Installez un serveur Debian 12 de façon sécurisée

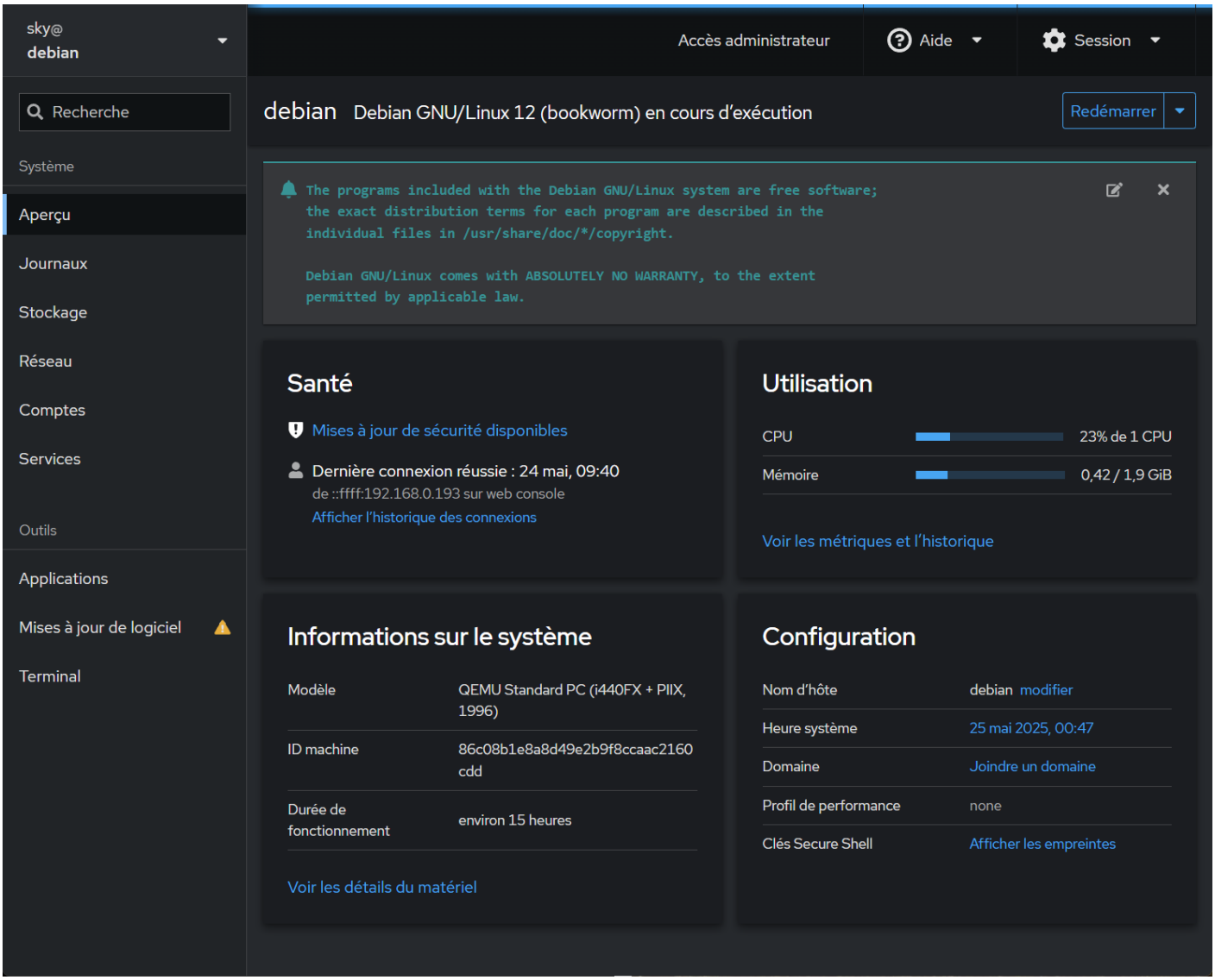
Nous allons installer Debian 12 de manière simple et sécurisée avec un mot de passe à usage unique (OTP) pour le serveur SSH et Cockpit, un pare-feu pour bloquer les IP malveillantes avec CrowdSec, et des mises à jour automatiques.

