

# Dimensionnement : Strategies de Detection et de Remediation

Catégorie : Virtualisation    Lecture : 13 min    Publié le : 07/12/2025    Auteur : Ayi NEDJIMI

*Guide complet et méthodique pour dimensionner votre infrastructure Proxmox VE 9.0 : CPU, RAM, stockage, réseau, cluster HA, avec 5 cas d. Guide.*

Dimensionnement : Strategies de Detection et de Remediation constitue un enjeu majeur pour les professionnels de la sécurité informatique et les équipes techniques. Ce guide détaillé sur dimensionnement proxmox propose une méthodologie structurée, des outils éprouvés et des recommandations opérationnelles directement applicables. L'objectif est de fournir aux praticiens — consultants, ingénieurs sécurité, administrateurs systèmes — les connaissances et les techniques nécessaires pour aborder ce sujet avec rigueur. Chaque section s'appuie sur des retours d'expérience terrain et intègre les évolutions les plus récentes du domaine. Les recommandations présentées sont adaptées aux environnements d'entreprise et tiennent compte des contraintes opérationnelles réelles.

**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 Complet de Dimensionnement Proxmox VE 9.0

Ce guide complet vous accompagne dans le dimensionnement méthodique de votre infrastructure Proxmox VE 9.0, de l'inventaire initial jusqu'à la mise en production, avec formules de calcul, exemples concrets et 5 cas d'usage détaillés.



**Par Ayi NEDJIMI**

Guide technique exhaustif sur le dimensionnement des infrastructures Proxmox VE 9.0, intégrant les dernières évolutions : snapshots LVM, SDN Fabrics, et règles HA améliorées. Basé sur Debian 13 "Trixie", kernel Linux 6.14.8, QEMU 10.0.2, et Ceph Squid 19.2.3.



Architecture de virtualisation multi-couches

Que se passerait-il si un attaquant s'échappait d'une de vos machines virtuelles ? Guide complet et méthodique pour dimensionner votre infrastructure Proxmox VE 9.0 : CPU, RAM, stockage, réseau, cluster HA, avec 5 cas d. Guide. Ce guide couvre les aspects essentiels de dimensionnement proxmox : méthodologie structurée, outils recommandés et retours d'expérience opérationnels. Les professionnels y trouveront des recommandations directement applicables.

## 1. Introduction à Proxmox VE 9.0

Proxmox Virtual Environment 9.0, sorti en août 2025, est basé sur Debian 13 "Trixie" et apporte des évolutions majeures pour l'infrastructure virtualisée.

Composant	Version	Évolution Majeure
Base OS	Debian 13 "Trixie"	Support matériel dernière génération
Kernel Linux	6.14.8	Performances I/O et architectures hybrides
QEMU	10.0.2	AVX-512, P-cores/E-cores optimisés
LXC	6.0.4	cgroupv2 obligatoire
ZFS	2.3.3	Performances et nouvelles fonctionnalités
Ceph	Squid 19.2.3	Efficacité espace et performances

### Nouveautés Critiques pour le Dimensionnement

#### Innovations Proxmox VE 9.0

- **Snapshots sur LVM thick-provisioned** : Création de snapshots sur volumes LVM partagés (iSCSI, Fibre Channel), impactant le dimensionnement du stockage et les stratégies de backup
- **SDN Fabrics** : Fabrics pour Software-Defined Networking permettant des architectures réseau complexes type spine-leaf à deux couches

- **cgroupv2 obligatoire** : Abandon complet de cgroupv1, nécessitant systemd 230+ pour les conteneurs, impact sur le dimensionnement des workloads legacy

### Notre avis d'expert

Les évasions de conteneurs représentent un risque croissant avec l'adoption massive de Docker et Kubernetes. Nos tests montrent que les configurations par défaut sont rarement suffisantes pour isoler efficacement les workloads. L'approche defense-in-depth est non négociable dans un environnement conteneurisé.

## 2. Principes Fondamentaux du Dimensionnement

---

### Méthodologie de Dimensionnement en 4 Phases

Le dimensionnement d'une infrastructure Proxmox VE 9.0 doit suivre une approche méthodique structurée :

#### Phase 1 : Inventaire des Workloads

- Identifier tous les services et applications à virtualiser
- Catégoriser par criticité (Tier 1, 2, 3)
- Évaluer les besoins de performance (IOPS, latence, bande passante)
- Déterminer les exigences de disponibilité (SLA)

#### Phase 2 : Calcul des Ressources Brutes

- Additionner les besoins individuels par workload
- Appliquer les facteurs de surallocation (overcommit ratios)
- Prévoir la croissance sur 3-5 ans
- Ajouter les marges de sécurité (overhead)

#### Phase 3 : Optimisation et Consolidation

- Regrouper les workloads par affinité
- Équilibrer les ressources entre nœuds
- Identifier les opportunités de mutualisation
- Optimiser les coûts vs performances

#### Phase 4 : Validation et Ajustement

- Effectuer des tests de charge
- Valider les scénarios de failover
- Ajuster selon les résultats observés
- Documenter les hypothèses et décisions

## Facteurs de Surallocation Recommandés

Ressource	Conservative	Équilibré	Agressif	Contexte
CPU (vCPU/pCore)	1:1	2:1	4:1	Workloads mixtes
RAM	0% overcommit	10-20%	30-50%	Avec ballooning actif
Stockage (thin)	+40% overhead	+30%	+25%	Snapshots inclus

### 3. Dimensionnement CPU

#### Architecture CPU et Considérations

Proxmox VE 9.0 avec QEMU 10.0.2 supporte les dernières fonctionnalités CPU, incluant les instructions AVX-512, les extensions de virtualisation améliorées et l'optimisation pour architectures hybrides (P-cores/E-cores).

