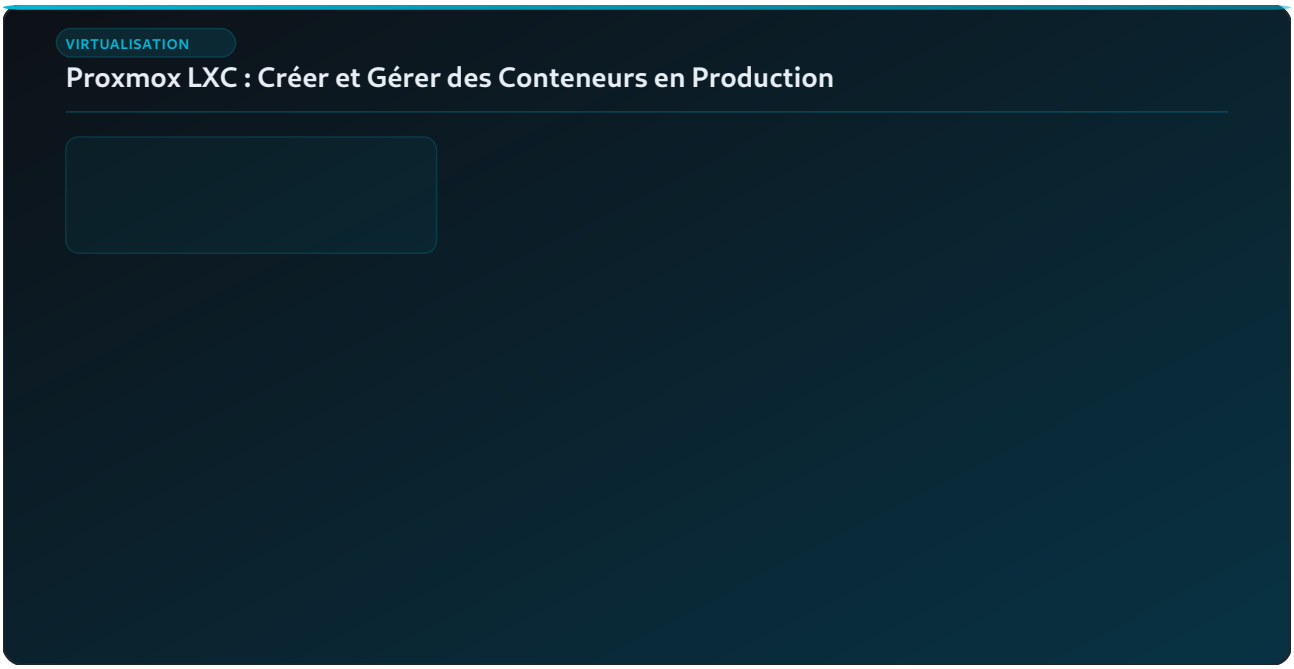
Proxmox VE : créer et gérer des conteneurs LXC en production

> Guide complet pour créer, configurer et sécuriser des conteneurs LXC sur Proxmox VE. Templates, réseau, stockage, snapshots, sauvegardes et bonnes pratiques...

La virtualisation légère par conteneurs LXC (Linux Containers) représente une alternative puissante aux machines virtuelles complètes pour de nombreux cas d'usage en production. Là où une VM KVM consomme des ressources pour émuler un matériel complet et faire tourner un kernel indépendant, un conteneur LXC partage le kernel de l'hôte et offre un environnement Linux isolé avec une surcharge proche de zéro. Proxmox VE, la plateforme de virtualisation open source qui s'est imposée comme standard dans les infrastructures IT de taille moyenne, intègre nativement LXC depuis sa version 4.0, aux côtés de la virtualisation KVM complète. Le résultat est une plateforme unifiée où vous pouvez gérer des VMs lourdes et des conteneurs légers depuis la même interface web, avec les mêmes outils de sauvegarde, snapshot et haute disponibilité. Ce guide complet vous accompagne dans la création, la configuration et la gestion de conteneurs LXC en production sur Proxmox VE : des templates disponibles jusqu'aux sauvegardes automatisées, en passant par la configuration réseau avancée, la gestion du stockage, et les bonnes pratiques de sécurité pour les environnements de production.



Le choix entre LXC et KVM sur Proxmox dépend du cas d'usage. Comprendre les différences fondamentales est essentiel pour prendre la bonne décision.

