AYI NEDJIMI Consultants

Proxmox VE 9.1 : Paramètres Avancés VM et Nested Virt

Catégorie : Virtualisation Lecture : 6 min Publié le : 22/03/2026 Auteur : Ayi NEDJIMI

Guide paramètres avancés VMs Proxmox VE 9.1 : CPU pinning, hugepages, UEFI/Secure Boot, VirtIO, agents QEMU, réseau optimisé et nested virtualization.

La configuration fine des **machines virtuelles** dans **Proxmox VE 9.1** offre un contrôle granulaire sur les performances CPU, la gestion mémoire, le firmware UEFI et les pilotes VirtIO. Ce guide des paramètres avancés détaille le **CPU pinning** pour dédier des cœurs physiques aux VMs critiques, les *hugepages (pages mémoire de grande taille, 2 Mo ou 1 Go, réduisant la pression sur le TLB)* pour améliorer les performances mémoire, la configuration réseau optimisée avec les pilotes **VirtIO paravirtualisés**, les **agents QEMU** pour l'intégration hôte-invité, ainsi que l'activation et les cas d'usage de la *virtualisation imbriquée (nested virtualization, permettant d'exécuter un hyperviseur à l'intérieur d'une VM)*. Ce guide s'adresse aux administrateurs cherchant à optimiser les performances de leurs workloads virtualisés sur Proxmox VE 9.1, avec des exemples concrets pour les cas d'usage courants : VMs haute performance, laboratoires de virtualisation, CI/CD avec Kubernetes, et environnements de test imbriqués. Chaque paramètre est expliqué avec son impact mesurable sur les performances et les compromis à considérer.

Points clés à retenir

- Le CPU pinning améliore significativement la latence des VMs critique en évitant les migrations de vCPU entre cœurs physiques (NUMA awareness).
- Les hugepages réduisent la pression sur le TLB et améliorent les performances mémoire pour les VMs avec de grandes quantités de RAM (> 4 Go).
- Les pilotes VirtIO sont obligatoires pour des performances réseau et stockage optimales : évitez les émulations SCSI/IDE/e1000 en production.
- La nested virtualization (nested KVM) est activée par un simple paramètre CPU mais nécessite que le processeur hôte supporte les extensions VMX/SVM imbriquées.

Configuration CPU Avancée : Types, Pinning et NUMA

Le **type CPU** de la VM détermine les instructions processeur exposées à l'invité. Pour Proxmox VE 9.1, les types recommandés :

- **host** : expose toutes les instructions du CPU physique (performances maximales, live migration limitée aux nœuds avec CPU identique)
- **x86-64-v3** : baseline moderne compatible avec la majorité des clusters hétérogènes
- **kvm64** : compatibilité maximale, performances réduites (éviter en production)

Le **CPU pinning** (affinité CPU) dédie des cœurs physiques spécifiques aux vCPU d'une VM. Configuration via **qm set {vmid} --cpuunits 8 --cpulimit 4** pour la priorité et la limite CPU, ou directement dans **/etc/pve/qemu-server/{vmid}.conf** avec le paramètre **affinity**. Le pinning est particulièrement bénéfique pour les VMs temps réel ou haute performance (bases de données, jeux vidéo, rendu 3D).

L'alignement **NUMA (Non-Uniform Memory Access)** est critique pour les serveurs multi-socket. Activer **numa: 1** dans la configuration de la VM assure que les vCPUs et la mémoire sont alloués dans le même domaine NUMA physique, évitant les accès mémoire cross-socket coûteux.

Gestion Mémoire : Hugepages et Ballooning

Les *hugepages* dans KVM allouent la mémoire des VMs en pages de **2 Mo (hugepages)** ou **1 Go (gigantic pages)** au lieu des pages standard de 4 Ko. Avantages : réduction des TLB misses, amélioration des performances pour les VMs mémoire-intensive de 5-15%.

