Installer et sécuriser un serveur Debian 13 « Trixie »

Nov 27, 2026 · @Serge Martins

Objectif

Installer un **serveur Debian 13 minimal** (sans interface graphique) qui servira de modèle à toutes les VM Linux du domaine **martinscie.lan** (PARC01, MAIL01, OCS01…), puis appliquer une configuration de base sûre : réseau fixe, sources APT, compte d'administration, SSH par clé, pare-feu, mises à jour automatiques et synchronisation de l'heure.

Élément	Valeur
Nom	deb-srv01.martinscie.lan
Adresse IP	192.168.10.50/24 (VLAN 10 Serveurs)
Passerelle / DNS	192.168.10.254 (RFW01) / 192.168.10.10 (DC01)
Image	debian-13.x-amd64-netinst.iso (dernière version de point)
Ressources	1 vCPU, 2 Go RAM, 20 Go disque

Un modèle Debian 13 sûr pour tous les serveurs Linux



DEB-SRV01 · serveur modèle Debian 13

L'administrateur se connecte uniquement par clé SSH ; le serveur prend son DNS et son heure sur DC01 et ses correctifs sur les dépôts de sécurité Debian.

Étape 1 – Installation

1. Démarrer sur l'ISO netinst → **Install** (mode texte).
2. Langue Français, pays France, clavier Français.
3. Nom de machine `deb-srv01`, domaine `martinscie.lan`.
4. Mot de passe root : **laisser vide** → le premier utilisateur (`admsys`) reçoit les droits `sudo`.
5. Partitionnement : *Assisté – utiliser un disque entier avec LVM*, *Partitions /home, /var et /tmp séparées*.
6. Miroir : `deb.debian.org`.
7. Sélection des logiciels : décocher les environnements de bureau, cocher uniquement **serveur SSH** et **utilitaires usuels du système**.
8. Installer GRUB sur le disque, redémarrer.

Étape 2 – Réseau en adresse fixe

`/etc/network/interfaces` (adapter le nom de l'interface vu avec `ip a`) :

```
auto ens18
iface ens18 inet static
    address 192.168.10.50/24
    gateway 192.168.10.254
    dns-nameservers 192.168.10.10
    dns-search martinscie.lan
```

```
sudo apt install -y resolvconf
sudo systemctl restart networking
ping -c2 dc01.martinscie.lan
```

Sur DC01 : `Add-DnsServerResourceRecordA -ZoneName "martinscie.lan" -Name "deb-srv01" -IPv4Address 192.168.10.50 -CreatePtr`.

Étape 3 – Sources APT et mises à jour

Debian 13 utilise le format **deb822** : `/etc/apt/sources.list.d/debian.sources`.

```
Types: deb
URIs: http://deb.debian.org/debian
Suites: trixie trixie-updates
Components: main contrib non-free non-free-firmware
Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg

Types: deb
URIs: http://security.debian.org/debian-security
Suites: trixie-security
Components: main contrib non-free non-free-firmware
Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg
```

```
sudo apt update && sudo apt full-upgrade -y
sudo apt install -y vim curl htop qemu-guest-agent unattended-upgrades apt-
listchanges
sudo dpkg-reconfigure -plow unattended-upgrades     # activer les mises à
jour de sécurité automatiques
```

Étape 4 – SSH par clé

Depuis le poste d'administration :

```
ssh-keygen -t ed25519
ssh-copy-id admsys@deb-srv01.martinscie.lan
```

Sur le serveur, créer `/etc/ssh/sshd_config.d/10-durcissement.conf` :

```
PermitRootLogin no
PasswordAuthentication no
KbdInteractiveAuthentication no
AllowGroups sudo
MaxAuthTries 3
```

```
sudo sshd -t && sudo systemctl reload ssh
```

Étape 5 – Pare-feu nftables

```
sudo apt install -y nftables
sudo tee /etc/nftables.conf >/dev/null <<'EOF'
#!/usr/sbin/nft -f
flush ruleset
table inet filtre {
  chain entree {
    type filter hook input priority 0; policy drop;
    ct state established,related accept
    iif lo accept
    ct state invalid drop
    ip protocol icmp accept
    ip saddr { 192.168.10.0/24, 192.168.20.0/24 } tcp dport 22 accept
  }
  chain transfert { type filter hook forward priority 0; policy drop; }
  chain sortie { type filter hook output priority 0; policy accept; }
}
EOF
sudo systemctl enable --now nftables
sudo nft list ruleset
```

Ajouter ensuite les ports du service hébergé (80/443 pour un serveur web, 25/587/993 pour la messagerie…).

Étape 6 – Heure, journaux et protection SSH

```
sudo apt install -y chrony fail2ban
echo "server 192.168.10.10 iburst" | sudo tee
/etc/chrony/sources.d/dc01.sources
sudo systemctl restart chrony && chronyc sources
sudo systemctl enable --now fail2ban
sudo fail2ban-client status sshd
journalctl -p warning -b      # erreurs depuis le démarrage
```

Étape 7 – En faire un modèle

Avant de convertir la VM en modèle : `sudo apt clean`, vider `/etc/machine-id` (`sudo truncate -s0 /etc/machine-id`), supprimer les clés d'hôte SSH (`sudo rm /etc/ssh/ssh_host_*`, à régénérer au premier démarrage avec `sudo dpkg-reconfigure openssh-server`), puis éteindre. Chaque clone recevra son nom et son IP.

Vérifications

Contrôle	Commande
Version	`cat /etc/debian_version`
Noyau	`uname -r` (6.12 LTS)
Mises à jour en attente	`apt list --upgradable`
Ports ouverts	`ss -tlnp`
Services en échec	`systemctl --failed`

Dépannage

Symptôme	Piste
Pas de réseau après redémarrage	Nom d'interface erroné dans `/etc/network/interfaces`
SSH refusé après durcissement	Clé non copiée ou utilisateur hors du groupe `sudo` : console de la VM
`apt update` en erreur	Fichier deb822 mal formé (ligne vide entre les blocs)
Clones avec la même adresse DHCP	`/etc/machine-id` non réinitialisé