#### Calcul du Nombre de Cœurs

##### 📌 Formule de Base

$$\text{Cœurs physiques requis} = (\sum \text{vCPU des VMs} \times \text{facteur utilisation}) / \text{ratio overcommit} + \text{cœurs système}$$

##### Détails des composants :

- **Cœurs système** : Réserver 2-4 cœurs pour Proxmox VE lui-même
- **Facteur utilisation** : Pourcentage moyen d'utilisation CPU attendu (30-60% typique)
- **Ratio overcommit** : Dépend du type de workload (voir tableau ci-dessus)

#### Types de Processeurs Recommandés

##### Pour Clusters Petits à Moyens (≤8 nœuds)

- **AMD EPYC 4004/8004 Series** : Excellent rapport performance/prix
- **Intel Xeon E-2400** : Alternative équilibrée
- **Minimum** : 8 cœurs / 16 threads par socket

##### Pour Clusters Moyens à Grands (8-32 nœuds)

- **AMD EPYC 9004 Series (Genoa/Bergamo)** : Jusqu'à 96/128 cœurs
- **Intel Xeon Scalable 4th/5th Gen** : Performances élevées
- **Recommandé** : 16-32 cœurs / 32-64 threads par socket

##### Pour Environnements HPC/Haute Densité

- **AMD EPYC Bergamo 9754 (128 cœurs)** : Densité maximale
- **Intel Xeon Max Series** : Performances extrêmes

- Configuration bi-socket pour évolutivité

## CPU Pinning et NUMA

Avec Proxmox VE 9.0, le CPU pinning est crucial pour les VMs exigeantes :

```
# Configuration NUMA-aware pour VM haute performance
numa: 1
cores: 8
sockets: 1
cpu: host
# Épingler les vCPUs sur des pCores spécifiques
hostpci0: 0000:00:1f.0,pcie=1
```

### ⚙️ Best Practices CPU Pinning

- Épingler les VMs critiques sur des cœurs physiques dédiés
- Respecter les frontières NUMA pour minimiser la latence
- Laisser 10-15% de cœurs libres pour l'ordonnanceur
- Utiliser `lstopo` pour visualiser la topologie NUMA

## Dimensionnement par Type de Workload

Type de Workload	Ratio Overcommit	Caractéristiques	Exemple
Compute-Intensive	1:1 à 1.5:1	Cœurs haute fréquence (≥3.0 GHz), CPU pinning obligatoire	2× EPYC 9654 (96c) → 192 vCPUs max
Équilibrés	2:1 à 3:1	Cœurs multiples fréquences moyennes, CPU type "host"	2× EPYC 9454P (48c) → 192-288 vCPUs
Légers	4:1 à 6:1	Densité maximale, CPU type "kvm64" acceptable	EPYC 9754 (128c) → 512-768 vCPUs

### Cas concret

En 2024, la vulnérabilité CVE-2024-21626 (Leaky Vessels) dans runc a démontré qu'une évvasion de conteneur Docker était possible via une manipulation du répertoire de travail. Cette faille affectait l'ensemble de l'écosystème de conteneurs et a nécessité des patches d'urgence sur toutes les plateformes Kubernetes majeures.

Vos conteneurs sont-ils réellement isolés les uns des autres ?

## 4. Dimensionnement Mémoire (RAM)

### Calcul de la Mémoire Totale

La mémoire est souvent la ressource la plus critique dans un environnement virtualisé. Proxmox VE 9.0 améliore le suivi de l'utilisation mémoire avec une meilleure visibilité sur l'overhead. Pour approfondir, consultez [NIS 2 : Guide Complet de la Directive Européenne sur la Cybersécurité](#).

#### Formule Globale

RAM totale = RAM VMs + RAM conteneurs + RAM système + RAM overhead + Marge sécurité

### Décomposition Détaillée

#### RAM Système (Proxmox VE)

- Base minimale : 4 GB
- Par TB de stockage ZFS : +1 GB (pour ARC)
- Par 10 VMs gérées : +512 MB
- Ceph OSD : +2-4 GB par OSD
- Exemple : Serveur avec 20 VMs, 10 TB ZFS, 4 OSDs = 4 + 10 + 1 + 12 = 27 GB

#### RAM VMs et Conteneurs

- Additionner les allocations de chaque VM/CT
- Ne PAS compter sur l'overcommit pour les environnements critiques
- Utiliser le ballooning uniquement comme optimisation secondaire

#### Overhead Mémoire Proxmox VE 9.0

Avec la version 9.0, Proxmox comptabilise mieux l'overhead mémoire :

- Overhead QEMU par VM : ~150-300 MB (dépend de la config)
- Tables de pages étendues : ~0.5-2% de la RAM VM
- Buffers réseau virtio : ~50-100 MB par VM
- Total typique : 3-5% de la RAM totale allouée

### Configuration Mémoire Optimale

Taille Cluster	RAM Minimum	RAM Recommandée	RAM Idéale
Petit (2-4 nœuds)	64 GB	128 GB	256 GB
Moyen (5-8 nœuds)	128 GB	256 GB	512 GB
Grand (>8 nœuds)	256 GB	512 GB	1+ TB

## ⚙️ Configuration DIMM Recommandée

- Privilégier 8 canaux mémoire remplis pour performance maximale
- Utiliser des barrettes identiques (capacité, vitesse, fabricant)
- DDR5 avec ECC obligatoire pour production
- Vitesse recommandée : DDR5-4800 minimum, DDR5-5600+ optimal

