Mise à jour Proxmox VE 9 : guide complet de l'upgrade

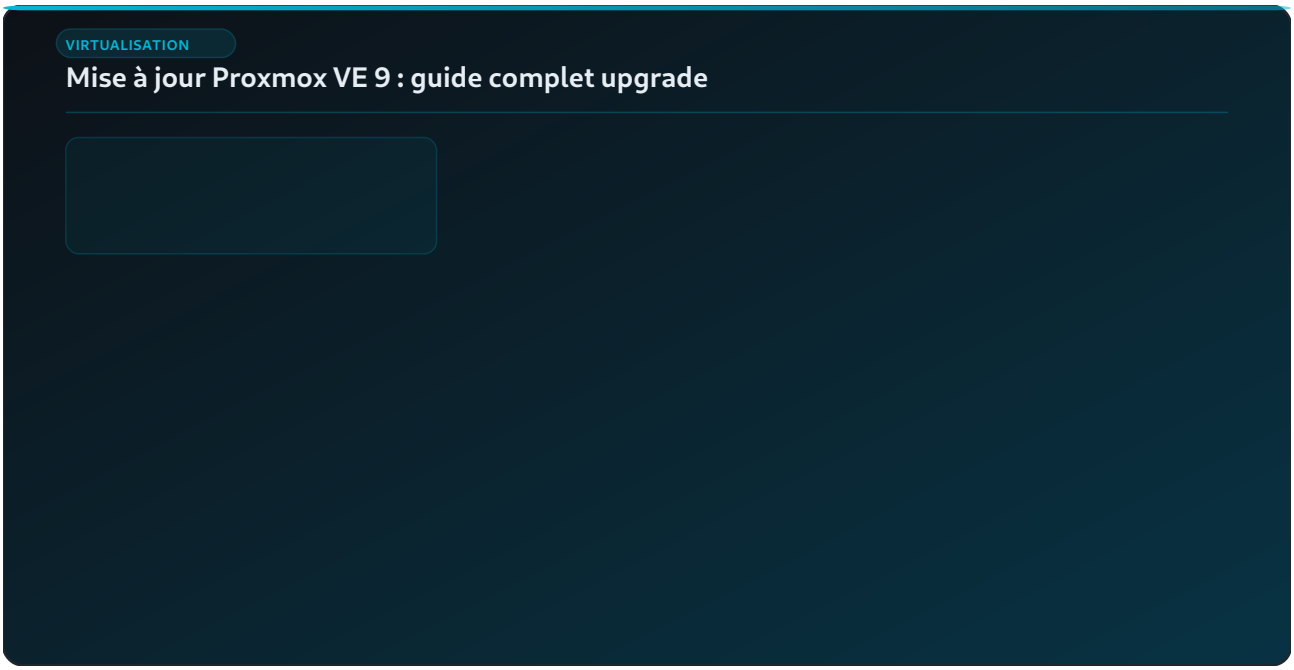
> Guide complet de mise à jour Proxmox VE 9 : dépôts apt enterprise/no-subscription, apt full-upgrade, rolling upgrade cluster, vérifications post-reboot et...

La mise à jour de Proxmox VE 9 est une opération structurante qui conditionne la sécurité, la stabilité et la compatibilité de votre infrastructure de virtualisation. Basé sur Debian 13 « Trixie », Proxmox VE 9 embarque le kernel Linux 6.14, QEMU 10.x, LXC 6.x, OpenZFS 2.3 et Ceph Squid 19.2. Maintenir chaque nœud à jour implique de maîtriser la gestion des dépôts apt (enterprise, no-subscription, pvetest), la procédure de `apt full-upgrade`, les vérifications pré et post-reboot, ainsi que la stratégie de rolling upgrade sur un cluster en production. Ce guide couvre l'ensemble du processus, des sauvegardes préalables à la validation finale, avec toutes les commandes et les pièges à éviter pour une mise à jour sans interruption de service ni perte de données. Que vous soyez sur un nœud isolé ou un cluster multi-nœuds, vous trouverez ici une méthode éprouvée et des tableaux de référence pour chaque phase de l'opération.