Prérequis :

- Debian 12 avec le compte root désactivé déjà installé
- Nom d'utilisateur qui ne contient pas de nom générique comme `Debian, Serveur, etc.`
- Compte avec un mot de passe fort
- Serveur SSH
- Installez l'application Android Google Authenticator ou un équivalent sur votre téléphone

Installation de Cockpit avec un mot de passe à usage unique (OTP)

Cockpit est une interface web de gestion de serveurs Linux. Il permet aux administrateurs de surveiller et de gérer facilement leurs serveurs via un navigateur web, offrant une vue d'ensemble des performances du système, des journaux, des utilisateurs, et des services en cours d'exécution. Cockpit simplifie la gestion des tâches administratives courantes sans nécessiter de compétences avancées en ligne de commande.



L'OTP (One-Time Password) est un mot de passe à usage unique qui change à chaque connexion, renforçant la sécurité en rendant difficile l'accès non autorisé. L'OTP est souvent utilisé dans le cadre de l'authentification à deux facteurs (2FA), où un code généré par une application comme Google Authenticator est requis en plus du mot de passe traditionnel. Cela ajoute une couche supplémentaire de sécurité en vérifiant non seulement ce que vous savez (le mot de passe), mais aussi ce que vous avez (le code OTP).

Installation de Cockpit et de libpam-google-authenticator :

```
sudo apt install -y cockpit libpam-google-authenticator
```

Flashez le QR code avec Google Authenticator après avoir exécuté cette commande :

```
google-authenticator -t -f -d -w 3 -e 10 -r 3 -R 30
```

Copiez vos paramètres 2FA dans un endroit sûr :

```
cat .google_authenticator
```

Explication des options :

- **-t** : Activer la vérification TOTP (Time-based One-Time Password).
- **-f** : Écrire la configuration dans le fichier `~/.google_authenticator`.
- **-d** : Interdire la réutilisation des jetons précédemment utilisés.
- **-w 3** : Définir la taille de la fenêtre des jetons autorisés. Par défaut, les jetons expirent toutes les 30 secondes. Une fenêtre de taille 3 permet l'authentification avec le jeton précédent et le jeton suivant pour compenser un éventuel décalage horaire.
- **-e 10** : Générer 10 codes de secours d'urgence.
- **-r 3 -R 30** : Limiter le taux de connexion. Autoriser 3 tentatives de connexion toutes les 30 secondes.

Ajoutez à la fin du fichier `/etc/pam.d/cockpit`. L'option `nullok` permet aux utilisateurs qui n'ont pas encore généré de code 2FA de se connecter, tandis que les codes sont requis si l'utilisateur a suivi l'étape 2 ci-dessus. Cette option est utile lors du déploiement. Une fois que tous les utilisateurs auront généré des codes, vous pourrez supprimer l'option `nullok` pour exiger la 2FA pour tous. :

```
echo 'auth required pam_google_authenticator.so nullok' | sudo tee -a /etc/pam.d/cockpit
```

Redémarrez le service cockpit :

```
sudo systemctl restart cockpit
```

Configurez SSH pour utiliser l'OTP

Modifiez le fichier de configuration PAM SSH :

```
sudo nano /etc/pam.d/sshd
```

Ajoutez la ligne suivante à la fin du fichier. L'option `nullok` permet aux utilisateurs qui n'ont pas encore généré de code 2FA de se connecter, tandis que les codes sont requis si l'utilisateur a suivi l'étape 2 ci-dessus. Cette option est utile lors du déploiement. Une fois que tous les utilisateurs auront généré des codes, vous pourrez supprimer l'option `nullok` pour exiger la 2FA pour tous.

```
auth required pam_google_authenticator.so nullok
```

Enregistrez et fermez le fichier.

Modifiez le fichier de configuration du démon SSH :

```
sudo nano /etc/ssh/sshd_config
```

Vérifiez que les options suivantes sont définies comme indiqué, ou ajoutez-les si elles n'existent pas :

```
KbdInteractiveAuthentication yes
ChallengeResponseAuthentication yes
X11Forwarding no
UsePAM yes
```

Enregistrez et fermez le fichier.