## Ballooning et KSM

### Ballooning Device (recommandé activé)

```
balloon: 2048 # Minimum garanti
```

- Permet à Proxmox de récupérer la RAM inutilisée
- Nécessite les pilotes virtio installés dans le guest
- Réduction typique : 10-30% de RAM totale

### Kernel Samepage Merging (KSM)

```
# Activer KSM globalement  
echo 1 > /sys/kernel/mm/ksm/run
```

- Fusionne les pages mémoire identiques
- Efficace avec VMs similaires (VDI, clusters identiques)
- Gains typiques : 15-40% avec VMs Windows identiques

## Dimensionnement par Use Case

### Environnement VDI (50-100 postes virtuels)

- 50 VMs × 4 GB RAM = 200 GB
- Overhead QEMU (5%) = 10 GB
- Système + ZFS = 15 GB
- KSM gain (-30%) = -60 GB
- TOTAL : 165 GB → Serveur 256 GB recommandé

### Cluster Bases de Données (PostgreSQL/MySQL)

- 3 VMs DB × 64 GB = 192 GB
- Overhead = 10 GB
- Système = 8 GB
- Marge 20% = 42 GB
- TOTAL : 252 GB → Serveur 384-512 GB recommandé

## Infrastructure Kubernetes (worker nodes)

- 10 VMs × 32 GB = 320 GB
- Overhead = 16 GB
- Système = 10 GB
- Marge 15% = 52 GB
- TOTAL : 398 GB → Serveur 512 GB recommandé

## 5. Dimensionnement Stockage

### Nouvelles Capacités Proxmox VE 9.0

#### Snapshots sur LVM Thick-Provisioned

Cette fonctionnalité bouleversant permet maintenant :

- Snapshots sur volumes LVM partagés (iSCSI/FC SAN)
- Chaînes de snapshots avec gestion automatique
- Support des LUNs de stockage externe

#### Impact sur le dimensionnement

Espace requis = Données + (Snapshots × Taux de modification × Durée rétention)

#### Exemple SAN iSCSI :

- Volume VM : 500 GB
- Taux modification : 10% par jour
- Rétention : 7 snapshots
- Espace requis :  $500 + (500 \times 0.10 \times 7) = 850$  GB
- Avec marge 20% : 1020 GB par VM

### Choix de la Technologie de Stockage

#### Stockage Local : ZFS (Recommandé par défaut)

Avec ZFS 2.3.3 dans Proxmox VE 9.0, configuration optimale :

```
# Pool RAIDZ2 pour équilibre performance/protection
zpool create -o ashift=12 vmpool raidz2 sda sdb sdc sdd sde sdf

# Paramètres recommandés
zfs set compression=lz4 vmpool
zfs set atime=off vmpool
zfs set recordsize=64K vmpool          # Pour VMs
zfs set recordsize=1M vmpool/backups  # Pour backups
```

#### Dimensionnement ZFS

- ARC : 1 GB par TB de stockage (max 50% RAM système)
- L2ARC (optionnel) : SSD 10-20% de la capacité HDD
- SLOG (recommandé) : 2× SSD NVMe 32-64 GB en miroir

- **Overhead** : Compter 25-30% d'espace pour métadonnées, copies, snapshots

### 💡 Exemple Configuration ZFS

Serveur avec 12× 4TB HDD, 256 GB RAM :

- Configuration : RAIDZ2 (10 data + 2 parity)
- Capacité brute : 48 TB
- Capacité utilisable : ~32 TB (after RAIDZ2)
- ARC requis : 32 GB RAM
- SLOG : 2× 64 GB NVMe en miroir
- Snapshots overhead : 20%
- **Capacité réelle pour VMs : ~25 TB**

### Stockage Local : LVM

Avec le support des snapshots en chaîne dans Proxmox VE 9.0 : Pour approfondir, consultez [Hyper-V 2025](#)

```
# Création volume group optimisé
vgcreate vg_vms /dev/sdb /dev/sdc
lvcreate -L 500G -n vm-100-disk-0 vg_vms

# Activer snapshots en chaîne (nouveau PVE 9.0)
pvesm set storage-name --content images,rootdir --snapshot-mode chain
```

- **Overhead snapshots** : 15-25% du volume
- Pas d'overhead métadonnées significatif
- Performance excellente (surtout avec thin-provisioning)

### Stockage Partagé : Ceph

Ceph Squid 19.2.3 dans Proxmox VE 9.0 apporte des améliorations significatives.

### 📏 Dimensionnement Ceph Cluster

$$\text{OSDs requis} = \frac{(\text{Capacité totale} \times \text{Facteur réplication})}{\text{Capacité par OSD} / \text{Taux remplissage}}$$

### Exemple :

- Besoin : 50 TB utilisables
- Réplication : 3× (size=3, min\_size=2)
- Capacité OSD : 8 TB
- Taux remplissage max : 75%
- OSDs requis :  $(50 \times 3) / 8 / 0.75 = 25$  OSDs
- Nœuds (3+ recommandé) : 9 nœuds × 3 OSDs = 27 OSDs

## Configuration Ceph pour Proxmox VE 9.0

```
[global]
osd pool default size = 3
osd pool default min size = 2
osd pool default pg num = 128
osd pool default pgp num = 128

# Performance optimizations
osd op threads = 8
osd disk threads = 4
osd recovery max active = 3
```