📋 EN BREF

- Proxmox VE 9 repose sur Debian 13 Trixie : kernel 6.14, QEMU 10.x, OpenZFS 2.3
- Trois dépôts apt : enterprise (payant), no-subscription (gratuit production), pvetest (hors production)
- Toujours utiliser `apt full-upgrade`, jamais `apt upgrade` seul
- Sur un cluster : rolling upgrade nœud par nœud, migration HA préalable obligatoire
- Snapshot ZFS du root avant upgrade = rollback rapide en cas de régression

- Checklist post-reboot : pveversion, uname -r, pvecm status, zpool status, pvesr status



Proxmox VE 9, sorti à l'été 2025, suit deux rythmes de maintenance distincts : des mises à jour fréquentes (corrections de sécurité, bugfixes ZFS/QEMU, micro-évolutions) livrées via `apt`, et des versions ponctuelles 9.x (9.1, 9.2…) introduisant des évolutions plus structurantes. Proxmox distribue ses paquets via trois dépôts : **pve-enterprise** (souscription payante, qualité validée), **pve-no-subscription** (gratuit, qualité production sans engagement) et **pvetest** (préversions à éviter en production). Sur un nœud sans souscription, le dépôt enterprise doit être désactivé sous peine d'erreurs `apt 401 Unauthorized`.

Avant d'initier toute mise à jour, une vérification de l'état initial du nœud s'impose. Cette étape évite de lancer une procédure sur un système déjà dégradé.

```
 # Vérification rapide de l'état initial
pveversion -v              # Détail complet des paquets Proxmox installés
uname -r                   # Kernel actuellement chargé
pvecm status               # État cluster : quorum, nœuds, version corosync
zpool status               # Santé des pools ZFS
df -h /                    # Espace libre sur la racine (min. 2-3 GiB requis)
systemctl --failed         # Services en échec avant toute opération
journalctl -p err -b       # Erreurs kernel/systemd depuis le boot
apt list --upgradable 2>/dev/null  # Paquets en attente
```

Pour aller plus loin sur la supervision de votre infrastructure Proxmox, consultez notre guide Proxmox VE : hyperviseur KVM et LXC open source.

2. Préparation avant la mise à jour : sauvegardes et points de retour

Une mise à jour Proxmox bien conduite commence avant la commande `apt`. Il s'agit de garantir un retour en arrière possible si quelque chose tourne mal. Si le nœud est installé sur ZFS, un snapshot du root avant l'upgrade prend quelques secondes et permet un rollback complet via l'environnement de secours en cas d'incident kernel ou de régression majeure.

⚡ **Retour terrain**

Pour une collectivité qui migrait de VMware vSphere vers Proxmox VE après la fin des licences perpétuelles, le point de friction principal était la migration des VMs avec des snapshots anciens et des disques fragmentés. J'ai développé un playbook de migration en 3 phases : consolidation des snapshots avec qemu-img, conversion OVA→qcow2, et validation post-migration par comparaison MD5 des fichiers critiques. Sur 180 VMs, 94 % ont migré sans incident.

```
 # Sauvegarde des VM critiques
vzdump --all --storage <storage>
# Sauvegarde ciblée d'une VM (mode snapshot, sans arrêt)
vzdump <VMID> --mode snapshot --storage <storage>

# Snapshot ZFS du root avant upgrade (rollback possible)
zfs snapshot rpool/ROOT/pve-1@pre-upgrade-$(date +%F)
# Vérifier la présence du snapshot
zfs list -t snapshot | grep pre-upgrade
```

Avant d'agir, vérifier également l'espace disque disponible et l'état du cluster. Sur un nœud appartenant à un cluster, l'upgrade ne doit jamais être lancé si le cluster est hors quorum. Une mise à jour partielle entraînerait des incohérences de `pmxcfs` et bloquerait l'API Proxmox.

```
 # Vérification espace et cluster
df -h / /var /boot         # Au moins 2-3 GiB libres sur /
pvecm status               # Quorum présent (Expected/Total cohérents)
pvecm nodes                # Liste des nœuds vus par corosync
ha-manager status          # État des ressources HA (si HA configuré)
qm list ; pct list         # Inventaire VM/CT avant migration
```

Consultez notre guide des stratégies de backup et restauration Proxmox pour une politique de sauvegarde complète avant toute opération d'upgrade.

