiliser Linux, veuillez suivre ce tutoriel pour pouvoir créer un cron job afin d'automatiser le processus :

<https://gavinwiener.medium.com/how-to-schedule-a-python-script-cron-job-dea6cbf69f4e>