## Dimensionnement RAM pour Ceph

- **OSD HDD** : 4 GB RAM par OSD minimum
- **OSD SSD** : 5 GB RAM par OSD recommandé
- **OSD NVMe** : 6-8 GB RAM par OSD pour performance maximale
- **Monitors** : 2 GB RAM par MON
- **Managers** : 2 GB RAM par MGR

## Tiering Stockage et Performances

Tier	Technologie	IOPS	Latence	Use Case
Tier 0	NVMe local (RAID1/10)	100,000+ IOPS	<100 µs	BD critiques, VDI, latence-sensitive
Tier 1	SSD SATA/SAS ou Ceph NVMe	10,000-50,000 IOPS	<1 ms	VMs production, conteneurs
Tier 2	HDD SATA/SAS ou Ceph HDD	500-2,000 IOPS	5-15 ms	Archivage, backups, données froides

## Calcul IOPS et Throughput

### 📌 Formule IOPS Totales

IOPS utilisables = (IOPS par disque × Nombre disques) / Pénalité RAID / Facteur write

### Exemple RAID10 :

- 4x SSD à 50,000 IOPS chacun
- RAID10 (2 miroirs stripés)
- Write penalty : 2 (RAID1)
- Mix lecture/écriture : 70/30
- IOPS lecture :  $(50,000 \times 4) / 1 \times 0.7 = 140,000$
- IOPS écriture :  $(50,000 \times 4) / 2 \times 0.3 = 30,000$
- TOTAL utilisable : ~170,000 IOPS

## 6. Dimensionnement Réseau

### Nouveautés SDN Fabric dans Proxmox VE 9.0

L'introduction des SDN Fabric change l'architecture réseau en permettant des topologies complexes routées avec haute disponibilité.

### Architecture Réseau Recommandée

#### Séparation des Réseaux (minimum 3 VLANs distincts)

Réseau	Vitesse	Redondance	Usage
Management	1-10 GbE	LACP ou active-backup	Accès Web UI, SSH, API
VM/CT	10-25 GbE minimum	LACP obligatoire	Trafic des machines virtuelles
Stockage	25-100 GbE recommandé	Multiple paths	Ceph, iSCSI, NFS (MTU 9000)
Corosync	10 GbE minimum	Lien dédié	Communication cluster (<2ms)

#### Configuration Réseau Type

```
Nœud Proxmox VE standard :  
├─ 2× 10GbE (LACP) → Management + Corosync + Migration  
├─ 2× 25GbE (LACP) → VMs/CTs (trunked VLANs)  
└─ 2× 25/100GbE → Stockage Ceph (avec jumbo frames)
```

#### Configuration /etc/network/interfaces

```
auto bond0  
iface bond0 inet manual  
    bond-slaves eno1 eno2  
    bond-mode 802.3ad  
    bond-miimon 100  
    bond-xmit-hash-policy layer3+4  
  
auto vbr0  
iface vbr0 inet static  
    address 10.0.0.10/24  
    gateway 10.0.0.1  
    bridge-ports bond0  
    bridge-stp off  
    bridge-fd 0  
    bridge-vlan-aware yes  
    bridge-vids 2-4094
```

## SDN Fabrics : Configuration Spine-Leaf

Proxmox VE 9.0 permet de configurer des fabrics OpenFabric ou OSPF pour une architecture spine-leaf :

```
Spine Layer (2-4 switches) :
├─ 2× 100GbE vers chaque Leaf
└─ Routing OSPF/BGP

Leaf Layer (par rack/pod) :
├─ Connexion aux Spines
└─ Connexion aux nœuds Proxmox

Configuration SDN Fabric :
datacenter → SDN → Zones → Add → Fabric
- Type : OpenFabric
- Nodes : node1,node2,node3,node4
- Peers : auto-discover
- MTU : 9000
```

### ✓ Bénéfices des Fabrics

- Failover automatique multi-path
- Load balancing ECMP
- Scalabilité horizontale
- Résistance aux pannes de liens

## Dimensionnement Bande Passante

### 📐 Formule Bande Passante Réseau

BP requise = (Trafic VMs + Trafic stockage + Trafic migration + Overhead) × Facteur pics Pour approfondir, consultez [Top 10 des Attaques](#).

### Exemple cluster 8 nœuds :

- ```
- 200 VMs × 100 Mbps moyen = 20 Gbps
- Stockage Ceph : 15 Gbps
- Migrations simultanées (2×) : 10 Gbps
- Overhead (20%) : 9 Gbps
- TOTAL : 54 Gbps
- Avec pics (×1.5) : 81 Gbps
→ Recommandation : 2× 100 GbE par nœud minimum
```

## Configuration OVS vs Linux Bridge

| Solution     | Avantages                                                        | Use Case         |
|--------------|------------------------------------------------------------------|------------------|
| Linux Bridge | Performance excellente, configuration simple, support VLAN natif | PME/Standard     |
| Open vSwitch | Support SDN Fabrics complet, QoS avancé, tunneling VXLAN/GRE     | Datacenter/Cloud |

## 7. Architecture Cluster et Haute Disponibilité

### Taille Optimale du Cluster

| Taille | Nœuds       | Caractéristiques                                                        | Use Case                          |
|--------|-------------|-------------------------------------------------------------------------|-----------------------------------|
| Petit  | 2-4 nœuds   | Quorum 3 nœuds minimum, QDevice si 2 nœuds, Storage local ou petit Ceph | PME, lab, succursale              |
| Moyen  | 5-16 nœuds  | Configuration optimale, Ceph performant 5+ nœuds, bonne résilience      | Entreprise standard, hébergeur    |
| Grand  | 17-32 nœuds | Gestion complexe, segmentation en pods, Ceph haute performance          | Cloud provider, grande entreprise |

