ROCH Maud
MOREAU Romain
BLEIN Louise

Mémoire Technique du Projet d'Archivage

Introduction

Dans le cadre de ce projet, nous avons mis en place un système d'archivage basé sur deux machines virtuelles (VM). La première, une VM Debian, est configurée en tant que serveur Apache pour héberger des fichiers. La seconde, une VM Lubuntu, est utilisée comme serveur SFTP pour l'archivage des données. L'objectif principal est de télécharger un fichier SQL compressé à partir du serveur Apache, de le décompresser, puis de le transférer et l'archiver sur le serveur SFTP. Un processus automatique de nettoyage des fichiers anciens est également mis en place sur le serveur SFTP pour éviter toute surcharge inutile.

De plus, le système envoie un rapport par mail après chaque exécution du script, contenant les logs détaillés des opérations réalisées. Cette fonctionnalité permet le suivi et la traçabilité des processus d'archivage.

Ce document va détailler la méthodologie utilisée, l'organisation des fichiers et des scripts, le fonctionnement du système ainsi que la justification des choix techniques et fonctionnels qui ont guidé notre implémentation.

Organisation des Fichiers et du Code

Le projet est organisé autour de plusieurs fichiers, chacun ayant un rôle bien défini :

- "setup_apache2.sh" : Automatisation de la configuration du serveur Apache.
- “setup_SFTP.sh" : Configuration automatique du serveur SFTP sur la machine Lubuntu.
- "clean_sftp.sh" : Script responsable du nettoyage périodique des fichiers sur le serveur SFTP.
- "Scripting_System.py" : Script Python principal qui orchestre le téléchargement, la décompression, la recompression et le transfert des fichiers vers le serveur SFTP.
- "config.json" : Fichier de configuration contenant les adresses IP des serveurs et les paramètres d'email.
- "README.md" : Documentation qui explique comment installer, configurer et utiliser le projet.

L’utilisation de ces fichiers séparés permet de rendre le système modulaire et facilement maintenable. Chaque fichier a une responsabilité claire, évitant ainsi les redondances et facilitant les mises à jour.

Variables Utilisées et leur Rôle

Variables dans le Script Python

Les variables dans le script Python Scripting_System.py sont chargées depuis un fichier de configuration (config.json) pour simplifier la maintenance et la gestion des adresses IP des serveurs ainsi que des paramètres d'email.

- ip1 : Adresse IP du serveur Apache (VM Debian).
- ip2 : Adresse IP du serveur SFTP (VM Lubuntu).
- email_from : Adresse email utilisée pour envoyer les logs.
- email_to : Adresse email du destinataire des logs.
- smtp_server : Serveur SMTP utilisé pour l'envoi des mails.
- smtp_port : Port du serveur SMTP.
- email_password : Mot de passe pour l’authentification sur le serveur SMTP.

Ces adresses IP et paramètres de messagerie sont utilisés dans les différentes étapes pour se connecter aux serveurs, effectuer les opérations de téléchargement, transfert, et envoyer les rapports par email.

Variables dans le Script de Nettoyage

Dans le script "clean_sftp.sh", nous utilisons une variable pour définir l’ancienneté des fichiers avant leur suppression.

"AGE_LIMIT" : Définit la limite en minutes au-delà de laquelle les fichiers sont supprimés sur le serveur SFTP. Cette valeur est actuellement fixée à 3 minutes, mais peut être ajustée selon les besoins.

Fonctionnement du Système

Le système est basé sur une série de scripts Bash et Python qui assurent la configuration initiale, l'automatisation des tâches et la gestion des fichiers entre les deux serveurs. Voici un aperçu du fonctionnement détaillé.

1. Installation et Configuration Automatisée des Serveurs

a. Serveur Apache (VM Debian)

Le script "setup_apache2.sh" configure automatiquement le serveur Apache. Voici les principales étapes :

- **Mise à jour des paquets** : Le script commence par mettre à jour les paquets pour s'assurer que le système est à jour.
- **Installation d'Apache2** : Il installe le serveur Apache2 pour héberger les fichiers.
- **Configuration du pare-feu** : Il configure le pare-feu pour autoriser les connexions HTTP (port 80) et garantit qu'Apache2 démarrera automatiquement au démarrage du système.
- **Téléchargement du fichier** : Le script copie le fichier "test100.sql.zip" dans le répertoire "/var/www/html", le rendant accessible via une adresse HTTP.

b. Serveur SFTP (VM Lubuntu)

Le script "setup_SFTP.sh" configure le serveur SFTP sur la VM Lubuntu. Les étapes comprennent :

- **Installation d'OpenSSH** : Il installe OpenSSH, qui comprend un serveur SFTP intégré.
- **Création du groupe SFTP** : Un groupe dédié ("sftpusers") est créé pour limiter les accès au serveur SFTP.
- **Sécurisation de l'environnement utilisateur** : Grâce à la directive "ChrootDirectory", chaque utilisateur est confiné dans un environnement isolé, garantissant ainsi une sécurité renforcée.
- **Configuration du répertoire partagé** : Un répertoire de partage est créé pour accueillir les fichiers à archiver.

