Procédure de mise en place d'un script d'automatisation de maintenance système

Table des matières / Sommaire

1. Cahier des charges – Expression des besoins

Descriptif de l'existant

Le serveur Debian 13 accumule des fichiers temporaires, des caches de paquets d'installation (apt) et des journaux anciens au fil du temps. Sans intervention manuelle, l'espace disque disponible diminue, ce qui peut impacter la stabilité des services comme OpenVPN ou Redmine.

Besoin(s)

- Automatiser le nettoyage des résidus d'installations de paquets.
- Vider les répertoires temporaires inutilement encombrés.
- Planifier ces actions pour qu'elles s'exécutent de nuit sans intervention humaine.

Contrainte(s)

Type	**Description**
Technique	Utilisation du langage de script Bash.
Planification	Recours à l'utilitaire **Cron** de Debian.

2. Ressources

Ressources mises à disposition

- **Serveur :** Une machine virtuelle avec **Debian 13**.
- **Environnement :** Le shell **Bash**, qui est l'interpréteur de commandes par défaut sous Linux.
- **Privilèges :** Un accès avec les droits sudo pour pouvoir nettoyer les dossiers système.

Ressources dont vous avez besoin pour la réalisation

- **Matériel d'administration :** Un PC pour coder et tester le script.
- **Éditeur de texte :** L'outil **Nano** pour rédiger le script **.sh**.
- **Planificateur :** L'utilitaire système **Cron**, déjà présent sur Debian, pour gérer l'automatisation.

Façon dont vous allez gérer ses ressources

On va centraliser les commandes de nettoyage dans un seul fichier script situé dans ***/usr/local/bin/***. Ce script sera ensuite appelé par la table de planification du système (**crontab**). Cette méthode permet de garder un système propre de manière totalement transparente, sans consommer de ressources processeur supplémentaires durant la journée.

3. Analyse

Descriptifs des solutions

Pour maintenir un système Debian propre, deux approches sont possibles :

- **Maintenance manuelle :** L'administrateur se connecte régulièrement pour taper les commandes de nettoyage (***apt clean***, suppression des logs, etc.).
- **Maintenance automatisée (Script + Cron) :** Création d'un script Bash regroupant toutes les commandes, exécuté automatiquement à intervalles réguliers par le système.

Comparaison des solutions

Critère	Maintenance manuelle	Maintenance automatisée
Fiabilité	Risque d'oubli élevé	Exécution garantie à 100%
Temps passé	Chronophage sur le long terme	Temps initial de setup uniquement
Risque	Erreur de manipulation humaine	Comportement prévisible et stable

Choix d'une solution

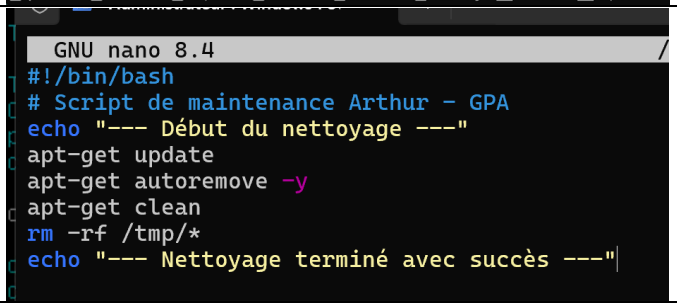
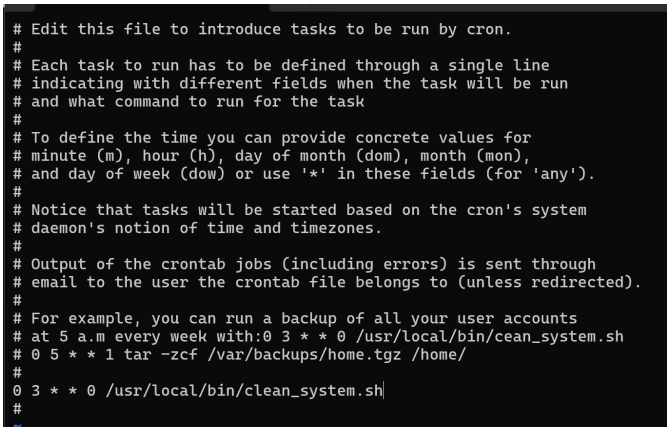
Le choix s'est porté sur la **maintenance automatisée**. Un serveur de production (hébergeant ton VPN ou Redmine) ne doit pas tomber en panne à cause d'un disque dur saturé. L'automatisation via un script Bash couplé à une tâche Cronpermet de garantir que le système restera performant sans aucune intervention humaine.