### ⚠ Limite Maximale

32 nœuds par cluster dans Proxmox VE 9.0

### Configuration Haute Disponibilité

#### HA Rules (nouveau dans PVE 9.0)

Les nouvelles règles d'affinité permettent un contrôle précis :

```
# Règle d'affinité nœud (garder VM sur nœuds spécifiques)
pvsh create /cluster/ha/groups --group db-group --nodes node1,node2,node3 --
comment "DB dedicated nodes"

# Règle d'anti-affinité (séparer VMs critiques)
pvsh create /cluster/ha/resources --sid vm:100 --group db-group --max_relocate
1

# Affinité resource-to-resource (VMs ensemble)
pvsh create /cluster/ha/resources --sid vm:101 --group db-group --depends
vm:100
```

#### Use Cases HA Rules

- **Licensing** : Contraindre VMs sur nœuds licenciés
- **Hardware** : VMs GPU sur nœuds avec GPU
- **Anti-affinité** : Séparer les réplicas applicatifs
- **Affinité** : Garder ensemble app + DB pour latence

## Dimensionnement pour HA

### Formule N+1 (minimum recommandé)

$$\text{Capacité cluster} = \text{Capacité utile} \times (N / (N-1))$$

### Exemple 4 nœuds :

- Chaque nœud : 64 vCPU, 256 GB RAM
- Capacité totale : 256 vCPU, 1024 GB RAM
- Avec N+1 : 192 vCPU, 768 GB RAM utilisables
- Perte 1 nœud : Reste 75% capacité

### Formule N+2 (haute résilience)

$$\text{Capacité utile} = \text{Capacité totale} \times ((N-2)/N)$$

### Exemple 6 nœuds :

- Capacité utile : 67% (4/6)
- Survit à 2 pannes simultanées

## Dimensionnement Corosync

Le réseau Corosync est critique dans Proxmox VE 9.0 : Pour approfondir, consultez [NTP Proxmox](#)

### ⚙ Requirements Corosync

- **Latence max** : 2 ms (intra-DC), 10 ms (metro-cluster)
- **Bande passante** : 10 Mbps minimum par nœud
- **MTU** : 1500 (standard) ou 9000 (avec jumbo frames)
- **Lien dédié recommandé** (pas de partage avec storage/VMs)

## Configuration Redondante

```
# /etc/corosync/corosync.conf
totem {
  version: 2
  cluster_name: pve-cluster
  transport: knet
  crypto_cipher: aes256
  crypto_hash: sha256

  interface {
    linknumber: 0
    knet_link_priority: 20
  }

  interface {
    linknumber: 1
    knet_link_priority: 10
  }
}
```

## 8. Best Practices Avancées

---

### Optimisations Performance

#### VirtIO et Drivers - Configuration Optimale VM

```
# Pour performance maximale
cpu: host
balloon: 2048
scsihw: virtio-scsi-pci
net0: virtio,bridge=vibr0,firewall=0
scsi0: local-lvm:vm-100-disk-0,cache=writeback,discard=on,iotthread=1

# Options avancées
args: -cpu host,+pdpelgb
numa: 1
```

#### Tuning Disque

- **iothread** : Activer pour I/O intensives
- **cache=writeback** : Meilleure performance (avec UPS)
- **discard=on** : Support TRIM/UNMAP
- **aio=native** : I/O asynchrones

## Tuning ZFS

```
# ARC optimisations
echo "$((32 * 1024 * 1024 * 1024))" > /sys/module/zfs/parameters/zfs_arc_max

# Tuning async write
zfs set sync=disabled vmpool      # ATTENTION : Risque perte données
zfs set logbias=throughput vmpool
zfs set primarycache=metadata vmpool

# Compression
zfs set compression=lz4 vmpool    # Meilleur ratio perf/gain
# ou
zfs set compression=zstd vmpool  # Meilleure compression
```

## Tuning Ceph

```
# Client optimizations
rbd cache = true
rbd cache size = 268435456      # 256 MB
rbd cache max dirty = 201326592 # 192 MB

# Network tuning
ms_async_op_threads = 5
ms_async_max_op_threads = 10

# OSD optimizations
osd_max_backfills = 1
osd_recovery_max_active = 3
```

## Sécurité et Isolation

### Isolation Réseau

```
# Configurer firewall Proxmox
pveum role modify PVEAuditor -privs VM.Audit

# Créer Security Groups
pvesh create /cluster/firewall/groups --group web-servers
pvesh set /cluster/firewall/groups/web-servers --rule "IN ACCEPT -p tcp -dport 80"
pvesh set /cluster/firewall/groups/web-servers --rule "IN ACCEPT -p tcp -dport 443"
```

## Backup et Disaster Recovery

### Stratégie de Backup 3-2-1

- 3 copies des données
- 2 supports différents
- 1 copie off-site

## Configuration Backup Proxmox

```
# Backup complet quotidien
vzdump --mode snapshot --compress zstd --storage backup-store --all 1

# Backup incrémental (avec PBS)
proxmox-backup-client backup vm/100 --repository user@pbs:backup-datastore
```

## Dimensionnement Backup Storage

Capacité backup = (Taille VMs × Nombre rétention × Taux changement) / Ratio compression