3. Redémarrez le service SSH.

```
sudo systemctl restart ssh
```

Accès

Accédez à la page d'administration via le navigateur web :

```
adresseipduserveur:9090
```

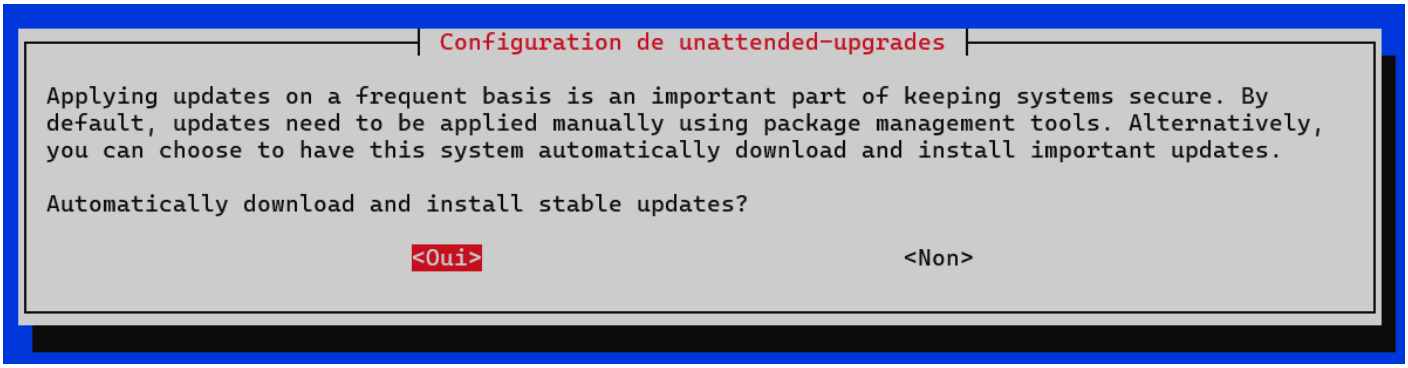
Désactivez le serveur SSH dans les services lorsque vous n'en avez pas besoin.

Mise à jour des paquets de sécurité

Activez les mises à jour automatiques à l'aide de la commande suivante, qui vous demandera si vous souhaitez activer les mises à jour automatiques. Sélectionnez Oui et appuyez sur Entrée, ce qui confirmera que le service unattended-upgrades est actif et prêt à gérer les mises à jour pour vous.

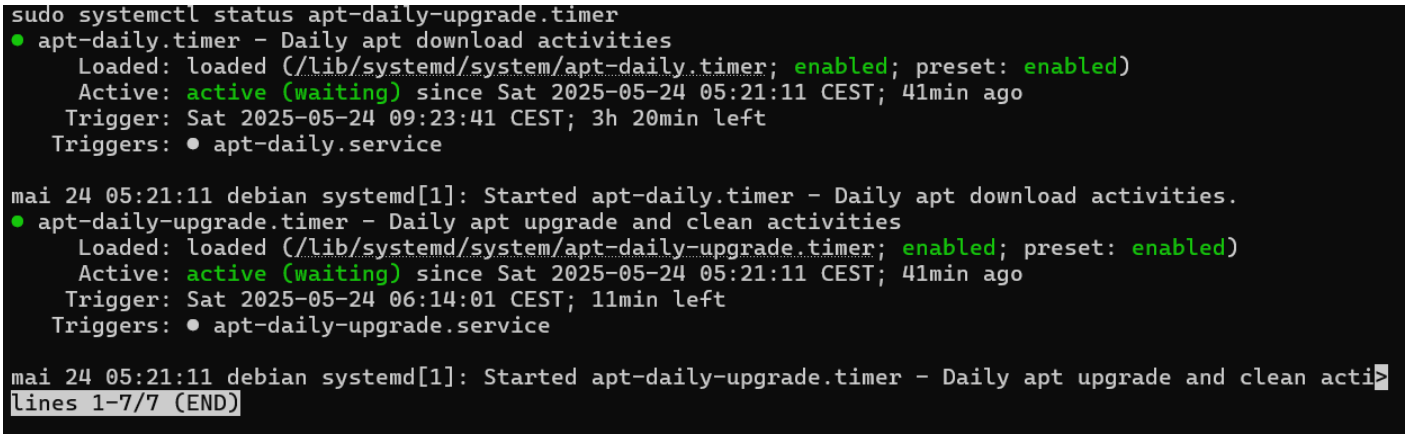
```
sudo apt install -y unattended-upgrades
```

```
sudo dpkg-reconfigure unattended-upgrades
```



