

# Optimisation Proxmox - Guide Pratique Cybersecurite

Catégorie : Virtualisation    Lecture : 6 min    Publié le : 07/12/2025    Auteur : Ayi NEDJIMI

*optimisation Proxmox VE 9.0 : CPU, mémoire, stockage ZFS/Ceph, réseau, cluster HA. Commandes, recettes par workload et checklist complète. Guide...*

---

**Avertissement :** Les techniques présentées dans cet article sont destinées exclusivement à des fins éducatives et de tests autorisés. Toute utilisation malveillante est illégale et contraire à l'éthique professionnelle.

## Guide d'Optimisation Proxmox VE 9.0

---



**Par Ayi NEDJIMI**

Ce guide complet vous accompagne dans l'optimisation de votre infrastructure Proxmox VE 9.0. De la configuration système aux réglages avancés du cluster, découvrez les meilleures pratiques pour maximiser les performances, la stabilité et l'efficacité de votre environnement de virtualisation.

### Notre avis d'expert

La sécurité des hyperviseurs est le talon d'Achille de nombreuses infrastructures virtualisées. Une vulnérabilité d'évasion de VM peut compromettre l'ensemble de l'infrastructure en une seule exploitation. Le durcissement de l'hyperviseur doit être traité avec la même rigueur que celui du contrôleur de domaine.

Vos hyperviseurs sont-ils durcis selon les recommandations du CIS Benchmark ? optimisation Proxmox VE 9.0 : CPU, mémoire, stockage ZFS/Ceph, réseau, cluster HA. Commandes, recettes par workload et checklist complète. Guide... Ce guide couvre les aspects essentiels de optimisation proxmox guide pratique : méthodologie structurée, outils recommandés et retours d'expérience opérationnels. Les professionnels y trouveront des recommandations directement applicables.

## 1. Introduction

---

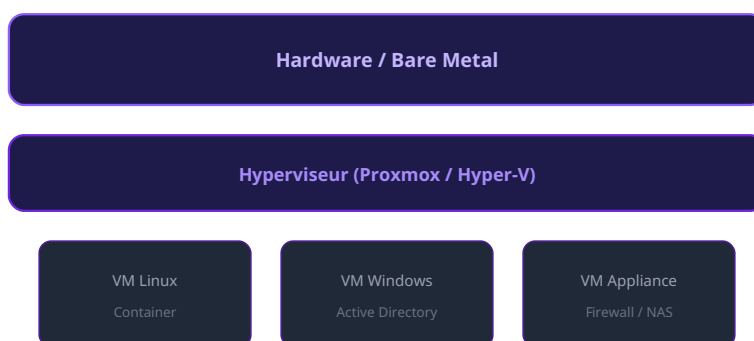
L'optimisation d'une infrastructure Proxmox VE 9.0 est essentielle pour garantir des performances optimales, une haute disponibilité et une utilisation efficace des ressources. Ce guide détaille les techniques d'optimisation à tous les niveaux : système hôte, machines virtuelles, conteneurs, stockage et réseau.

### ⚠ Avertissement Important

Les optimisations présentées dans ce guide doivent être testées dans un environnement de développement avant d'être appliquées en production. Chaque infrastructure a des besoins spécifiques qui peuvent nécessiter des ajustements personnalisés.

## Objectifs de l'Optimisation

- **Performance** : Maximiser les IOPS, réduire la latence, optimiser l'utilisation CPU/RAM
- **Stabilité** : Éviter les contentions de ressources et les points de défaillance
- **Efficacité** : Optimiser la densité de VMs/CTs par nœud
- **Scalabilité** : Préparer l'infrastructure pour la croissance future



Architecture de virtualisation multi-couches

## 2. Optimisation du Système Hôte (Host)

### 2.1. Tuning du Noyau Linux

Les paramètres du noyau jouent un rôle crucial dans les performances globales de l'hyperviseur. Pour approfondir, consultez [Evasion d'EDR/XDR : techniques](#).

#### Configuration Swappiness

```
# Réduire l'utilisation du swap (recommandé pour hosts avec beaucoup de RAM)
echo "vm.swappiness=10" >> /etc/sysctl.conf

# Pour environnements critiques avec RAM suffisante
echo "vm.swappiness=1" >> /etc/sysctl.conf

# Appliquer immédiatement
sysctl -p
```

#### 💡 Explication

**vm.swappiness** contrôle l'agressivité avec laquelle le noyau swap les pages mémoire. Une valeur de 10 favorise l'utilisation de la RAM physique, réduisant ainsi la latence liée au swap.