---

3. Configuration des dépôts apt pour Proxmox VE 9

Sur Proxmox VE 9, les dépôts pointent vers la suite Debian **trixie**. Trois fichiers sont concernés : `/etc/apt/sources.list` (dépôts Debian standard), `/etc/apt/sources.list.d/pve-enterprise.list` (créé par défaut à l'installation) et `/etc/apt/sources.list.d/ceph.list` (si Ceph est installé). Le dépôt enterprise exige un identifiant et une clé de souscription valides ; sans souscription, il retournera `HTTP/1.1 401 Unauthorized` et bloquera l'ensemble du processus.

3.1 Désactivation du dépôt enterprise

```
 # Sauvegarder le fichier original avant modification
cp /etc/apt/sources.list.d/pve-enterprise.list \
   /etc/apt/sources.list.d/pve-enterprise.list.bak

# Commenter toutes les lignes 'deb' du dépôt enterprise
sed -i 's/^deb/# deb/g' /etc/apt/sources.list.d/pve-enterprise.list

# Idem pour le dépôt Ceph enterprise s'il existe
[ -f /etc/apt/sources.list.d/ceph.list ] && \
  sed -i 's/^deb/# deb/g' /etc/apt/sources.list.d/ceph.list

# Vérification
grep -h '^deb' /etc/apt/sources.list.d/*.list /etc/apt/sources.list 2>/dev/null
```

3.2 Ajout du dépôt no-subscription

```
 # Créer le fichier du dépôt no-subscription pour PVE 9 (Debian Trixie)
echo "deb http://download.proxmox.com/debian/pve trixie pve-no-subscription" \
  > /etc/apt/sources.list.d/pve-install-repo.list

# Si Ceph est installé, ajouter aussi le dépôt no-subscription Ceph Squid
echo "deb http://download.proxmox.com/debian/ceph-squid trixie no-subscription" \
  > /etc/apt/sources.list.d/ceph-no-subscription.list

# Vérifier la clé GPG du dépôt Proxmox (déjà présente après install)
ls -la /usr/share/keyrings/proxmox-archive-keyring.gpg
```

Le dépôt no-subscription est techniquement de qualité production : les paquets y sont validés et stables. Il convient parfaitement aux homelabs et aux PME. Pour un environnement critique avec exigence de support 24/7, la souscription enterprise reste l'option recommandée par Proxmox Server Solutions.

4. Procédure de mise à jour complète

La procédure standard se déroule en quatre étapes : rafraîchir les listes de paquets (`apt update`), inspecter ce qui va changer (`apt list --upgradable`), appliquer la mise à jour complète (`apt full-upgrade`), puis redémarrer si le kernel, libc ou le module ZFS ont été mis à jour. L'option `full-upgrade` est préférée à `upgrade` car elle gère les changements de dépendances inévitables sur Proxmox (paquets retirés, ajout de nouveaux paquets noyau).

4.1 Rafraîchir les listes de paquets

```
 # Rafraîchir l'index des paquets disponibles
apt update

# Lister précisément les paquets à mettre à jour avant action
apt list --upgradable 2>/dev/null

# Voir le détail d'un paquet précis (versions installée / candidate)
apt-cache policy pve-manager pve-kernel-6.14 proxmox-ve
```

4.2 Appliquer la mise à jour complète

```
 # Mise à jour complète avec gestion des dépendances changées
apt full-upgrade

# Variante non interactive pour scripting
DEBIAN_FRONTEND=noninteractive apt full-upgrade -y \
  -o Dpkg::Options::="--force-confdef" \
  -o Dpkg::Options::="--force-confold"

# Nettoyage post-upgrade
apt autoremove --purge
apt clean
```

Note : `apt upgrade` refuse d'installer un paquet qui en supprimerait un autre, ce qui bloque fréquemment sur Proxmox lors de la rotation de kernels ou du remplacement de paquets méta.