2. Téléchargement, Décompression et Archivage Automatisé (Script Python)

Le fichier principal du projet, "Scripting_System.py", est chargé d'automatiser le processus d'archivage. Voici les différentes fonctionnalités qu'il implémente :

a. Chargement des Adresses IP

Les adresses IP des serveurs Apache et SFTP sont stockées dans le fichier "config.json", qui est chargé au début du script Python. **Cela permet de centraliser la configuration et d'éviter de modifier le code à chaque changement d'adresse IP.**

b. Téléchargement du Fichier

Le script télécharge le fichier "test100.sql.zip" depuis le serveur Apache à l'aide de la bibliothèque "requests". La requête GET est envoyée à l'adresse IP du serveur Apache et le fichier est enregistré localement.

c. Décompression du Fichier ZIP

Une fois le fichier téléchargé, il est décompressé à l'aide de la bibliothèque "zipfile" de Python. Cette étape extrait le fichier SQL contenu dans l'archive ZIP pour qu'il soit prêt à être transféré.

d. Compression en Format ".tgz"

Le fichier SQL extrait est ensuite re-compressé au format ".tgz" (Tarball compressé avec gzip), qui est plus adapté aux transferts de fichiers sur un réseau, notamment en raison de sa meilleure gestion des fichiers volumineux.

e. Connexion et Transfert vers le Serveur SFTP

Le fichier ".tgz" est transféré vers le serveur SFTP à l'aide de la bibliothèque "pysftp", qui établit une connexion sécurisée à l'aide des identifiants SFTP. Le fichier est stocké dans le répertoire partagé du serveur SFTP, conformément à l'organisation définie dans "setup_SFTP.sh".

3. Nettoyage Automatique des Fichiers Obsolètes

Pour éviter une accumulation de fichiers sur le serveur SFTP, un mécanisme de nettoyage est mis en place via le script "clean_sftp.sh". Ce script est exécuté régulièrement grâce à une tâche "cron", et supprime tous les fichiers plus anciens que 3 minutes. Cette durée est configurable via la variable "AGE_LIMIT".

La tâche "cron" est définie comme suit :

```
*/5 * * * * /home/tse/scriptingsystem/clean_sftp.sh
```

Cela garantit une gestion automatique de l'espace disque et une suppression régulière des fichiers obsolètes.

4. Fonctionnalité d'Envoi de Mail avec les Logs

Une fonctionnalité clé de notre solution est l'envoi automatique d'un email contenant les logs du processus d'archivage. Cela permet à l'administrateur de suivre en temps réel l'état des opérations et de détecter toute erreur éventuelle.

Le script "Scripting_System.py" génère des logs tout au long des étapes : téléchargement, décompression, recompression et transfert via SFTP. Une fois le processus terminé, ces logs sont envoyés par email grâce à la bibliothèque "smtplib".

Voici comment cette fonctionnalité est implémentée :

- **Connexion au serveur SMTP** : Le script se connecte à un serveur SMTP configuré pour envoyer des emails (par exemple, Gmail ou un serveur interne).
- **Création du message** : Le contenu du mail inclut les logs détaillant les différentes étapes du processus.
- **Envoi du mail** : L'email est envoyé à l'adresse de l'administrateur spécifiée dans la configuration du script.

L'envoi de mails est essentiel pour assurer un suivi automatique des opérations, permettant à l'administrateur d'être alerté immédiatement en cas de problème, sans avoir à consulter les logs manuellement sur le serveur. Cela offre un moyen proactif de gestion et améliore la réactivité en cas d'incidents.

Justification des Choix Techniques et Fonctionnels

Choix de SFTP

Nous avons choisi d'utiliser le protocole SFTP pour les raisons suivantes :

- **Sécurité** : SFTP chiffre l'ensemble des données transférées, assurant ainsi la confidentialité des fichiers durant leur transfert. Ceci est particulièrement important lorsque l'on traite des fichiers sensibles.
- **Intégration avec SSH** : Comme SFTP fonctionne sur le protocole SSH, il offre des mécanismes de sécurité supplémentaires, comme l'authentification par clé publique, qui permet un contrôle plus strict des accès.
- **Gestion des utilisateurs et des permissions** : La gestion des utilisateurs via des groupes spécifiques ("sftpusers") permet de restreindre l'accès aux fichiers, tandis que l'utilisation de "ChrootDirectory" isole les utilisateurs dans un répertoire spécifique, renforçant ainsi la sécurité.

Choix de Python

Nous avons opté pour Python comme langage principal pour les raisons suivantes :

- **Simplicité et lisibilité** : Python est connu pour sa simplicité syntaxique, ce qui rend le code plus facile à maintenir et à comprendre.
- **Large gamme de bibliothèques** : Python offre de nombreuses bibliothèques, telles que "requests" pour les téléchargements HTTP et "pysftp" pour les transferts SFTP, ce qui permet d'automatiser le processus avec peu de complexité.
- **Modularité** : Le langage permet une organisation modulaire, avec des étapes distinctes pour le téléchargement, la décompression et le transfert, facilitant ainsi les ajouts ou modifications futures.

Organisation des Scripts Bash

L'utilisation de scripts Bash ("setup_apache2.sh", "setup_SFTP.sh", "clean_sftp.sh") pour la configuration initiale et l'automatisation des tâches système est un choix naturel dans un environnement Linux (afin d'éviter d'utiliser plusieurs commandes à chaque mise en place). Le but étant de simplifier l'utilisation de cette méthode. Ces scripts permettent une installation rapide des services et une gestion efficace des tâches administratives comme le nettoyage des fichiers.

Conclusion

Nous avons mis en place un système d'archivage automatique et sécurisé basé sur deux VM distinctes. Le choix de SFTP assure une sécurité renforcée pour le transfert des fichiers, tandis que l'utilisation de Python permet une gestion automatisée et flexible du processus d'archivage.