## Optimisation du Cache et de la Mémoire

```
# Augmenter le pourcentage de dirty pages avant flush
echo "vm.dirty_ratio=40" >> /etc/sysctl.conf
echo "vm.dirty_background_ratio=10" >> /etc/sysctl.conf

# Optimiser la libération de mémoire
echo "vm.vfs_cache_pressure=50" >> /etc/sysctl.conf

# Appliquer
sysctl -p
```

## Paramètres Réseau pour Host

```
# Augmenter les buffers réseau
echo "net.core.rmem_max=134217728" >> /etc/sysctl.conf
echo "net.core.wmem_max=134217728" >> /etc/sysctl.conf
echo "net.ipv4.tcp_rmem=4096 87380 67108864" >> /etc/sysctl.conf
echo "net.ipv4.tcp_wmem=4096 65536 67108864" >> /etc/sysctl.conf

# Optimiser la file d'attente
echo "net.core.netdev_max_backlog=30000" >> /etc/sysctl.conf

# Appliquer
sysctl -p
```

## 2.2. Configuration BIOS/UEFI

Paramètre BIOS	Valeur Recommandée	Impact
Intel VT-x / AMD-V	Activé	Virtualisation matérielle essentielle
Intel VT-d / AMD IOMMU	Activé	Passthrough PCIe (GPU, NIC)
C-States	Désactivé (prod)	Latence CPU prévisible
Turbo Boost	Activé	Performances CPU en burst
Hyper-Threading	Selon workload	Densité vs latence prévisible
NUMA	Activé	Optimisation mémoire multi-socket

## 2.3. Configuration du Scheduler CPU

```
# Vérifier le scheduler actuel
cat /sys/block/sda/queue/scheduler

# Pour SSD/NVMe, utiliser noop ou deadline
echo "noop" > /sys/block/nvme0n1/queue/scheduler

# Rendre permanent dans /etc/default/grub
GRUB_CMDLINE_LINUX_DEFAULT="quiet elevator=noop"

# Mettre à jour GRUB
update-grub
```

## Cas concret

L'exploitation de la vulnérabilité VMware ESXi CVE-2021-21974 par le ransomware ESXiArgs début 2023 a paralysé des milliers de serveurs de virtualisation dans le monde. L'attaque ciblait le service OpenSLP et rappelait l'importance critique de la mise à jour des hyperviseurs, souvent négligée par les équipes d'exploitation.

## 3. Optimisation CPU et Mémoire pour VM/CT

### 3.1. CPU Pinning (Épinglage CPU)

Le CPU pinning permet d'assigner des cœurs physiques spécifiques à des VMs critiques, réduisant la latence et améliorant la prévisibilité des performances.

#### Identifier la Topologie CPU

```
# Afficher la topologie complète
lscpu -e

# Ou avec numactl
numactl --hardware
```

#### Configuration du CPU Pinning

```
# Éditer la configuration de la VM
nano /etc/pve/qemu-server/100.conf

# Ajouter les lignes suivantes :
# Assigner les cœurs 0,1,2,3 à cette VM
affinity: 0,1,2,3

# Pour un épinglage NUMA-aware (serveurs multi-socket)
# VM sur NUMA node 0 avec cores 0-7
numa: 1
hostpci0: 0000:00:1f.2,numa=0
```

#### ⚠ Attention au Surengagement

Évitez de sur-allouer les cœurs physiques. Si vous avez 8 cœurs physiques, n'assignez pas plus de 8 vCPUs au total en épinglage strict. Laissez toujours 1-2 cœurs pour l'hyperviseur. Pour approfondir, consultez [RAG Architecture | Guide](#).