`apt full-upgrade` (équivalent de `apt-get dist-upgrade`) gère ces changements correctement et est la commande recommandée par Proxmox depuis PVE 7.

4.3 Migration majeure PVE 8 vers PVE 9

Une migration de version majeure (8.x vers 9.x) ne se réduit pas à un `apt full-upgrade`. L'outil `pve8to9` doit être exécuté en amont pour détecter les prérequis non remplis, services incompatibles et configurations à ajuster. Proxmox ne supporte pas le saut direct PVE 7 vers PVE 9 ; la séquence imposée est 7 vers 8 vers 9, avec validation et redémarrage entre chaque étape.

```
 # Prérequis : la version PVE 8 doit être à jour AVANT le saut
apt update && apt full-upgrade -y

# Lancer le contrôle pve8to9 (rapport détaillé)
pve8to9 --full

# Bascule des dépôts bookworm vers trixie (après validation du rapport)
sed -i 's/bookworm/trixie/g' /etc/apt/sources.list
sed -i 's/bookworm/trixie/g' /etc/apt/sources.list.d/*.list

# Upgrade majeur
apt update && apt full-upgrade -y
```

4.4 Redémarrage du nœud

```
 # Vérifier si un reboot est nécessaire (kernel/libc/zfs)
[ -f /run/reboot-required ] && cat /run/reboot-required.pkgs

# Comparer kernel installé vs kernel actif
echo "Actif : $(uname -r)"
echo "Installé : $(ls /lib/modules/ | sort -V | tail -1)"

# Redémarrage
reboot
```

Un redémarrage est obligatoire après mise à jour du kernel (`proxmox-kernel-*`), de la glibc (`libc6`), du module ZFS (`zfs-dkms` , `zfs-modules-*`) ou de systemd. Le fichier `/run/reboot-required` liste précisément ces dépendances.

5. Vérifications après redémarrage

Le redémarrage doit être suivi d'une série de contrôles méthodiques pour confirmer que tout est rentré dans l'ordre. Cette checklist détecte les régressions silencieuses qui peuvent ne pas être visibles immédiatement.

```
 # Checklist post-reboot
pveversion                 # Version PVE active — doit refléter la mise à jour
uname -r                   # Kernel chargé — doit être le plus récent installé
pvecm status               # Quorum cluster et version corosync homogène
pvecm nodes                # Tous les nœuds en statut 'A' (alive)
zpool status               # Pas de pool DEGRADED, FAULTED ou erreur de CRC
pvesr status               # Jobs de réplication ZFS : dernier passage OK
qm list                    # Inventaire VM : statuts running attendus
pct list                   # Inventaire CT : statuts running attendus
ha-manager status          # État des ressources HA (si configuré)
systemctl --failed         # Aucun service en failed après reboot
journalctl -p err -b       # Aucune erreur kernel/systemd depuis ce boot
curl -k https://localhost:8006  # Interface web PVE accessible
```

Pour la supervision de la réplication ZFS, consultez notre guide Proxmox réplication ZFS. La documentation officielle des commandes PVE est disponible sur le wiki Proxmox — gestion des dépôts.

6. Rolling upgrade sur un cluster Proxmox

Sur un cluster, la règle d'or est de mettre à jour les nœuds un par un. L'ordre recommandé : nœud(s) de witness/quorum en premier, nœud le plus chargé en dernier. Avant de redémarrer un

nœud, migrer ses VM HA vers un autre nœud du cluster pour éviter un basculement de service imprévu. Un cluster Proxmox tolère un écart temporaire d'une version mineure entre ses nœuds le temps d'une rolling-update.

```
 # Migration à chaud d'une VM vers un autre nœud
qm migrate <VMID> <node-cible> --online

# Migration à chaud incluant les disques locaux
qm migrate <VMID> <node-cible> --online --with-local-disks

# Migration d'un conteneur LXC (redémarrage)
pct migrate <CTID> <node-cible> --restart

# Migration HA propre (respecte les contraintes)
ha-manager migrate <sid> <node-cible>

# Sortir temporairement une ressource du HA
ha-manager set <sid> --state stopped
```