LXC (Conteneurs) :

Partage le kernel Linux de l'hôte Proxmox

Démarrage en moins d'une seconde

Overhead mémoire minimal (quelques Mo)

Isolation des processus, réseau et système de fichiers via namespaces

Idéal pour : serveurs web, bases de données légères, services DNS, VPN, monitoring

Limitation : ne peut faire tourner que des distributions Linux, pas de Windows ni de kernel personnalisé

KVM (VMs complètes) :

Virtualisation complète du matériel (CPU, RAM, disque, réseau)

Kernel indépendant, isolation maximale

Overhead de 5 à 15% sur les performances

Idéal pour : Windows Server, distributions Linux nécessitant un kernel spécifique, applications nécessitant un accès bas niveau au matériel

Règle de décision simple : utilisez LXC pour les services stateless ou facilement reproductibles (nginx, bind9, cron jobs, petits services), et KVM pour les workloads critiques nécessitant une isolation maximale ou des OS non-Linux. En production, une combinaison typique serait : 3 à 5 VMs KVM pour les systèmes critiques (Active Directory, base de données primaire, pare-feu pfSense) et une quinzaine de conteneurs LXC pour les services périphériques.

Télécharger et utiliser les templates LXC (pct templates, Debian, Ubuntu, Alpine)

Proxmox VE propose un catalogue de templates LXC officiels téléchargeables directement depuis l'interface web ou la CLI. Ces templates sont des archives compressées contenant un système de base minimal.

⚡ **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.

Depuis l'interface web Proxmox, naviguez vers votre stockage (généralement "local"), sélectionnez "Content > CT Templates", puis cliquez sur "Templates". Un catalogue apparaît avec les distributions disponibles : Debian 12, Ubuntu 22.04/24.04, Alpine 3.19, CentOS Stream, Fedora, OpenSUSE, etc.

Via la CLI sur le nœud Proxmox :

```
 # Lister les templates disponibles en ligne
pveam update
pveam available --section system

# Télécharger Debian 12
pveam download local debian-12-standard_12.7-1_amd64.tar.zst

# Télécharger Ubuntu 22.04
pveam download local ubuntu-22.04-standard_22.04-1_amd64.tar.zst

# Télécharger Alpine Linux (très léger, ~3Mo)
pveam download local alpine-3.19-default_20240207_amd64.tar.xz

# Lister les templates locaux
pveam list local
```

Les templates sont stockés dans `/var/lib/vz/template/cache/` sur le nœud Proxmox. Ils servent de base pour créer des conteneurs : Proxmox extrait l'archive et crée le système de fichiers du conteneur à partir du template.

Pour les environnements avec des exigences spécifiques, vous pouvez créer vos propres templates LXC en exportant un conteneur existant que vous avez personnalisé (packages pré-installés, configuration de base, utilisateurs). Cette approche de "golden template" garantit la cohérence de tous vos conteneurs.

Créer un conteneur LXC via l'interface web Proxmox

L'interface web Proxmox (accessible sur le port 8006 en HTTPS) offre un assistant de création de conteneur en plusieurs étapes.

Cliquez sur "Create CT" dans le menu du nœud. L'assistant vous guide à travers :

General : VMID (identifiant unique, ex: 200), Hostname (ex: web-nginx), Password et/ou clé SSH publique, Resource Pool si applicable

Template : sélectionnez le template téléchargé (debian-12-standard)

Disks : taille du disque racine (recommandé : 8 Go minimum), type de stockage (local-lvm, ZFS, Ceph)

CPU : nombre de cores alloués (1-2 pour les services légers)

Memory : RAM en Mo (512 à 2048 selon le service), Swap recommandé à 512 Mo

Network : interface réseau (eth0), bridge (vmbr0), adresse IP (DHCP ou statique)

DNS : serveurs DNS et domaine de recherche

Options importantes à configurer :

Unprivileged Container : cochez cette case pour la sécurité (recommandé, voir section dédiée)

Nesting : activez si le conteneur doit faire tourner Docker à l'intérieur

Start after created : démarrage automatique après création

Une fois le conteneur créé, vous pouvez ouvrir une console depuis l'interface web (onglet "Console") ou via SSH si vous avez configuré une IP statique et une clé SSH.

Créer un conteneur LXC via la CLI (pct create, pct start, pct enter)

La commande `pct` (Proxmox Container Tool) permet de gérer entièrement les conteneurs LXC depuis la ligne de commande. C'est l'approche préférée pour l'automatisation et les déploiements en masse.