### 3.2. Type de CPU et Optimisation

Type CPU	Usage	Performances
host	Meilleures performances, pas de migration live	★★★★★
kvm64	Compatible migration, performances moyennes	★★★★
x86-64-v2-AES	Bon équilibre, support AES-NI	★★★★★
EPYC / Skylake-Server	Modèles spécifiques, migration dans même génération	★★★★★

```
# Configuration dans /etc/pve/qemu-server/100.conf
cpu: host

# Ou pour permettre la migration avec performances optimales
cpu: x86-64-v2-AES,flags=+aes
```

### 3.3. Gestion de la Mémoire

#### Ballooning

Le ballooning permet une allocation dynamique de la RAM entre les VMs.

```
# Activer le ballooning dans la VM (défaut : activé)
balloon: 2048

# Désactiver pour VMs critiques nécessitant RAM garantie
balloon: 0
```

#### 💡 Quand Désactiver le Ballooning ?

- Bases de données (PostgreSQL, MySQL, Oracle)
- Serveurs de cache (Redis, Memcached)
- Applications temps-réel

#### Huge Pages

```
# Configurer les huge pages sur l'hôte
echo "vm.nr_hugepages=1024" >> /etc/sysctl.conf
sysctl -p

# Dans la configuration VM
hugepages: 1024
```

#### KSM (Kernel Same-page Merging)

```
# Activer KSM pour économiser la RAM (VDI, environnements homogènes)
systemctl enable ksmtuned
systemctl start ksmtuned

# Vérifier l'état de KSM
cat /sys/kernel/mm/ksm/pages_shared
cat /sys/kernel/mm/ksm/pages_sharing
```

## 4. Optimisation du Stockage : ZFS et LVM

### 4.1. Optimisation ZFS

#### Configuration ARC (Adaptive Replacement Cache)

```
# Limiter l'ARC à 50% de la RAM disponible (serveur 64 GB = 32 GB ARC)
echo "options zfs zfs_arc_max=34359738368" > /etc/modprobe.d/zfs.conf

# Pour serveurs dédiés au stockage, augmenter à 75%
echo "options zfs zfs_arc_max=51539607552" > /etc/modprobe.d/zfs.conf

# Appliquer (nécessite reboot ou reload du module)
update-initramfs -u -k all
```

#### Optimisation des Pools ZFS

```
# Création d'un pool optimisé pour VMs
zpool create -o ashift=12 \
  -o atime=off \
  -o compression=lz4 \
  -o recordsize=16K \
  vmdata raidz2 sdb sdc sdd sde

# Pour bases de données (IOPS élevés)
zfs set recordsize=8K vmdata/postgres
zfs set primarycache=metadata vmdata/postgres
zfs set logbias=throughput vmdata/postgres

# Pour fichiers volumineux (vidéos, backups)
zfs set recordsize=1M vmdata/backups
zfs set compression=gzip-9 vmdata/backups
```

Paramètre	Valeur	Usage
ashift	12 (4K) / 13 (8K)	Alignement secteurs (4K pour SSD/NVMe)
atime	off	Désactiver mise à jour access time
compression	lz4	Compression rapide, peu d'overhead CPU
recordsize	16K (VM) / 8K (DB)	Taille des blocs ZFS
sync	standard / disabled	Synchronisation (disabled = risque perte données)

#### SLOG et L2ARC

```
# Ajouter un SLOG (ZIL) sur NVMe pour les écritures synchrones
zpool add vmdata log /dev/nvme0n1p1

# Ajouter un L2ARC pour étendre le cache de lecture
zpool add vmdata cache /dev/nvme0n1p2
```

## 4.2. Optimisation LVM

```
# Création d'un LV thin-provisionné optimisé
lvcreate -L 500G -T vg0/thinpool

# Ajuster le chunk size pour VMs (512K - 1M)
lvconvert --chunksize 1024K vg0/thinpool

# Créer des volumes pour VMs
lvcreate -V 100G -T vg0/thinpool -n vm-100-disk-0
```

### Tuning LVM pour Performances

```
# Éditer /etc/lvm/lvm.conf
issue_discards = 1 # Pour SSDs (TRIM)
use_lvmetad = 0    # Désactiver pour clusters

# Optimiser le read-ahead
blockdev --setra 8192 /dev/vg0/vm-100-disk-0
```

## 5. Optimisation du Stockage Distribué : Ceph

### 5.1. Configuration OSD

```
# Optimisation des OSDs dans /etc/ceph/ceph.conf
[osd]
osd_max_backfills = 1
osd_recovery_max_active = 1
osd_recovery_max_single_start = 1
osd_recovery_sleep_hdd = 0.1
osd_recovery_sleep_ssd = 0
osd_recovery_sleep_hybrid = 0.025

# Pour SSD/NVMe
osd_op_threads = 8
osd_disk_threads = 4
osd_journal_size = 10240
```