Par défaut, unattended-upgrades fonctionne tous les jours, pour vérifier ou modifier l'horaire, vérifiez la minuterie systemd :

```
sudo systemctl status apt-daily.timer
sudo systemctl status apt-daily-upgrade.timer
```



Pour vous assurer que tout fonctionne, simulez une mise à niveau sans surveillance. Si vous venez de faire une nouvelle installation ou si vous aviez déjà mis à jour récemment le système, vous ne devriez pas avoir de réponse :


```
sudo unattended-upgrade --dry-run
```

Activez les redémarrages automatiques après les mises à jour du noyau en ajoutant cette ligne :

```
echo 'Unattended-Upgrade::Automatic-Reboot "true";' | sudo tee -a
/etc/apt/apt.conf.d/50unattended-upgrades
```

Vous pouvez également planifier des redémarrages à un moment précis :

```
echo 'Unattended-Upgrade::Automatic-Reboot-Time "02:00";' | sudo tee -a
/etc/apt/apt.conf.d/50unattended-upgrades
```

Vous pouvez surveiller les mises à jour automatiques en vérifiant les journaux :

```
sudo less /var/log/unattended-upgrades/unattended-upgrades.log
```

Planification des migrations vers Debian 13 et versions suivantes

Il est important de noter dans votre agenda la future migration vers Debian 13, ainsi que vers les versions suivantes, afin de toujours bénéficier des mises à jour de sécurité officielles. Debian maintient un cycle de support défini pour chaque version stable, généralement d'environ 5 ans, comprenant le support principal (security updates) et le support LTS (Long Term Support). Passé ces périodes, aucune mise à jour de sécurité n'est fournie, ce qui expose votre système à des risques importants.

Version Debian	Date de sortie	Fin du support standard	Fin du support LTS
Debian 11 (Bullseye)	14/08/2021	30/06/2024	30/06/2026
Debian 12 (Bookworm)	10/06/2023	10/06/2026	-
Debian 13 (Trixie) (prévue)	Mi-2026 (prévision)	Mi-2029 (prévision)	-

Conseils :

- Planifiez vos mises à jour majeures avant la fin du support standard pour éviter toute interruption de sécurité.
- Mettez un rappel périodique dans votre agenda pour suivre les annonces de Debian et les dates officielles.
- Faites systématiquement des sauvegardes complètes avant toute migration pour pouvoir revenir en arrière en cas de problème.
- Testez soigneusement vos systèmes avant la migration pour garantir la compatibilité des services critiques.

Installation du par-feu

Installez `iptables` :

```
sudo apt install iptables
```

Configuration des règles `iptables`

1. **Ouvrez les ports 22 et 9090 :**

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 9090 -j ACCEPT
```

2. **Enregistrez les règles pour qu'elles soient persistantes après un redémarrage :**

```
sudo apt install iptables-persistent
sudo netfilter-persistent save
```

Pour vérifier que les règles ont été appliquées correctement, utilisez la commande suivante :

```
sudo iptables -L -v
```

Pour ouvrir un port spécifique (par exemple, le port 8080) :

```
sudo iptables -A INPUT -p tcp --dport 8080 -j ACCEPT
sudo netfilter-persistent save
```

Pour fermer un port spécifique (par exemple, le port 8080) :

```
sudo iptables -D INPUT -p tcp --dport 8080 -j ACCEPT
sudo netfilter-persistent save
```

Explication des commandes

- `iptables -A INPUT -p tcp --dport <port> -j ACCEPT` : Ajoute une règle pour accepter les connexions entrantes sur le port spécifié.
- `iptables -D INPUT -p tcp --dport <port> -j ACCEPT` : Supprime la règle pour accepter les connexions entrantes sur le port spécifié.
- `netfilter-persistent save` : Enregistre les règles actuelles pour qu'elles soient appliquées automatiquement au démarrage.

Installation de CrowdSec pour bloquer les IP malveillantes

Prérequis :

- iptables

CrowdSec est une solution open-source de sécurité qui protège les serveurs et les applications contre les attaques malveillantes en analysant les journaux et en détectant les comportements

suspects. En utilisant des règles de détection et des listes de réputation, CrowdSec peut bloquer automatiquement les adresses IP malveillantes, partager ces informations avec une communauté mondiale, et ainsi améliorer collectivement la sécurité de tous les utilisateurs. Il s'intègre facilement avec divers services et plateformes, offrant une protection proactive et collaborative contre les cybermenaces.