Pour la gestion avancée de la haute disponibilité en cluster, consultez notre guide clustering Proxmox et haute disponibilité.

```
 # État des jobs de réplication ZFS
pvesr status
pvesr list
# Forcer l'exécution immédiate d'un job
pvesr run --id <vmid>-<nb>
# Snapshots de réplication
zfs list -t snapshot | grep __replicate
```

7. Résolution des problèmes courants

Quatre incidents reviennent régulièrement après une mise à jour Proxmox : erreur 401 sur le dépôt enterprise, espace disque insuffisant en cours d'upgrade, échec d'un job de réplication ZFS au reboot, et boot dégradé sur un nouveau kernel.

Symptôme	Résolution
`E: Failed to fetch ... 401 Unauthorized`	Dépôt enterprise actif sans souscription — désactiver (§3.1)
`E: You don't have enough free space`	Manque d'espace dans /var — `apt clean` puis relancer
pvesr status : LAST_SYNC ancien / FAIL	Job cassé — `pvesr disable` puis `pvesr enable`
Kernel panic au boot après upgrade	Sélectionner ancien kernel via GRUB, puis pin du kernel
Interface web 502 Bad Gateway	`systemctl restart pveproxy pvedaemon pve-cluster`
Conflit apt avec proxmox-ve	`apt-get install -f` — ne JAMAIS retirer proxmox-ve

7.2 Rollback via snapshot ZFS du root

```
 # Si le root est sur ZFS et qu'un snapshot pre-upgrade existe :
# Booter en mode rescue depuis l'ISO Proxmox, puis :
zpool import -f rpool
zfs rollback -r rpool/ROOT/pve-1@pre-upgrade-AAAA-MM-JJ
zpool export rpool
reboot
```

Pour les stratégies d'optimisation post-upgrade, consultez notre guide d'optimisation Proxmox VE expert. Pour la sécurisation post-upgrade, notre guide de durcissement Proxmox VE 9 couvre toutes les bonnes pratiques.

8. Récapitulatif des commandes par phase

Phase	Commande	Description
Préparation	`pveversion -v`	État détaillé des paquets PVE
Préparation	`vzdump --all --storage <st>`	Sauvegarde globale des VM
Préparation	`zfs snapshot rpool/ROOT/pve-1@pre-upgrade`	Snapshot du root ZFS
Préparation	`pvecm status`	Quorum cluster
Dépôts	`sed -i 's/^deb/# deb/g' .../pve-enterprise.list`	Désactiver enterprise
Dépôts	`apt update`	Rafraîchir listes
Upgrade	`apt full-upgrade -y`	Mise à jour complète
Upgrade	`apt autoremove --purge`	Nettoyage post-upgrade
Post-reboot	`pveversion ; uname -r`	Confirmer versions
Post-reboot	`pvecm status ; zpool status`	Cluster et ZFS sains
Post-reboot	`pvesr status ; systemctl --failed`	Réplications et services

FAQ sur la mise à jour Proxmox VE 9

Peut-on mettre à jour Proxmox VE 9 sans arrêter les VM ?

Oui, pour les mises à jour courantes (corrections de sécurité, bugfixes sans changement de kernel), les VM restent actives durant l'`apt full-upgrade`. Le redémarrage n'est nécessaire que si le kernel, la glibc ou le module ZFS ont été mis à jour (vérifier avec `/run/reboot-required`). Dans ce cas, sur un cluster, migrez les VM vers un autre nœud avant de redémarrer. Sur un nœud isolé, une brève interruption de service est inévitable.

Quelle est la différence entre pve-enterprise et pve-no-subscription ?