Syntaxe complète pour créer un conteneur Debian 12 non-privilégié :

```
pct create 201 local:vztmpl/debian-12-standard_12.7-1_amd64.tar.zst \
  --hostname mon-service \
  --memory 1024 \
  --swap 512 \
  --cores 2 \
  --rootfs local-lvm:8 \
  --net0 name=eth0,bridge=vmbr0,ip=192.168.1.201/24,gw=192.168.1.1 \
  --nameserver 1.1.1.1 \
  --searchdomain lan.local \
  --unprivileged 1 \
  --features nesting=0 \
  --ostype debian \
  --ssh-public-keys /root/.ssh/authorized_keys \
  --start 1
```

Commandes de gestion courantes :

```
# Démarrer/arrêter/redémarrer
pct start 201
pct stop 201
pct reboot 201

# Entrer dans le conteneur (shell interactif)
pct enter 201

# Exécuter une commande dans le conteneur sans ouvrir un shell
pct exec 201 -- apt-get update && apt-get upgrade -y

# Voir le statut
pct status 201

# Lister tous les conteneurs
pct list
```

Pour modifier les ressources d'un conteneur en cours de fonctionnement :

```
# Ajouter de la RAM à chaud
pct set 201 --memory 2048

# Redimensionner le disque (uniquement agrandissement)
pct resize 201 rootfs +4G
```

La possibilité de redimensionner le disque à chaud est un avantage majeur de LXC par rapport aux VMs qui nécessitent souvent un redémarrage et une opération de filesystem resize manuelle.

Configurer le réseau des conteneurs (bridge, VLAN, IP statique)

La configuration réseau des conteneurs LXC sur Proxmox offre une grande flexibilité. Proxmox gère les interfaces réseau via des bridges Linux (`vmbr0` , `vmbr1` , etc.) qui correspondent aux segments réseau physiques ou virtuels.

Configuration IP statique post-création :

```
pct set 201 --net0 name=eth0,bridge=vmbr0,ip=192.168.1.201/24,gw=192.168.1.1
```

Pour une configuration VLAN (indispensable en production pour segmenter les réseaux) :

```
pct set 201 --net0 name=eth0,bridge=vmbr0,ip=10.20.30.201/24,gw=10.20.30.1,tag=20
```

Le paramètre `tag=20` configure le VLAN 802.1Q sur l'interface du conteneur. Le bridge `vmbr0` doit être configuré en mode "VLAN aware" dans la configuration réseau Proxmox.

Pour un conteneur avec plusieurs interfaces réseau (DMZ + LAN, par exemple) :

```
pct set 201 \
  --net0 name=eth0,bridge=vmbr0,ip=192.168.1.201/24,gw=192.168.1.1 \
  --net1 name=eth1,bridge=vmbr1,ip=10.0.0.201/24
```

La configuration réseau interne au conteneur (/etc/network/interfaces sur Debian) est gérée automatiquement par Proxmox lors de la création. Pour des configurations avancées, éditez directement les fichiers réseau à l'intérieur du conteneur via `pct enter` .

Gérer le stockage LXC (volumes, bind mounts pour données persistantes)

Le stockage des conteneurs LXC sur Proxmox offre plusieurs options selon les besoins de performance et de résilience.

Stockages supportés pour les rootfs LXC :

LVM-thin : recommandé pour les déploiements classiques, snapshots efficaces

ZFS : snapshots natifs, compression, checksumming — idéal pour les environnements critiques

Directory : simple répertoire sur le système de fichiers hôte, moins performant mais plus simple

Ceph RBD : pour les clusters Proxmox avec haute disponibilité

Bind mounts : pour monter des répertoires de l'hôte dans un conteneur (données persistantes, partages NFS) :

```
 # Monter /data/nginx-www de l'hôte dans /var/www/html du conteneur
pct set 201 --mp0 /data/nginx-www,mp=/var/www/html

# Monter en lecture seule
pct set 201 --mp1 /data/configs,mp=/etc/app-config,ro=1
```

Attention : les bind mounts sur des conteneurs non-privilégiés nécessitent que les UID/GID soient correctement mappés. L'utilisateur www-data (uid 33) dans le conteneur correspond à un UID différent sur l'hôte (uid 100033 avec le mapping par défaut). Ajustez les permissions en conséquence sur le répertoire hôte :

```
chown 100033:100033 /data/nginx-www
```

Pour les volumes de données importants (bases de données), préférez un disque dédié plutôt qu'un bind mount :

```
pct set 201 --mp0 local-lvm:20,mp=/var/lib/mysql
```

Conteneurs privilégiés vs non-privilégiés — sécurité et cas d'usage

La distinction entre conteneurs privilégiés et non-privilégiés est fondamentale pour la sécurité des déploiements LXC en production.