### Exemple :

- 50 VMs × 100 GB moyen = 5 TB
- Rétention : 30 jours × daily = 30 backups
- Taux changement : 10% par jour
- Compression zstd : 2:1
- Capacité :  $(5 \times 30 \times 0.10) / 2 = 7.5$  TB par backup set
- Avec 2 sets (production + off-site) : 15 TB

## Monitoring et Capacité

### Métriques Critiques

| Niveau         | Métrique             | Seuil Recommandé               |
|----------------|----------------------|--------------------------------|
| Nœud Proxmox   | CPU usage            | < 80% en moyenne               |
|                | RAM usage            | < 85% (laisser pour ARC/cache) |
|                | IOPS disk            | < 70% capacité maximum         |
|                | Network bandwidth    | < 60% capacité                 |
| VMs/Conteneurs | CPU ready time       | < 5%                           |
|                | RAM ballooned        | < 20%                          |
|                | Disk latency         | < 20 ms                        |
|                | Network drop packets | < 0.1%                         |
| Cluster        | Corosync latency     | < 2 ms                         |
|                | Ceph health          | HEALTH_OK                      |
|                | HA relocate time     | < 2 minutes                    |
|                | Backup success rate  | > 98%                          |

## Outils de Monitoring

```
# Installer exporters
apt install prometheus-pve-exporter

# Configuration datasource
curl -X POST http://grafana:3000/api/datasources \\\
  -H "Content-Type: application/json" \\\
  -d '{"name":"Proxmox","type":"prometheus","url":"http://localhost:9090"}'
```



### Dashboards Recommandés

- PVE Cluster Overview
- Node Resource Utilization
- VM/CT Performance Details
- Ceph Cluster Health
- Network Traffic Analysis

## 9. Use Cases Détaillés

---

### Use Case 1 : PME (50-100 Employés)

#### Besoins

- 30 VMs (serveurs métiers)
- 20 postes VDI
- 10 conteneurs (dev/test)
- Disponibilité 99.5% (43h downtime/an)

#### Architecture : Cluster 3 nœuds

```
Nœud Standard ×3 :
├─ CPU : 2× AMD EPYC 4124P (4c/8t) = 8c/16t par nœud
├─ RAM : 128 GB DDR5 ECC
├─ Stockage OS : 2× 500GB NVMe (ZFS mirror)
├─ Stockage VMs : 6× 2TB SSD SATA (RAIDZ1)
├─ Réseau : 2× 10GbE (LACP management+VMs)
│           2× 10GbE (Ceph + Corosync)
└─ Coût : ~8,000€ par nœud
```

```
Stockage Partagé Ceph :
├─ 3 nœuds × 3 OSDs (2TB SSD) = 9 OSDs
├─ Réplication : 3× (size=3, min_size=2)
├─ Capacité utilisable : ~4 TB
└─ Performance : ~30,000 IOPS, 3 GB/s
```

```
Coût Total : ~30,000€ (hardware seulement)
```

## Dimensionnement Détaillé

vCPUs totaux : 3 nœuds × 16 threads = 48 threads  
Avec N+1 : 32 vCPUs utilisables  
Overcommit 2:1 : 64 vCPUs disponibles  
Allocation : 30 VMs (2 vCPU) + 20 VDI (2 vCPU) + 10 CT (0.5 vCPU) = 65 vCPU ✓

RAM totale : 3 × 128 GB = 384 GB  
Avec N+1 : 256 GB utilisables  
Allocation : 30 VMs (4 GB) + 20 VDI (4 GB) + 10 CT (2 GB) = 220 GB ✓

## Use Case 2 : Hébergeur Web (500+ Sites)

### Besoins

- 200 VMs clients (mutualisé)
- 50 VMs dédiées (haute performance)
- Disponibilité 99,95% (4,4h downtime/an)
- Isolation stricte entre clients

### Architecture : Cluster 8 nœuds (2 pods de 4)

Nœud Performance ×4 (pod 1 - VMs dédiées) :

- ├ CPU : 2× AMD EPYC 9354 (32c/64t) = 64c/128t
- ├ RAM : 512 GB DDR5
- ├ OS : 2× 1TB NVMe (RAID1)
- ├ VM Tier0 : 4× 4TB NVMe (RAID10)
- ├ VM Tier1 : 8× 4TB SSD (RAIDZ2)
- ├ Réseau : 2× 25GbE management  
| 2× 100GbE VMs/Ceph
- └ Coût : ~25,000€ par nœud

Nœud Capacité ×4 (pod 2 - VMs mutualisées) :

- ├ CPU : 2× AMD EPYC 9254 (24c/48t) = 48c/96t
- ├ RAM : 256 GB DDR5
- ├ Stockage VM : Ceph pool partagé
- ├ Réseau : 2× 25GbE
- └ Coût : ~15,000€ par nœud

Ceph Cluster :

- ├ 8 nœuds × 4 OSDs (4TB NVMe) = 32 OSDs
- ├ Réplication 3× : ~40 TB utilisables
- ├ Pool SSD haute perf : 16 OSDs (20 TB)
- ├ Pool capacité : 16 OSDs (20 TB)
- └ Performance : ~500,000 IOPS

Coût Total : ~160,000€

## Dimensionnement

Pod 1 (Performance) : 256 threads → 50 VMs × 4-8 vCPU  
Pod 2 (Capacité) : 192 threads × 4:1 = 768 vCPU → 200 VMs × 2-4 vCPU

## Use Case 3 : Infrastructure Kubernetes

### Besoins

- 30 worker nodes K8s (VMs)
- 3 control plane (HA)
- 200+ pods par worker
- Autoscaling horizontal

### Architecture : Cluster 6 nœuds

Nœud Standard ×6 :