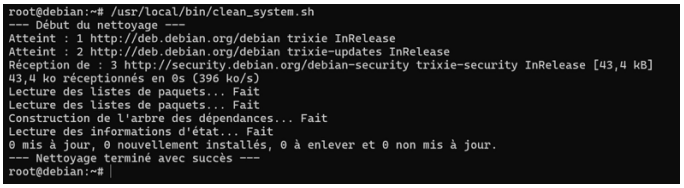
Impact sur le SI :

- **Disponibilité :** L'exécution du script est programmée la nuit (3h du matin) pour éviter tout ralentissement des services pendant les heures de production.
- **Intégrité :** Le nettoyage des fichiers /tmp et des caches de paquets n'affecte pas les données utilisateurs mais prévient les erreurs système liées à un disque plein.
- **Sécurité :** L'automatisation réduit les interventions humaines en root, limitant ainsi les risques de mauvaises manipulations manuelles.

4. Mise en place

Réalisation en suivant le phasage énoncé précédemment

Étape	Description	Images
1	**Création du script** : Ouverture d'un nouveau fichier script dans le répertoire des exécutables locaux. **sudo nano /usr/local/bin/clean_system.sh**	
2	**Rédaction du code** : Écriture des commandes de nettoyage (**apt clean, apt autoremove, nettoyage /tmp**). *(Voir le code ci-dessous)*	
3	**Permissions :** Attribution des droits d'exécution pour que le système puisse lancer le script. **sudo chmod +x /usr/local/bin/clean_system.sh**	
4	**Planification (Cron)** : Ouverture de la table de planification pour programmer le script tous les dimanches à 3h. sudo crontab -e	

5	**Test manuel :** Exécution du script à la main pour vérifier qu'il n'y a pas d'erreur de syntaxe. sudo /usr/local/bin/clean_system.sh	

Rapport de tests

Test de conformité	Commande de contrôle	Résultat attendu	Résultat obtenu
Exécutabilité	ls -l /usr/local/bin/clean_system.sh	Droits -rwxr-xr-x	**OK**
Syntaxe Bash	/usr/local/bin/clean_system.sh	Message "Nettoyage terminé"	**OK**
Planification	crontab -l	Ligne de programmation présente	**OK**

5. Bilan

Conclusion

La mise en place de ce script d'automatisation transforme une gestion manuelle fastidieuse en un processus système autonome et fiable. En garantissant un nettoyage hebdomadaire des caches et des fichiers temporaires, on assure la stabilité du serveur Debian 13 sur le long terme. Cette approche préventive est une règle d'or en administration système pour éviter les pannes critiques liées à la saturation des disques.

Auto-évaluation sur la qualité du travail réalisé

Le script Bash est fonctionnel et la planification via Cron a été testée avec succès. L'utilisation d'un script centralisé permet de faire évoluer facilement les tâches de maintenance (ajout de sauvegardes ou de vérifications de logs) sans modifier la planification. Le travail est conforme aux standards professionnels de "Hardening" et de maintenance préventive.

Compétence SISR mobilisée	Description du travail réalisé
Gérer le patrimoine informatique	Automatisation des tâches d'administration pour optimiser le temps de gestion et la pérennité du serveur.
Mettre en œuvre la sécurité du SI	Prévention des risques de déni de service (DoS) local par saturation de l'espace de stockage.