Configuration des hugepages sur l'hôte Proxmox :

- Activer dans **/etc/sysctl.conf** : **vm.nr_hugepages = 512** (pour 1 Go de hugepages à 2 Mo)
- Dans la configuration VM : **hugepages: 2** (2 Mo) ou **hugepages: 1024** (1 Go)
- La mémoire hugepages est allouée statiquement et n'est pas disponible pour les autres VMs

Le **memory ballooning** (driver virtio-balloon) permet au driver KVM de récupérer dynamiquement la mémoire non utilisée d'une VM pour la redistribuer aux autres VMs. Activé par défaut dans Proxmox avec l'agent QEMU installé dans la VM. Pour les VMs de production critiques, désactiver le ballooning en fixant **balloon: 0** pour garantir la mémoire allouée.

Firmware UEFI et Secure Boot

Proxmox VE 9.1 supporte le firmware *UEFI (Unified Extensible Firmware Interface)* via OVMF (Open Virtual Machine Firmware), remplaçant le BIOS legacy. L'UEFI est requis pour :

- Les systèmes d'exploitation modernes nécessitant Secure Boot (Windows 11, RHEL 9)
- Les disques > 2 To (GPT obligatoire avec UEFI)
- Les VMs avec plus de 4 Go de mémoire adressable
- La nested virtualization avec les features CPU modernes

Configuration dans Proxmox : VM → Hardware → BIOS → OVMF (UEFI). Ajouter également un disque EFI : **Add** → **EFI Disk**. Le **Secure Boot** peut être activé ou désactivé dans les options UEFI. Attention : la migration live entre nœuds avec BIOS différents (UEFI/SeaBIOS) est impossible.

Pilotes VirtIO : Performances Réseau et Stockage

Les pilotes **VirtIO paravirtualisés** sont essentiels pour des performances optimales dans les VMs Proxmox. Ils éliminent l'overhead de l'émulation matérielle complète :

- **VirtIO SCSI** (virtio-scsi-pci) : contrôleur de stockage recommandé, supporte le TRIM/UNMAP et jusqu'à 255 disques par contrôleur
- **VirtIO Block** (virtio) : plus simple, légèrement plus performant que SCSI pour un seul disque
- **VirtIO Net** (virtio) : pilote réseau paravirtualisé, 10× plus performant que l'émulation e1000
- **VirtIO Balloon** : gestion dynamique de la mémoire
- **VirtIO RNG** : source d'entropie pour les VMs (accélère les opérations cryptographiques)

Pour les VMs Windows, les pilotes VirtIO doivent être installés depuis l'ISO **virtio-win** disponible sur le wiki Proxmox. Pour Linux, les modules VirtIO sont inclus dans le kernel depuis la version 2.6.25.

Agents QEMU et Guest Integration

L'**agent QEMU** (qemu-guest-agent) est un démon s'exécutant dans la VM invitée qui améliore l'intégration hôte-invité : snapshots cohérents (freeze/thaw du filesystem), récupération de l'adresse IP de la VM, arrêt propre depuis Proxmox et informations système dans l'interface web.

Installation dans la VM : **apt install qemu-guest-agent** (Debian/Ubuntu) ou **dnf install qemu-guest-agent** (RHEL/Fedora). Activation dans Proxmox : **qm set {vmid} --agent enabled=1**. Vérification : **qm agent {vmid} ping**. L'agent est indispensable pour des sauvegardes cohérentes (mode snapshot) et le power management correct. Pour l'administration CLI complète, consultez notre guide CLI Proxmox.

Nested Virtualization : Activation et Cas d'Usage

La **nested virtualization** (virtualisation imbriquée) permet d'exécuter un hyperviseur (VMware ESXi, KVM, Hyper-V) à l'intérieur d'une VM Proxmox. Elle est utilisée pour les laboratoires de test, la formation, le développement CI/CD avec Kubernetes (Kind, Minikube) et les environnements de test d'hyperviseurs.

Activation sur l'hôte Proxmox (Intel) : **echo "options kvm-intel nested=Y" > /etc/modprobe.d/kvm-intel.conf** puis **modprobe -r kvm-intel && modprobe kvm-intel**. Pour AMD : remplacer **kvm-intel** par **kvm-amd**. Vérification : **cat /sys/module/kvm_intel/parameters/nested** doit retourner **Y**.