Conteneurs non-privilégiés (recommandés) : les UID/GID à l'intérieur du conteneur sont mappés vers des UID/GID élevés sur l'hôte via les user namespaces. L'utilisateur root (uid 0) dans le conteneur correspond à l'uid 100000 sur l'hôte. Même en cas de sortie du conteneur

(container escape), l'attaquant n'obtient que les droits d'un utilisateur non-privilégié sur l'hôte. C'est la configuration par défaut et recommandée pour tous les nouveaux conteneurs.

Conteneurs privilégiés : le root dans le conteneur est le root sur l'hôte. Plus simple à configurer (pas de problèmes de mapping UID), mais en cas de vulnérabilité du kernel ou de l'isolation LXC, un attaquant peut compromettre l'hôte entier. Utilisez les conteneurs privilégiés uniquement pour des cas spécifiques nécessitant un accès direct au kernel (NFS server, certains drivers).

Fonctionnalités nécessitant des configurations spéciales sur conteneurs non-privilégiés :

```
 # Activer Docker dans un conteneur LXC non-privilégié
pct set 201 --features nesting=1,keyctl=1

# Activer FUSE (nécessaire pour certains filesystems)
pct set 201 --features fuse=1

# Activer le support tun/tap (VPN dans un conteneur)
pct set 201 --features tun=1
```

Consultez notre article sur la sécurité des conteneurs et isolation pour approfondir les mécanismes de sécurité kernel utilisés par LXC.

Snapshots et clones de conteneurs LXC

Proxmox VE offre des fonctionnalités de snapshot et clone pour les conteneurs LXC, permettant des workflows de déploiement efficaces.

Snapshots : capture de l'état complet du conteneur (système de fichiers + configuration) à un instant T. Idéal avant une mise à jour majeure ou un changement de configuration risqué.

```
 # Créer un snapshot (le conteneur peut être running)
pct snapshot 201 avant-mise-a-jour --description "Avant upgrade Debian 11 → 12"

# Lister les snapshots
pct listsnapshot 201

# Restaurer un snapshot
pct rollback 201 avant-mise-a-jour

# Supprimer un snapshot
pct delsnapshot 201 avant-mise-a-jour
```

Note importante : les snapshots LXC sur LVM-thin fonctionnent en mode copy-on-write et sont très efficaces en espace disque. Sur ZFS, les snapshots sont encore plus performants et atomiques. Sur les stockages directory classiques, les snapshots ne sont pas disponibles.

Clones : crée une copie complète d'un conteneur existant. Utile pour déployer rapidement des environnements identiques (staging, tests).

```
 # Clone complet (full clone)
pct clone 201 202 --full --hostname clone-test --storage local-lvm

# Clone lié (linked clone) — partage les blocs initiaux, économise de l'espace
pct clone 201 203 --hostname linked-test
```

Les clones liés sont particulièrement efficaces pour créer des environnements de test temporaires : ils démarrent quasi-instantanément et ne consomment de l'espace que pour leurs données différentielles.

Sauvegarder les conteneurs avec Proxmox Backup Server

La sauvegarde en production est non-négociable. Proxmox propose deux solutions de sauvegarde pour les conteneurs LXC :

Backup intégré Proxmox (vzdump) : crée une archive du conteneur, avec ou sans compression. Peut s'exécuter sur un conteneur en cours de fonctionnement (mode snapshot) ou après arrêt.

```
 # Sauvegarde manuelle vers le stockage local
vzdump 201 --storage local --compress zstd --mode snapshot

# Planifier des sauvegardes automatiques
# Dans l'interface web : Datacenter > Backup > Add
# Ou via /etc/pve/jobs.cfg
```

Proxmox Backup Server (PBS) : solution dédiée aux sauvegardes avec déduplication et chiffrement. Installez PBS sur un serveur séparé ou une VM dédiée :

```
 # Configurer PBS comme storage dans Proxmox
pvesm add pbs backup-server \
  --server backup.votre-domaine.fr \
  --datastore main \
  --username backup-user@pbs \
  --fingerprint EMPREINTE_CERTIFICAT
```

PBS offre la déduplication inter-conteneurs : si 10 conteneurs Debian 12 partagent les mêmes fichiers système, PBS les stocke une seule fois. La réduction d'espace peut atteindre 70-80% par rapport aux sauvegardes classiques. La restauration granulaire permet de récupérer des fichiers individuels depuis une sauvegarde sans restaurer l'intégralité du conteneur.