Le dépôt `pve-enterprise` exige une souscription payante auprès de Proxmox Server Solutions. Il offre un cycle de validation plus long et un engagement contractuel de support. Le dépôt `pve-no-subscription` est gratuit, de qualité production, avec un cycle de validation légèrement plus court. Les paquets sont techniquement identiques ; la différence porte sur l'engagement support et la priorité de correction. Pour un homelab ou une PME sans exigence de SLA, `pve-no-subscription` est parfaitement adapté.

Comment gérer la migration majeure de PVE 8 vers PVE 9 ?

La migration PVE 8 vers 9 n'est pas un simple `apt full-upgrade`. Elle requiert : (1) que PVE 8 soit complètement à jour, (2) l'exécution de `pve8to9 --full` pour détecter les incompatibilités, (3) la bascule des sources.list de `bookworm` vers `trixie`, (4) un `apt update && apt full-upgrade`, (5) un redémarrage et la checklist de validation. Ne jamais sauter de version majeure.

Que faire si apt full-upgrade propose de supprimer proxmox-ve ?

Ne jamais accepter cette suppression. Le méta-paquet `proxmox-ve` porte toutes les dépendances qui font fonctionner l'hyperviseur (pve-manager, qemu-server, pve-cluster). Sa suppression désinstalle Proxmox et rend le nœud inutilisable. Résoudre le conflit via `apt-get install -f` ou en épinglant manuellement les versions problématiques.

Comment effectuer un rollback après une mise à jour qui pose problème ?

Si un snapshot ZFS du root a été créé avant l'upgrade, booter depuis l'ISO Proxmox en mode rescue, importer le pool (`zpool import -f rpool`), appliquer le rollback (`zfs rollback -r rpool/ROOT/pve-1@pre-upgrade-AAAA-MM-JJ`), puis redémarrer. Cette procédure restaure le système en quelques minutes. Sans snapshot, une réinstallation complète et restauration depuis sauvegarde vzdump est nécessaire.

Quand un redémarrage est-il obligatoire après apt full-upgrade ?

Un redémarrage est obligatoire lorsque les paquets `proxmox-kernel-*` (nouveau kernel), `libc6` (glibc), `zfs-dkms` ou `zfs-modules-*` (module ZFS), ou `systemd` ont été mis à jour. Le fichier `/run/reboot-required` est créé automatiquement dans ces cas et liste les paquets concernés dans `/run/reboot-required.pkgs`. Hors de ces cas, un `systemctl restart pveproxy pvedaemon pve-cluster` suffit.

---

9. Gestion avancée du kernel Proxmox VE 9

Proxmox VE 9 propose plusieurs options de kernel pour répondre à des besoins différents. Le kernel par défaut est la branche 6.14 (Proxmox kernel basé sur Ubuntu HWE), stable et bien testé pour la virtualisation KVM et les conteneurs LXC. Une branche opt-in 6.17 est disponible pour les matériels très récents (carte réseau 10 GbE/25 GbE, support NVMe dernière génération). Comprendre comment gérer les versions de kernel installées et éviter qu'un kernel cassé passe en défaut est essentiel en production.

```
 # Lister les kernels Proxmox installés
dpkg -l | grep proxmox-kernel
ls /lib/modules/

# Voir le kernel actif
uname -r

# Sélectionner un kernel par défaut via GRUB (si plusieurs installés)
proxmox-boot-tool kernel list
proxmox-boot-tool kernel pin 6.14.0-1-pve  # Épingler un kernel spécifique

# Supprimer un ancien kernel (libérer de l'espace)
apt remove proxmox-kernel-6.12.0-1-pve
proxmox-boot-tool refresh
```

L'outil `proxmox-boot-tool` gère l'EFI stub et le bootloader systemd-boot sur les systèmes installés en mode UEFI. Sur les systèmes BIOS legacy, c'est GRUB qui est utilisé. Après

suppression d'un kernel, toujours vérifier que le kernel par défaut est bien celui attendu avec `proxmox-boot-tool kernel list`.

