

## Exercice 1 – Sauvegarde automatique et rotation

**Objectif :** Créer un script Bash qui effectue la sauvegarde de certains dossiers du système et supprime les anciennes sauvegardes.

### Consignes :

- Le script doit sauvegarder les dossiers /etc et /var/www dans un fichier compressé nommé backup-YYYYMMDD.tar.gz, enregistré dans /opt/backups/.
- Il doit conserver uniquement les 7 dernières sauvegardes et supprimer les plus anciennes.
- Il doit enregistrer ses actions dans un fichier de log /var/log/backup.log.
- Planifier l'exécution du script tous les jours à 2h00 avec cron.

### Livrables :

- Script backup\_rotate.sh
- Ligne cron correspondante
- Extrait du fichier de log

## Exercice 2 – Vérification et redémarrage automatique d'un service

**Objectif :** Créer un script qui vérifie régulièrement si un service fonctionne et le redémarre si nécessaire.

### Consignes :

- Écrire un script healthcheck\_nginx.sh qui vérifie si le service nginx est actif.
- Si le service est arrêté, il doit le redémarrer et écrire un message dans /var/log/healthcheck\_nginx.log.
- Si le service est actif, il écrit simplement 'nginx OK' dans le log.
- Planifier l'exécution du script toutes les 2 minutes avec cron.

### Livrables :

- Script healthcheck\_nginx.sh
- Ligne cron
- Extrait du log

## Exercice 3 – Déploiement automatique d'un site web

**Objectif :** Mettre en place un script qui déploie une nouvelle version d'un site statique et fait un retour arrière en cas d'erreur.

### Consignes :

- Les fichiers du site sont fournis sous forme d'archives /var/www/site-.tgz.

- Écrire un script `deploy_site.sh` qui prend en argument la version à déployer.
- Mettre à jour le lien symbolique `/var/www/site` vers la nouvelle version et recharger `nginx`.
- Vérifier le bon fonctionnement du site avec `curl` et faire un `rollback` si le test échoue.
- Planifier le déploiement tous les jours à 3h05.

#### **Livrables :**

- Script `deploy_site.sh`
- Exemple de log de déploiement et `rollback`

## **Exercice 4 – Audit automatique de la sécurité du système**

**Objectif :** Vérifier automatiquement quelques règles de sécurité et produire un score.

#### **Consignes :**

- Écrire un script `hardening_audit.sh` qui vérifie les paramètres de `/etc/ssh/sshd_config` : `PasswordAuthentication no` et `PermitRootLogin no`.
- Vérifier que le pare-feu (`ufw` ou `iptables`) est actif.
- Calculer un score sur 100 et générer un rapport dans `/var/reports/hardening-YYYY-MM-DD.md`.
- Planifier l'exécution chaque lundi à 7h00.

#### **Livrables :**

- Script `hardening_audit.sh`
- Fichier de rapport généré
- Extrait de log