Planification recommandée en production : sauvegarde quotidienne sur PBS avec rétention 30 jours, sauvegarde hebdomadaire sur un stockage externe (bande, S3) avec rétention 6 mois. Consultez notre guide sur la stratégie de sauvegarde 3-2-1 pour les détails.

Automatiser avec l'API Proxmox et des scripts Bash

Proxmox expose une API REST complète pour l'automatisation de toutes les opérations. Cette API est documentée et accessible à `https://votre-proxmox:8006/api2/json`.

Exemple de script Bash pour créer automatiquement 5 conteneurs de test identiques :

```
 #!/bin/bash
TEMPLATE="local:vztmpl/debian-12-standard_12.7-1_amd64.tar.zst"
BASE_VMID=300
BASE_IP="192.168.2"
PASSWORD=$(openssl rand -base64 16)

for i in $(seq 1 5); do
  VMID=$((BASE_VMID + i))
  IP="${BASE_IP}.${VMID}/24"

  pct create $VMID $TEMPLATE \
    --hostname "test-${i}" \
    --memory 512 \
    --cores 1 \
    --rootfs local-lvm:4 \
    --net0 name=eth0,bridge=vmbr0,ip=$IP,gw=192.168.2.1 \
    --unprivileged 1 \
    --password $PASSWORD

  pct start $VMID
  echo "Conteneur $VMID (test-${i}) créé, IP: $IP"
done
```

Pour l'API REST avec authentification par token :

```
 # Créer un token API dans l'interface web : Datacenter > Permissions > API Tokens
TOKEN="USER@pam!mon-token=VALEUR-TOKEN"

# Lister tous les conteneurs via l'API
curl -s -H "Authorization: PVEAPIToken=$TOKEN" \
  https://proxmox.local:8006/api2/json/nodes/pve/lxc \
  --insecure | python3 -m json.tool

# Créer un conteneur via l'API
curl -s -X POST \
  -H "Authorization: PVEAPIToken=$TOKEN" \
  -d "vmid=205&hostname=api-test&ostemplate=local:vztmpl/debian-12-standard_12.7-1_amd64.tar.zst&m
  https://proxmox.local:8006/api2/json/nodes/pve/lxc \
  --insecure
```

Pour les déploiements infrastructure-as-code, le provider Terraform Proxmox (`bpg/proxmox`) permet de gérer vos conteneurs LXC dans des fichiers HCL versionnés. Consultez la documentation officielle Proxmox LXC et linuxcontainers.org pour les spécifications techniques détaillées.

L'automatisation de la gestion des conteneurs LXC s'intègre naturellement dans un pipeline de déploiement complet. Associée à Ansible pour la configuration interne des conteneurs et à GitLab CI pour l'orchestration, vous obtenez une infrastructure entièrement déclarative et reproductible. Consultez notre article sur la automatisation avec Ansible pour l'étape suivante de votre infrastructure.

Haute disponibilité et clustering Proxmox pour les conteneurs LXC

En production, la haute disponibilité (HA) est souvent une exigence critique. Proxmox VE supporte le clustering de plusieurs nœuds et la haute disponibilité automatique pour les conteneurs LXC, permettant le redémarrage automatique d'un conteneur sur un autre nœud en cas de défaillance matérielle.

Un cluster Proxmox HA nécessite :

Minimum 3 nœuds Proxmox (pour le quorum)

Un stockage partagé accessible depuis tous les nœuds (Ceph, NFS/iSCSI, ou Proxmox Backup Server)

Un réseau dédié pour la réplication (optionnel mais recommandé)

Configuration HA pour un conteneur LXC depuis la CLI :

```
 # Ajouter un conteneur au groupe HA
ha-manager add ct:201 --group production --max_restart 3 --max_relocate 1

# Lister les ressources HA
ha-manager status

# Vérifier l'état du quorum
pvecm status
```

En cas de panne du nœud hébergeant le conteneur 201, le gestionnaire HA détecte la défaillance (délai typique : 30 à 60 secondes) et redémarre automatiquement le conteneur sur le nœud disponible suivant dans le groupe HA. Cette capacité est particulièrement précieuse pour les services critiques comme les serveurs DNS internes, les proxies ou les serveurs DHCP.