- ├ CPU : 2× AMD EPYC 9454P (48c/96t) = 96c/192t
- ├ RAM : 512 GB DDR5
- ├ OS : 2× 1TB NVMe
- ├ Ceph OSDs : 3× 4TB NVMe par nœud
- ├ Réseau : 2× 25GbE management
- |           2× 100GbE Ceph/Storage network
- └ Coût : ~30,000€ par nœud

Configuration K8s :

- ├ 3 VMs Control Plane : 8 vCPU, 16 GB RAM
- ├ 30 VMs Workers : 16 vCPU, 64 GB RAM
- └ Stockage : Ceph RBD avec CSI driver

Ceph Cluster :

- ├ 18 OSDs NVMe (6 nœuds × 3)
- ├ Pool replica 3 : ~24 TB utilisables
- ├ PVCs dynamiques pour pods
- └ Performance : 800,000+ IOPS

Coût Total : ~180,000€

### Dimensionnement

vCPUs :  $6 \times 192 = 1152$  threads total  
K8s allocation :  $(3 \times 8) + (30 \times 16) = 504$  vCPU  
Ratio : 1:2.3 (conservateur pour K8s)  
Marge scaling : 50% restant pour autoscaling

## Use Case 4 : VDI Entreprise (500 Postes)

### Besoins

- 500 postes virtuels Windows 10/11
- Sessions simultanées : 80% (400 postes)
- Profil utilisateur : Office + Web
- Expérience utilisateur fluide

## Architecture : Cluster 12 nœuds VDI optimisé

### Nœud VDI ×12 :

- ├ CPU : 2× AMD EPYC 9354P (32c/64t) = 64c/128t
- ├ RAM : 768 GB DDR5 (avec KSM actif)
- ├ OS : 2× 1TB NVMe
- ├ Stockage VDI : Ceph pool SSD dédié
- ├ Réseau : 2× 25GbE management
  - ├ 2× 50GbE VDI network (low latency)
  - ├ 2× 100GbE Ceph backend
- ├ GPU : NVIDIA A16 (4 GPUs) pour vGPU optionnel
- └ Coût : ~35,000€ par nœud

### Ceph VDI Pool :

- ├ 24 OSDs SSD (12 nœuds × 2 OSDs 4TB)
- ├ Replica 3 : ~32 TB utilisables
- ├ Pool dédié VDI (persistent desktops)
- └ Non-persistent : Linked clones sur local NVMe

### Template VDI :

- ├ Windows 10 LTSC : 80 GB
- ├ Avec VirtIO drivers et QEMU guest agent
- ├ KSM-friendly (pages mémoire optimisées)
- └ 500 linked clones : ~200 GB storage overhead

### Configuration VM VDI :

- ├ vCPU : 2-4 cores (selon profil utilisateur)
- ├ RAM : 4-8 GB (avec ballooning)
- ├ Disk : 80 GB (thin-provisioned)
- ├ vGPU : 1/4 GPU (pour CAO/graphisme)
- └ Network : VirtIO avec multiqueue

Coût Total : ~450,000€

## Dimensionnement

- 400 sessions simultanées × 4 vCPU = 1600 vCPU
- 12 nœuds × 128 threads = 1536 threads
- Overcommit 2:1 = 3072 vCPU disponibles ✓
  
- RAM : 400 × 6 GB moyenne = 2400 GB
- Avec KSM (-30%) : 1680 GB requis
- 12 × 768 GB = 9216 GB total
- Avec N+1 : 8448 GB utilisables ✓

## Use Case 5 : Cloud Gaming Platform

### Besoins

- 100 instances gaming simultanées
- GPU dédiée par instance
- Latence <20ms (streaming)
- 4K 60fps support

## Points d'attention spécifiques

### Architecture : Cluster 10 nœuds Gaming

#### Nœud Gaming ×10 :

- └─ CPU : AMD EPYC 9654 (96c/192t)
- └─ RAM : 1.5 TB DDR5
- └─ OS : 2× 2TB NVMe RAID1
- └─ VM Storage : 8× 8TB NVMe (RAID10) local
- └─ GPU : 4× NVIDIA L40S (48GB chacun)
- └─ Réseau : 2× 100GbE ultra-low latency
- └─ Coût : ~80,000€ par nœud

#### Configuration Gaming VM :

- └─ vCPU : 8-16 cores (pinned)
- └─ RAM : 32-64 GB (non-swappable)
- └─ GPU : 1 GPU passthrough complet
- └─ Disk : 500 GB NVMe local
- └─ Network : SR-IOV pour latence minimale
- └─ vBIOS : GPU mode optimal gaming

#### Optimisations :

- └─ Huge pages : 1 GB pages pour RAM
- └─ CPU pinning : Cores physiques dédiés
- └─ NUMA pinning : Local memory access
- └─ Kernel tuning : Realtime scheduler
- └─ Network : Bypass stack avec DPDK

#### Stockage Games :

- └─ NFS ultra-rapide pour bibliothèque games
- └─ 200 TB capacité (50+ games AAA)
- └─ 4× 100GbE aggregated vers storage array
- └─ Lecture seule, monté sur tous les nœuds

Coût Total : ~850,000€ (sans games storage)

### Dimensionnement

- 10 instances × 10 nœuds = 100 concurrentes max
- vCPU : 100 × 12 moyen = 1200 vCPU
- Ratio : 1:1.6 (proche bare-metal)
- GPU : 40 GPUs total = 100 instances (répartition)

