Projet PPE

Linux (Debian 13)

AFORP PN2

Projet:	**Infrastructure Réseau & Système Sécurisée Pédagogique de l'AFORP**
Chef de projet:	**Zakaria BOUDHERA**
Classe:	**BTS SIO SISR/BTS CIEL IR (Groupe C)**
Date/Période:	**Janvier 2026**

Sommaire

1 — Introduction générale

Dans la continuité de la sécurisation de l'infrastructure réseau et du déploiement des serveurs Debian, il est impératif d'assurer la pérennité des données. Ce projet porte sur la création et la mise en place d'un outil de **sauvegarde et de restauration automatisé** via un script Bash.

L'objectif est de prévenir la perte de données critiques (configurations systèmes et données utilisateurs) sur le serveur Debian en utilisant des outils natifs robustes comme `rsync`.

2 — Présentation du domaine et du réseau

Ce module a pour but de fournir une solution simple et efficace pour l'administrateur système. Les objectifs principaux sont :

- **Sécuriser les données :** Sauvegarder les répertoires critiques (/home pour les utilisateurs et /etc pour les configurations).
- **Historiser les sauvegardes :** Créer des dossiers horodatés pour conserver plusieurs versions des fichiers.
- **Permettre la restauration :** Offrir une option simple pour remettre en place les données en cas d'incident.
- **Assurer la traçabilité :** Générer des fichiers de logs pour suivre le succès ou l'échec des opérations.

3 — Objectifs techniques du projet

Le projet a été développé et testé dans l'environnement suivant :

- **Système d'exploitation :** Debian (Linux).
- **Langage de script :** Bash (Bourne Again SHell).
- **Outil de transfert :** `rsync` (Remote Sync), choisi pour sa fiabilité et sa capacité à conserver les permissions et les liens symboliques.
- **Editeur de texte :** Nano, utilisé pour l'écriture du script .
- **Utilisateur :** Le script est exécuté dans le répertoire /home/rayanair/sauvegarde.

4 — Analyse du script de sauvegarde

Le cœur du projet repose sur un script interactif nommé svg (ou backup.sh). Voici les variables clés définies au début du script:

- SOURCE_DIRS="/home /etc" : Définit les dossiers sources à sauvegarder. Ici, nous ciblons les données utilisateurs et les fichiers de configuration système.
- BACKUP_ROOT="/home/rayanair/sauvegarde/backup" : L'emplacement racine où seront stockées toutes les archives.
- DATE=$(date ...) : Génère un horodatage précis (Année-Mois-Jour_Heure-Minute-Seconde) pour nommer les dossiers de sauvegarde de manière unique.
- LOG_FILE : Chemin vers le fichier journal pour enregistrer l'activité du script.
- RSYNC_OPTS="-avh" : Options de la commande rsync :
    - -a (archive) : préserve les permissions, dates et groupes.
    - -v (verbose) : affiche les détails de la copie.
    - -h (human-readable) : affiche les tailles de fichiers de manière lisible.

Le script propose un menu interactif demandant à l'utilisateur de choisir entre **"1) Sauvegarde"** et **"2) Restauration"**.

5 — Fonctionnement du module de sauvegarde (choix 1)

Lorsque l'administrateur sélectionne l'option 1, le script exécute les actions suivantes :

1. **Initialisation du Log :** Il écrit la date de démarrage dans le fichier `backup.log`.
2. **Création du répertoire :** Il crée un nouveau dossier spécifique pour cette sauvegarde (ex: backup_2025-10-10_14-30-00) via la commande mkdir -p.
3. **Boucle de sauvegarde :**
    - Le script parcourt la liste SOURCE_DIRS (/home et /etc).
    - Pour chaque dossier, il lance la commande rsync qui copie les fichiers vers le dossier de destination fraîchement créé.
4. **Confirmation :** Un message "Sauvegarde terminée" s'affiche à l'écran et est inscrit dans les logs.

Cette méthode permet d'avoir des sauvegardes "complètes" à un instant T, isolées dans leur propre dossier.

6 — Fonctionnement du module de restauration (choix 2)

L'option 2 permet de récupérer des données en cas de perte ou de corruption:

1. **Affichage des sauvegardes :** Le script liste le contenu du dossier BACKUP_ROOT pour montrer à l'utilisateur les sauvegardes disponibles (classées par date).
2. **Sélection :** L'utilisateur tape le nom du dossier de sauvegarde qu'il souhaite restaurer.
3. **Vérification :** Le script vérifie si le dossier saisi existe réellement. Si ce n'est pas le cas, il affiche "Backup inexistant" et s'arrête.
4. **Restauration :**
    - Il utilise rsync pour copier les données depuis le dossier de sauvegarde vers la racine `/`.
    - Cela permet de remettre en place les configurations /etc et les fichiers /home à leur emplacement d'origine.

7 — Gestion des logs et surveillance

La sécurité d'un système de sauvegarde repose sur sa surveillance. Le script intègre une gestion des logs via la redirection >> "$LOG_FILE" 2>&1.

- **Fichier de log :** /home/rayanair/sauvegarde/backup.log.
- **Contenu :** Chaque exécution note l'heure de début, les erreurs potentielles (grâce à la redirection `2>&1` qui capture aussi les erreurs standards), et l'heure de fin.
- **Utilité :** Cela permet à l'administrateur de vérifier périodiquement si les sauvegardes s'exécutent sans erreur sans avoir à surveiller l'écran pendant le processus.

8 — Tests de bon fonctionnement

Pour valider le script, une série de tests a été effectuée sur le serveur Debian :

1. **Test de création :** Lancement de l'option 1. Vérification avec la commande ls que le dossier backup_DATE a bien été créé dans /home/rayanair/sauvegarde/backup.
2. **Test de contenu :** Vérification que les sous-dossiers home et etc sont bien présents dans la sauvegarde.
3. **Test de restauration :**
    - Suppression d'un fichier test dans le répertoire personnel.
    - Lancement de l'option 2 et sélection de la dernière sauvegarde.
    - Vérification que le fichier supprimé est réapparu.
4. **Test de permissions :** Confirmation que les fichiers restaurés ont conservé leurs propriétaires et droits d'accès initiaux (grâce à l'option `-a` de rsync).
5. **Pour afficher le script :** il faut mettre la commande (4 svg)
6. **Pour faire fonctionner le script :** (chmod +x svg)
7. **Pour lancer le script :** (.\svg)

9 — Répartition des rôles

Comme pour le projet d'infrastructure précédent, ce travail a été réalisé en équipe. Voici la répartition des tâches pour le développement de cet outil de sauvegarde :

- **Adam :** mise en place de l'infrastructure
- **Ibrahim :** Développement de la fonction **"Sauvegarde"** (boucle for, commande rsync) et gestion de la création des répertoires horodatés.
- **Nathael :** Développement de la fonction **"Restauration"** (menu interactif, saisie utilisateur read -p) et intégration finale du code. Mise en place du système de **Logs** (redirection des sorties standards et d'erreurs) et débogage du script
- **Zakaria :** Chef de projet, coordination de l'équipe et réalisation des tests finaux de validation (scénarios de panne et récupération). Conception de l'algorithme global du script et définition des variables d'environnement (SOURCE_DIRS, BACKUP_ROOT).

10 — Conclusion

L'équipe a pu acquérir des compétences en scripting Bash avancé et en administration système Linux. Le script est désormais fonctionnel et prêt à être intégré dans le planificateur de tâches “cron” pour une automatisation totale.