Pour les environnements nécessitant une disponibilité encore plus forte (RTO proche de zéro), la **réplication Proxmox** permet de synchroniser régulièrement un conteneur vers un nœud secondaire. En cas de panne, le basculement est quasi-instantané car les données sont déjà présentes sur le nœud de secours :

```
 # Configurer la réplication d'un conteneur vers noeud2 (toutes les 15 min)
pvesr create --guest 201 --target noeud2 --schedule "*/15" --rate 10
```

Le paramètre `--rate 10` limite la bande passante de réplication à 10 Mo/s pour ne pas saturer le réseau de production. Cette configuration de réplication est transparente pour les utilisateurs et ne perturbe pas le fonctionnement du conteneur.

La combinaison clustering Proxmox + HA + réplication constitue une infrastructure LXC de niveau entreprise, comparable en termes de disponibilité aux solutions propriétaires VMware vSphere ou Microsoft Hyper-V, mais sans les coûts de licence associés. Consultez notre article sur la configuration d'un cluster Proxmox avec Ceph pour le guide détaillé de déploiement.

Questions fréquentes sur Proxmox LXC

Peut-on faire tourner Docker dans un conteneur LXC Proxmox ?

Oui, mais cela nécessite une configuration spécifique. Activez le nesting et keyctl sur le conteneur : `pct set 201 --features nesting=1,keyctl=1`. Ensuite, installez Docker normalement à l'intérieur du conteneur. Cette configuration "Docker dans LXC" est couramment utilisée pour des raisons d'économie de ressources, mais elle nécessite un conteneur non-privilégié avec nesting et réduit légèrement l'isolation. Pour des charges de travail critiques, préférez Docker dans une VM KVM dédiée.

Quelle est la différence entre LXC et LXD sur Proxmox ?

Proxmox utilise LXC directement (via la bibliothèque liblxc), non LXD. LXD est une surcouche développée par Canonical qui ajoute des fonctionnalités comme la gestion d'images distantes, la migration à chaud et un daemon REST. Sur Proxmox, la gestion des conteneurs passe par `pct` et l'API Proxmox, qui offrent des fonctionnalités équivalentes intégrées à l'écosystème Proxmox (sauvegardes, HA, clustering).

Comment migrer un conteneur LXC d'un nœud Proxmox à un autre ?

La migration se fait via l'interface web (clic droit sur le conteneur > Migrate) ou via CLI : `pct migrate 201 noeud-destination --online`. La migration en ligne (online) est possible si les deux nœuds partagent le même stockage (Ceph, NFS). Sans stockage partagé, le conteneur doit être arrêté pour la migration. Proxmox copie les données du conteneur sur le nœud destination et le redémarre automatiquement.

Peut-on convertir une VM KVM en conteneur LXC ?

Il n'existe pas de conversion automatique directe. La procédure manuelle consiste à créer un conteneur LXC vide, puis à synchroniser les données depuis la VM via rsync (`rsync -aAXv / root@ip-lxc:/ --exclude=/proc --exclude=/sys`) depuis l'intérieur de la VM. Cette opération

nécessite une vérification attentive de la configuration réseau et des services de démarrage post-migration.

Les conteneurs LXC Proxmox sont-ils supportés pour un usage en production ?

Absolument. Proxmox LXC est utilisé en production par des milliers d'entreprises dans le monde, des hébergeurs web aux entreprises industrielles. Les conteneurs LXC Proxmox bénéficient du support des sauvegardes PBS, de la haute disponibilité avec Proxmox Cluster, et du même niveau de support que les VMs KVM. La différence est que LXC est moins isolé que KVM — pour des workloads de confiance contrôlée (vos propres services), c'est parfaitement adapté à la production.

À RETENIR

À retenir

- LXC est idéal pour les services Linux standards (web, DNS, VPN) — utilisez KVM pour Windows ou les besoins d'isolation maximale
- Toujours créer des conteneurs non-privilégiés en production pour limiter l'impact d'une éventuelle évasion de conteneur
- Les bind mounts permettent de monter des données hôte dans le conteneur, mais requièrent un ajustement des UID/GID
- Proxmox Backup Server avec déduplication est la solution recommandée pour les sauvegardes en production
- L'API REST et la CLI pct permettent d'automatiser entièrement la création et la gestion des conteneurs LXC

Sources et références

[ANSSI — Recommandations de sécurité relatives aux hyperviseurs]

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