10. Gestion des conteneurs LXC lors d'une mise à jour

Les conteneurs LXC sur Proxmox VE 9 utilisent le kernel de l'hôte. Une mise à jour du kernel hôte peut donc affecter les conteneurs qui dépendent de modules spécifiques ou de syscalls particuliers. En pratique, les conteneurs LXC Debian/Ubuntu suivent bien les mises à jour du kernel Proxmox, mais certains conteneurs spécialisés (notamment ceux utilisant des modules kernel compilés, comme des VPN ou des outils de capture réseau) peuvent nécessiter une vérification post-upgrade.

```
 # Arrêter tous les conteneurs avant le reboot (recommandé en production)
for ctid in $(pct list | awk 'NR>1 {print $1}'); do pct stop $ctid; done

# Vérifier l'état de tous les conteneurs
pct list

# Démarrer les conteneurs après reboot
for ctid in $(pct list | awk 'NR>1 {print $1}'); do pct start $ctid; done

# Vérifier un conteneur spécifique
pct status <CTID>
pct exec <CTID> -- uname -r  # Kernel vu par le conteneur (= kernel hôte)
```

Pour les conteneurs qui doivent rester disponibles en permanence, la stratégie recommandée est de les migrer vers un autre nœud du cluster avant le reboot du nœud en cours de mise à jour. Consultez notre guide de clustering et haute disponibilité Proxmox pour la configuration de la HA sur les conteneurs LXC.

11. Automatiser les mises à jour non critiques avec unattended-upgrades

Pour les mises à jour de sécurité Debian qui ne nécessitent pas de redémarrage (corrections de bibliothèques en userspace, updates de paquets non-kernel), il est possible de configurer `unattended-upgrades` pour une application automatique. Cette approche est recommandée uniquement pour les dépôts Debian de sécurité, jamais pour les dépôts Proxmox eux-mêmes qui nécessitent une supervision humaine.

```
 # Installer unattended-upgrades
apt install unattended-upgrades apt-listchanges

# Configuration recommandée pour Proxmox
cat > /etc/apt/apt.conf.d/50unattended-upgrades << 'EOF'
Unattended-Upgrade::Allowed-Origins {
    "Debian:trixie-security";
};
Unattended-Upgrade::Package-Blacklist {
    "proxmox-*";
    "pve-*";
    "qemu-server";
    "libpve-*";
};
Unattended-Upgrade::AutoFixInterruptedDpkg "true";
Unattended-Upgrade::Mail "admin@exemple.com";
EOF

# Activer et vérifier
systemctl enable --now unattended-upgrades
unattended-upgrade --dry-run -d
```

En blacklistant tous les paquets `proxmox-*` et `pve-*`, on s'assure que les mises à jour automatiques ne touchent que les composants Debian standard. Les mises à jour Proxmox restent sous contrôle humain total, appliquées manuellement lors de fenêtres de maintenance planifiées. Pour la sécurisation complète de votre nœud Proxmox, consultez notre guide de durcissement Proxmox VE 9 qui couvre les configurations de pare-feu, l'accès SSH renforcé et l'audit des services.

À RETENIR

À retenir

Toujours créer un snapshot ZFS du root et sauvegarder les VM critiques avec vzdump avant toute mise à jour majeure

Sur un nœud sans souscription : désactiver `pve-enterprise.list` AVANT `apt update`, sans quoi l'erreur 401 bloque tout le processus

Utiliser systématiquement `apt full-upgrade` (jamais `apt upgrade`) : seul full-upgrade gère les rotations de kernels et changements de dépendances Proxmox

Sur un cluster : rolling upgrade nœud par nœud, migration HA préalable obligatoire, vérifier `pvecm status` entre chaque nœud

Ne jamais accepter la suppression du méta-paquet `proxmox-ve` lors d'un conflit apt — résoudre via `apt-get install -f`

Checklist post-reboot obligatoire : pveversion, uname -r, pvecm status, zpool status, pvesr status, systemctl --failed, journalctl -p err -b

Pour PVE 8 vers 9 : exécuter `pve8to9 --full` avant la bascule des dépôts ; ne jamais sauter une version majeure

Sources et références

ANSSI — Recommandations de sécurité relatives aux hyperviseurs

NIST SP 800-125 — Guide to Security for Full Virtualization