```
sudo apt install -y curl && curl -s
https://packagecloud.io/install/repositories/crowdsec/crowdsec/script.deb.sh | sudo bash
```

```
sudo apt install -y crowdsec crowdsec-firewall-bouncer-iptables
```

Description des paquets :

- **crowdsec** : CrowdSec est une solution de sécurité open-source qui aide à détecter et à bloquer les comportements malveillants en temps réel. Elle fonctionne en analysant les journaux (logs) pour identifier les comportements suspects et en prenant des mesures appropriées.
- **crowdsec-firewall-bouncer-iptables** : Ce paquet est un "bouncer" pour CrowdSec qui utilise iptables (une interface pour Netfilter, le pare-feu Linux) pour bloquer les adresses IP identifiées comme malveillantes par CrowdSec. Un "bouncer" est un composant qui applique les règles de blocage définies par CrowdSec.

```
sudo systemctl reload crowdsec
```

Redémarrez le système :

```
sudo reboot
```

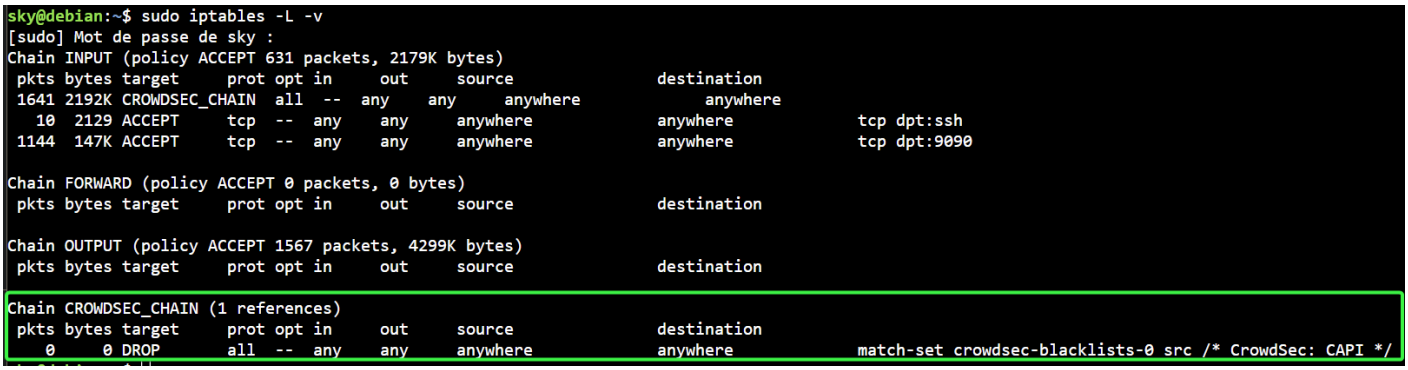
Vérifiez si le service et les règles de par-feu est lancez et fonctionne correctement :

```
sudo systemctl status crowdsec
```

```
sky@debian:~$ sudo systemctl status crowdsec
● crowdsec.service - Crowdsec agent
     Loaded: loaded (/lib/systemd/system/crowdsec.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-05-24 09:22:58 CEST; 2min 55s ago
   Main PID: 3043 (crowdsec)
      Tasks: 7 (limit: 2311)
     Memory: 43.5M
        CPU: 1.526s
     CGroup: /system.slice/crowdsec.service
             ├─3043 /usr/bin/crowdsec -c /etc/crowdsec/config.yaml
             └─3049 journalctl --follow -n 0 _SYSTEMD_UNIT=ssh.service

mai 24 09:22:54 debian systemd[1]: Starting crowdsec.service - Crowdsec agent...
mai 24 09:22:58 debian systemd[1]: Started crowdsec.service - Crowdsec agent.
```

```
sudo iptables -L -v
```



Documentation complémentaire

- Cockpit
- CrowdSec

Révision #13
Créé 2025-05-24 05:27:32 CEST par Philippe Favre
Mis à jour 2025-05-29 00:35:08 CEST par Philippe Favre