### 5.2. Pool Ceph Optimisé

```
# Créer un pool avec réplication 3 et PG optimisés
ceph osd pool create vmdata 128 128

# Définir la taille de réplication
ceph osd pool set vmdata size 3
ceph osd pool set vmdata min_size 2

# Activer les règles CRUSH pour distribution optimale
ceph osd pool set vmdata crush_rule replicated_rule

# Pour VMs critiques : configurer QoS
rbd config image set vmdata/vm-100-disk-0 rbd_qos_iops_limit 5000
rbd config image set vmdata/vm-100-disk-0 rbd_qos_bw_limit 500M
```

## 5.3. Client RBD

```
# Configuration client dans /etc/ceph/ceph.conf
[client]
rbd_cache = true
rbd_cache_size = 268435456 # 256 MB
rbd_cache_max_dirty = 201326592 # 192 MB
rbd_cache_target_dirty = 134217728 # 128 MB
rbd_cache_writethrough_until_flush = false

# Pour performances extrêmes (à tester prudemment)
rbd_readahead_trigger_requests = 10
rbd_readahead_max_bytes = 524288
```

## 6. Optimisation Réseau et SDN

---

### 6.1. Configuration des Bridges

```
# Éditer /etc/network/interfaces
auto vmbr0
iface vmbr0 inet static
    address 192.168.1.10/24
    gateway 192.168.1.1
    bridge-ports ens18
    bridge-stp off
    bridge-fd 0
    bridge-vlan-aware yes
    bridge-vids 2-4094
```

### 6.2. Bonding (Agrégation de Liens)

```
# LACP (802.3ad) pour redondance et performance
auto bond0
iface bond0 inet manual
    bond-slaves ens18 ens19
    bond-mode 802.3ad
    bond-miimon 100
    bond-downdelay 200
    bond-updelay 200
    bond-lacp-rate fast
    bond-xmit-hash-policy layer3+4

auto vmbr0
iface vmbr0 inet static
    address 192.168.1.10/24
    bridge-ports bond0
    bridge-stp off
    bridge-fd 0
```

### 6.3. Optimisation des Cartes Réseau (NIC)

```
# Activer les offloads matériels
ethtool -K ens18 gso on gro on tso on

# Augmenter la taille du ring buffer
ethtool -G ens18 rx 4096 tx 4096

# Activer le RSS (Receive Side Scaling) pour multi-cœurs
ethtool -L ens18 combined 4

# Configuration permanente dans /etc/network/interfaces
post-up /usr/sbin/ethtool -K ens18 gso on gro on tso on
post-up /usr/sbin/ethtool -G ens18 rx 4096 tx 4096
```

### 6.4. SDN Proxmox (Software Defined Networking)

```
# Créer une zone VXLAN pour isolation réseau
pvesh create /cluster/sdn/zones --zone vxlan-zone --type vxlan --peers
192.168.1.11,192.168.1.12

# Créer un VNet
pvesh create /cluster/sdn/vnets --vnet vnet100 --zone vxlan-zone --tag 100

# Créer un subnet
pvesh create /cluster/sdn/vnets/vnet100/subnets --subnet 10.100.0.0/24 --gateway
10.100.0.1

# Appliquer la configuration SDN
pvesh set /cluster/sdn
```

## 7. Optimisation du Cluster et Haute Disponibilité

### 7.1. Configuration Corosync

```
# Éditer /etc/pve/corosync.conf
totem {
    version: 2
    cluster_name: pve-cluster
    transport: knet
    token: 3000
    token_retransmits_before_loss_const: 10

    interface {
        linknumber: 0
        knet_link_priority: 1
    }

    # Lien redondant (recommandé)
    interface {
        linknumber: 1
        knet_link_priority: 0
    }
}

# Appliquer
systemctl restart corosync pve-cluster
```

### 7.2. Groupes HA et Priorités

```
# Créer un groupe HA
ha-manager groupadd production --nodes node1:2,node2:1,node3:1

# Ajouter une VM au groupe avec priorité maximale
ha-manager add vm:100 --group production --max_restart 3 --max_relocate 3

# Configurer l'état désiré
ha-manager set vm:100 --state started
```

### 7.3. Fencing et Watchdog

```
# Activer le watchdog matériel
modprobe softdog
echo "softdog" >> /etc/modules

# Configuration dans /etc/pve/datacenter.cfg
ha: shutdown_policy=conditional
watchdog: action=reboot
```