Dans la configuration de la VM Proxmox, le type CPU doit exposer les extensions de virtualisation : **cpu: host** ou **cpu: kvm64,+vmx** (Intel) / **cpu: kvm64,+svm** (AMD). La performance de la nested virtualization est inférieure à la virtualisation native (overhead de 10-30%), ce qui la réserve aux environnements de test et développement. Pour l'optimisation globale des performances, consultez notre guide d'optimisation Proxmox VE. La documentation officielle Proxmox QEMU détaille tous les paramètres disponibles.

Paramètre	Valeur recommandée	Impact performance
Type CPU	host (cluster homogène)	+15-25% vs kvm64
Hugepages	2 Mo (VMs > 4 Go RAM)	+5-15% mémoire-intensive
Stockage	VirtIO SCSI	+200-400% vs IDE
Réseau	VirtIO Net	+10× vs émulation e1000
Agent QEMU	Activé (tous VMs)	Snapshots cohérents

Questions fréquentes

Comment activer et vérifier la nested virtualization dans Proxmox VE 9.1 ?

L'activation de la **nested virtualization** nécessite deux étapes : sur l'hôte Proxmox, activer le paramètre kernel avec **echo "options kvm-intel nested=Y" > /etc/modprobe.d/kvm-intel.conf** (ou kvm-amd pour AMD) et recharger le module KVM. Vérifier avec **cat /sys/module/kvm_intel/parameters/nested** qui doit retourner Y. Dans la VM Proxmox, configurer le type CPU en **host** ou ajouter le flag **+vmx** (Intel) ou **+svm** (AMD). Dans la VM invitée, vérifier que KVM est disponible avec **egrep -c '(vmx|svm)' /proc/cpuinfo** qui doit retourner une valeur > 0.

Pourquoi le CPU pinning améliore-t-il les performances des VMs dans Proxmox ?

Sans CPU pinning, le scheduler Linux migre librement les vCPUs d'une VM entre les cœurs physiques disponibles, en optimisant l'utilisation globale du serveur. Cette flexibilité a un coût : chaque migration de vCPU invalide les caches L1/L2 du processeur et peut provoquer des accès mémoire cross-NUMA. Le **CPU pinning** dédie des cœurs physiques spécifiques aux vCPU, maintenant les données en cache et assurant la localité NUMA. Pour les VMs temps réel ou haute performance (latence < 1ms, transactions financières, jeux en ligne), le pinning peut réduire la latence de 20-50%. La contrepartie est une réduction de la densité de VMs sur le nœud.

Quelle est la différence entre VirtIO SCSI et VirtIO Block dans Proxmox VE ?

VirtIO Block est le protocole de stockage paravirtualisé original : simple, efficace pour un disque unique, mais limité à 1 LUN par contrôleur. **VirtIO SCSI** (virtio-scsi-pci) est le successeur recommandé : il supporte jusqu'à 255 disques par contrôleur, les commandes SCSI complètes (TRIM/UNMAP pour la récupération d'espace sur le stockage thin-provisioned), le multiqueue I/O (meilleures performances sur VMs multi-cœurs) et l'éjection à chaud des disques. En pratique, utiliser VirtIO SCSI pour toutes les VMs de production, et VirtIO Block uniquement pour les compatibilités spéciales ou les cas d'usage simples avec un seul disque.

Sources et références : Proxmox VE Wiki · ANSSI

Articles connexes

- Hyper-V 2025 : Analyse Technique Approfondie et Securisation

- ESXi Hardening : Guide Complet de Sécurisation Avancée

Conclusion

La maîtrise des **paramètres avancés des VMs Proxmox VE 9.1** permet d'extraire le maximum des ressources physiques tout en garantissant l'isolation et la sécurité des workloads. CPU pinning, hugepages, VirtIO et nested virtualization sont les leviers clés d'une infrastructure virtualisée haute performance, à activer selon les besoins spécifiques de chaque VM.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr