

TP 1 : Automatisation de la création d'utilisateurs sous Linux

Vous venez d'acheter un VPS Linux dans le Cloud pour vos travaux pratiques. Vos camarades vous ont désigné pour créer un compte utilisateur pour chacun. L'objectif est de préparer l'automatisation à grande échelle avant une utilisation sur votre VPS.

Les informations des futurs utilisateurs sont stockées dans un fichier texte `users.txt`, dont chaque ligne suit la structure suivante :

`username;default_password;full_name;phone;email;preferred_shell`

Partie 0 : Procédure de modification du serveur SSH

1. Décrire la procédure correcte pour modifier la configuration du service SSH.
2. Expliquer le principal risque encouru si cette procédure n'est pas respectée.
3. Citer et justifier au moins cinq paramètres de sécurité du serveur SSH que vous devrez modifier pour renforcer la sécurité du système. Chaque paramètre doit être accompagné d'une justification claire.

Partie 1 : Écrire un Script Bash pour automatiser la création des utilisateurs

Écrire un script `create_users.sh` capable de lire le fichier `users.txt` ligne par ligne et respecter les contraintes suivantes :

1. Créer un groupe nommé `students-inf-361`, ce nom doit être passé en paramètre du script.
2. Créer automatiquement chaque utilisateur avec :
 - a. son nom d'utilisateur,
 - b. son nom complet, son numéro de Whatsapp et son adresse mail,
 - c. son shell préféré (le script doit vérifier que le shell sollicité par l'utilisateur existe, sinon l'installer. Si l'installation échoue, réattribuer `/bin/bash`),
 - d. son répertoire personnel.
3. Ajouter tous les comptes au groupe : `students-inf-361`.
4. Configurer son mot de passe par défaut, le mot de passe fourni doit être haché (SHA-512).
5. Forcer le changement du mot de passe à la première connexion.
6. Ajouter chaque utilisateur au groupe `sudo`. Toutefois, il faut empêcher l'usage de la commande `su` (switch user) aux membres du groupe `students-inf-361`.
7. Un message de bienvenue personnalisé doit s'afficher automatiquement à chaque connexion d'un utilisateur. Ajoutez un fichier `~/WELCOME.txt` que vous afficherez dans `~/bashrc`.
8. Configurer la limite d'espace disque pour qu'un utilisateur ne puisse pas stocker plus de 15 Go dans son répertoire personnel.
9. Limiter l'utilisation mémoire pour empêcher qu'un utilisateur de lancer un processus consommant plus de 20 % de la RAM.

10. Créer un fichier qui garde les traces d'exécution du script, la date et l'heure d'exécution du script doivent être y figurer, ainsi que chaque opération réalisée par le script.

Partie 2 : Écrire un playbook Ansible pour automatiser la création des utilisateurs

Vous devez maintenant reproduire exactement les mêmes opérations que dans le script Bash, mais cette fois au moyen d'un playbook Ansible.

Votre playbook Ansible doit également envoyer un email personnalisé à chaque utilisateur immédiatement après la création de son compte. L'email doit contenir une formule de politesse et les informations suivantes :

- l'adresse IP du serveur,
- le port d'écoute du serveur SSH,
- le nom d'utilisateur,
- le mot de passe initial,
- la commande SSH permettant de se connecter au serveur,
- la commande permettant de transmettre sa clé publique SSH, quel que soit son OS.

Partie 3 : Exécuter votre script en utilisant Terraform

Dans cette partie, vous devez utiliser Terraform pour exécuter votre script de création d'utilisateurs.

Livrables attendus (Chaque partie doit être dans un répertoire séparé et avoir son fichier [README.md](#)). **Le tout sur votre Github.**

1. Le fichier script create_users.sh
2. Le fichier playbook Ansible [create_users.yml](#)
3. Le fichier d'inventaire Ansible
4. Le fichier source des utilisateurs [users.txt](#) (ou .csv/.yaml selon l'outil)
5. Le dossier Terraform incluant au minimum : [main.tf](#) et [variables.tf](#)
6. Une courte documentation globale ([README.md](#))

Compétences visées

1. Développement de scripts Bash robustes
2. Création et gestion des utilisateurs et groupes Linux
3. Gestion des permissions et restrictions d'accès
4. Durcissement et configuration du service SSH
5. Gestion des ressources du système
6. Personnalisation de l'environnement utilisateur
7. Industrialisation de la configuration système avec Ansible, envoi automatique d'emails via Ansible
8. Infrastructure-as-Code avec Terraform, intégration Terraform / Bash
9. Documentation technique et organisation du projet