## 8. Monitoring et Ajustement Continu

### 8.1. Outils de Monitoring

```
# Installer les outils de diagnostic
apt install -y sysstat iotop htop nmon atop

# Activer la collecte sysstat
systemctl enable sysstat
systemctl start sysstat

# Analyser les performances I/O
iostat -xz 5

# Analyser les performances réseau
sar -n DEV 5

# Analyser les performances mémoire
vmstat 5
```

### 8.2. Métriques Clés à Surveiller

Métrique	Commande	Seuil d'Alerte
CPU Wait (iowait)	iostat	> 20%
RAM disponible	free -h	< 10%
Latence disque	iostat -x	> 10ms (SSD), > 50ms (HDD)
Network errors	ip -s link	> 0.1%
Corosync status	corosync-cfgtool -s	Token lost > 0

### 8.3. Intégration avec Prometheus/Grafana

```
# Installer l'exporteur Proxmox
apt install -y prometheus-pve-exporter

# Configuration dans /etc/prometheus/pve.yml
default:
  user: monitoring@pve
  password: your_secure_password
  verify_ssl: false

# Redémarrer le service
systemctl restart prometheus-pve-exporter
```

## 9. Table de Référence des Commandes

Catégorie	Commande	Description
<b>CPU</b>	<code>lscpu -e</code>	Afficher la topologie CPU
	<code>numactl --hardware</code>	Afficher les nœuds NUMA
<b>Mémoire</b>	<code>free -h</code>	Afficher l'utilisation mémoire
	<code>cat /sys/kernel/mm/ksm/pages_shared</code>	Vérifier KSM
<b>Stockage</b>	<code>zpool status</code>	État des pools ZFS
	<code>pvesm status</code>	État de tous les stockages
	<code>ceph -s</code>	État du cluster Ceph
<b>Réseau</b>	<code>ethtool ens18</code>	Info et stats de la NIC
	<code>brctl show</code>	Afficher les bridges
<b>Cluster</b>	<code>pvecm status</code>	État du cluster
	<code>ha-manager status</code>	État HA
<b>Performance</b>	<code>iostat -xz 5</code>	Statistiques I/O en temps réel
	<code>sar -n DEV 5</code>	Statistiques réseau

## 10. Recettes d'Optimisation par Workload

### 10.1. Base de Données (PostgreSQL, MySQL)

```
# Configuration VM
cores: 4
memory: 16384
balloon: 0
cpu: host
numa: 1

# Stockage
# LVM-thin ou ZFS avec recordsize=8K
zfs set recordsize=8K vmdata/db-vm
zfs set primarycache=metadata vmdata/db-vm
zfs set logbias=throughput vmdata/db-vm

# Disque dédié pour logs/WAL sur SSD rapide
scsil: local-lvm:vm-100-disk-1,iothread=1,cache=writeback,discard=on
```

## 10.2. VDI (Virtual Desktop Infrastructure)

```
# Configuration VM type (clonage linked-clone recommandé)
cores: 2
memory: 4096
balloon: 2048 # Ballooning activé pour densité

# Activer KSM sur l'hôte
systemctl enable ksmtuned

# Templates avec OS optimisé
# - Désactiver indexation
# - Désactiver services inutiles
# - Profil "thin" Windows
```

## 10.3. Conteneurs LXC (Micro-services)

```
# Template Debian/Ubuntu optimisé
pct create 200 local:vztmpl/debian-12-standard_12.0-1_amd64.tar.zst \
  --cores 2 \
  --memory 2048 \
  --swap 0 \
  --net0 name=eth0,bridge=vbr0,ip=dhcp \
  --storage local-lvm

# Pour applications critiques : désactiver swap
swap: 0

# Unprivileged containers (sécurité)
unprivileged: 1
```

# 11. Checklist Complète d'Optimisation

### ✓ Checklist Système Hôte

BIOS/UEFI : VT-x/AMD-V, VT-d/IOMMU activés

Kernel : Paramètres sysctl optimisés (swappiness, dirty\_ratio, réseau)

Scheduler : noop/deadline pour SSD/NVMe

Firmware : Mis à jour (serveur, RAID, NIC)