## 10. Checklist de Dimensionnement

### Phase Préparatoire

#### Collecte d'Information

- Inventaire complet des workloads existants
- Analyse des pics d'utilisation (CPU, RAM, IO)
- Identification des dépendances applicatives

- Évaluation des besoins de disponibilité (SLA)
- Budget disponible et contraintes temporelles
- Compétences équipe technique

### Exigences Fonctionnelles

- Nombre de VMs/conteneurs prévu
- Types de workloads (DB, Web, VDI, etc.)
- Besoins de stockage (capacité, IOPS, latence)
- Exigences réseau (bande passante, segmentation)
- Stratégie backup et rétention
- Plans de croissance 3-5 ans

## Dimensionnement Technique

### Calcul CPU

- Nombre de vCPUs total requis calculé
- Ratio overcommit déterminé (1:1 à 4:1)
- Type de processeur sélectionné (AMD/Intel)
- Configuration socket déterminée (single/dual)
- CPU pinning planifié pour VMs critiques
- Overhead système pris en compte (2-4 cores)

### Calcul RAM

- Mémoire totale VMs/CTs additionnée
- Overhead QEMU calculé (~5%)
- RAM système Proxmox prévue (selon stockage)
- Ballooning configuré
- KSM évalué (si VMs similaires)
- Marge sécurité 15-20% ajoutée

### Calcul Stockage

- Capacité brute requise calculée
- Technologie choisie (ZFS/LVM/Ceph)
- RAID/réplication niveau défini
- IOPS requis évalués
- Tiering défini (NVMe/SSD/HDD)
- Overhead snapshots/métadonnées compté
- Stratégie backup dimensionnée

### Réseau

- Bande passante totale calculée

- Séparation VLANs planifiée (min 3)
- Redondance configurée (LACP/active-backup)
- SDN Fabrics évalué (si >8 nœuds)
- Jumbo frames pour stockage (MTU 9000)
- Firewall et sécurité planifiés

### Cluster et HA

- Nombre de nœuds déterminé (3-32)
- Stratégie HA définie (N+1, N+2)
- HA rules configurées (affinité/anti-affinité)
- Corosync network planifié (latence <2ms)
- QDevice si cluster 2 nœuds
- Procédures failover documentées

## Validation et Tests

### Tests Pre-Production

- Environnement test créé (identique ou réduit)
- Migration de VMs pilotes effectuée
- Tests de charge réalisés
- Scénarios de panne testés (nœud, réseau, storage)
- Performances mesurées vs attendues
- Procédures rollback validées

### Performance Benchmarking

- CPU : stress-ng, sysbench
- RAM : stream, membench
- Stockage : fio (IOPS, latency, throughput)
- Réseau : iperf3, netperf
- E2E : application-specific benchmarks
- Résultats documentés et analysés

### Sécurité et Conformité

- Firewall Proxmox configuré
- Backup testés et vérifiés
- Disaster Recovery planifié
- Accès et audit configurés

- Chiffrement évalué (si requis)
- Conformité réglementaire vérifiée

## Production et Monitoring

### Mise en Production

- Migration planifiée par phases
- Fenêtres de maintenance communiquées
- Checkpoints et points de retour définis
- Équipe support disponible
- Monitoring actif pendant migration
- Documentation à jour

### Monitoring Continu

- Prometheus/Grafana déployé
- Alerting configuré (CPU, RAM, disk, network)
- Dashboards créés
- Seuils d'alerte définis
- Escalation procédures documentées
- Revues capacité mensuelles planifiées

### Maintenance

- Calendrier de mises à jour établi
- Procédures backup automatisées
- Tests de restauration planifiés (mensuel)
- Revue logs et incidents (hebdomadaire)
- Optimisations continues identifiées
- Documentation maintenue

### Ressources open source associées :

- [HyperVIntrospector](#) — Introspection Hyper-V pour comparaison avec Proxmox (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.

### 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

---

Le dimensionnement d'une infrastructure Proxmox VE 9.0 est un exercice complexe qui nécessite une approche méthodique et une compréhension approfondie des workloads, des nouvelles fonctionnalités (snapshots LVM, SDN Fabrics, HA rules améliorées) et des best practices.

### Points Clés à Retenir

1. **Approche progressive** : Commencer conservateur, monitorer, optimiser
2. **Marges de sécurité** : Toujours prévoir 15-25% de capacité supplémentaire
3. **Haute disponibilité** : N+1 minimum, N+2 pour criticité haute
4. **Stockage diversifié** : Tiering NVMe/SSD/HDD selon workloads
5. **Réseau robuste** : Séparation VLANs, redondance, bande passante suffisante
6. **Tests essentiels** : Valider en pre-production avant la migration
7. **Monitoring actif** : Anticiper les problèmes de capacité
8. **Documentation** : Maintenir à jour les configurations et procédures

## Évolutions Futures

Proxmox VE 9.0 pose les bases pour :

- Support amélioré des architectures hétérogènes (ARM64)
- Intégration IA/ML pour optimisation automatique
- SDN encore plus avancé (multi-datacenter)
- GPU virtualization généralisée

**L'investissement dans une infrastructure bien dimensionnée dès le départ se traduit par des économies substantielles en maintenance, performances supérieures et évolutivité facilitée sur le long terme.**

## Ressources Officielles

- **Documentation** : <https://pve.proxmox.com/wiki/>
- **Forum** : <https://forum.proxmox.com/>
- **Roadmap** : <https://pve.proxmox.com/wiki/Roadmap>
- **Support** : <https://www.proxmox.com/en/proxmox-virtual-environment/support>

---

**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.