### ✓ Checklist CPU et Mémoire

CPU type : host ou modèle spécifique selon besoin migration

CPU pinning : Configuré pour VMs critiques

NUMA : Activé et VM alignées sur nœuds NUMA

Ballooning : Désactivé pour DB et apps critiques

KSM : Activé pour environnements VDI homogènes

### ✓ Checklist Stockage

ZFS : ARC limité, compression lz4, recordsize adapté

ZFS : SLOG (NVMe) ajouté pour workloads synchrones

LVM : Thin provisioning configuré avec chunk size optimal

Ceph : OSDs optimisés, PG calculés, QoS configuré

Cache : iothread activé pour disques virtio-scsi critiques

### ✓ Checklist Réseau

Bridges : STP désactivé, VLAN-aware configuré si nécessaire

Bonding : LACP configuré pour redondance

NIC : Offloads matériels activés (GSO, GRO, TSO)

MTU : Jumbo frames (9000) pour réseau stockage

SDN : Configuré si isolation réseau avancée requise

### ✓ Checklist Cluster et HA

Corosync : Double lien configuré et testé

HA : Groupes définis avec priorités appropriées

Fencing : Watchdog activé et testé

Quorum : Compris et configuré (QDevice si 2 nœuds)

### ✓ Checklist Monitoring

Outils : sysstat, iotop, atop installés

Métriques : CPU wait, RAM, latence I/O surveillés

Alertes : Configurées pour seuils critiques

Grafana : Dashboards Proxmox déployés

## 12. Ressources et Références

---

### Documentation Officielle

- **Proxmox VE Administration Guide** : Documentation complète de référence
- **Proxmox Wiki** : Guides communautaires et cas d'usage
- **Ceph Documentation** : Configuration avancée du stockage distribué
- **ZFS Documentation** : Optimisation et tuning des pools

### Rappels Importants

- Testez toutes les optimisations en environnement de développement avant production
- Documentez chaque modification pour faciliter le troubleshooting
- Surveillez les métriques après chaque changement pendant au moins 7 jours
- Conservez des backups avant toute modification système critique
- Planifiez une fenêtre de maintenance pour les changements majeurs

### Ressources open source associées :

- [HyperVIntrospector](#) — Introspection Hyper-V pour comparaison (C++)
- [awesome-cybersecurity-tools](#) — Liste curatée de 100+ outils de cybersécurité

## Questions frequentes

---

### Comment ce sujet impacte-t-il la securite des organisations ?

Ce sujet a un impact significatif sur la securite des organisations car il touche aux fondamentaux de la protection des systemes d'information. Les entreprises doivent evaluer leur exposition, mettre en place des mesures preventives adaptees et former leurs equipes pour faire face aux risques associes a cette problematique. Pour approfondir, consultez [Guide Complet Proxmox](#).

### Quelles sont les bonnes pratiques recommandees par les experts ?

Les experts recommandent une approche basee sur les risques, incluant l'evaluation reguliere de la posture de securite, la mise en place de controles techniques et organisationnels, la formation continue des equipes et l'adoption des referentiels de securite reconnus comme ceux du NIST, de l'ANSSI et de l'OWASP.

### Pourquoi est-il important de se former sur ce sujet en 2026 ?

En 2026, la maitrise de ce sujet est devenue incontournable face a l'evolution constante des menaces et des exigences reglementaires. Les professionnels de la cyberscurite doivent maintenir leurs competences a jour pour proteger efficacement les actifs numeriques de leur organisation et repondre aux obligations de conformite.

Pour approfondir, consultez les ressources officielles : OWASP Testing Guide, CVE Details et ANSSI.

**Sources et références :** [Proxmox VE Wiki](#) · [ANSSI](#)

## Conclusion

---

L'optimisation d'une infrastructure Proxmox VE 9.0 est un processus continu qui nécessite une compréhension approfondie des workloads, une surveillance régulière des métriques et des ajustements progressifs. En suivant ce guide et en adaptant les recommandations à votre contexte spécifique, vous pouvez atteindre des performances optimales tout en maintenant la stabilité et la fiabilité de votre environnement de virtualisation. Pour approfondir, consultez [ISO 27001:2022 - Guide Complet de Certification et Mise en Conformité](#).

Les gains de performance peuvent être significatifs : jusqu'à **+30% d'IOPS** avec les optimisations ZFS/Ceph, **-20% de latence CPU** avec le pinning et NUMA, et une **densité accrue de 15-25% VMs/nœud** avec le tuning mémoire.

---

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

[ayinedjimi-consultants.fr](https://ayinedjimi-consultants.fr) · [ayi@ayinedjimi-consultants.fr](mailto:ayi@ayinedjimi-consultants.fr)

© 2025 — Reproduction interdite sans autorisation.