AYI NEDJIMI Consultants

Proxmox VE Clustering & Haute Disponibilité — Guide Expert

Catégorie : Virtualisation Lecture : 19 min Publié le : 12/04/2026 Auteur : Ayi NEDJIMI

Maîtrisez le clustering Proxmox VE : quorum Corosync, HA Manager, fencing STONITH, Ceph distribué, live migration. Architectures 2 à 5 nœuds documentées.

Déployer Proxmox VE sur un serveur unique est trivial. Le vrai défi commence quand il faut garantir la continuité de service face à une panne matérielle, planifier des maintenances sans interruption utilisateur, et orchestrer des centaines de machines virtuelles réparties sur plusieurs nœuds physiques. Le clustering Proxmox, appuyé sur Corosync pour la communication inter-nœuds et le HA Manager pour le basculement automatique des charges, constitue le socle de toute infrastructure de virtualisation professionnelle. Ce guide couvre l'ensemble du spectre : de la création du cluster avec pvecm à la configuration avancée du fencing STONITH, en passant par l'intégration Ceph pour le stockage distribué et les stratégies de prévention du split-brain. Les architectures présentées sont issues d'implémentations réelles chez des clients allant de la PME avec 2 nœuds à l'entreprise avec des clusters de 16 nœuds répartis sur plusieurs datacenters. Chaque décision de design est argumentée avec ses compromis techniques et financiers.

Création du cluster et gestion du quorum

Un cluster Proxmox repose sur le protocole Corosync, dérivé du projet OpenAIS, qui fournit un bus de communication fiable entre les nœuds et un système de vote pour déterminer quel groupe de nœuds est autorisé à fonctionner en cas de partition réseau. La notion de quorum — la majorité des votes nécessaire pour former un cluster opérationnel — est le mécanisme fondamental qui prévient le split-brain.

```
# Sur le premier nœud — Création du cluster
pvecm create mon-cluster --link0 10.0.1.1 --link1 10.0.2.1

# Sur les nœuds suivants — Rejoindre le cluster
pvecm add 10.0.1.1 --link0 10.0.1.2 --link1 10.0.2.2

# Vérifier l'état du cluster
pvecm status
pvecm nodes

# Vérifier le quorum
corosync-quorumtool -s
```

Règles de quorum selon le nombre de nœuds

Nombre de nœuds	Votes total	Quorum requis	Pannes tolérées	Notes
2	2	2 (ou 1 avec QDevice)	0 (ou 1 avec QDevice)	QDevice fortement recommandé
3	3	2	1	Configuration minimale recommandée
4	4	3	1	Ajouter QDevice pour tolérer 2 pannes
5	5	3	2	Configuration idéale pour la production
7	7	4	3	Multi-site (3+2+2)

QDevice pour les clusters à nombre pair

Le QDevice (corosync-qdevice) est un démon léger qui s'exécute sur un serveur tiers (hors cluster) et apporte un vote supplémentaire. Dans un cluster à 2 nœuds, il est indispensable pour permettre au cluster de survivre à la perte d'un nœud. Le serveur QDevice peut être une simple VM Debian avec des ressources minimales — il n'héberge aucune charge de travail Proxmox.

```
# Sur le serveur QDevice (Debian 12 externe au cluster)
apt install corosync-qnetd -y

# Sur un nœud du cluster Proxmox
pvecm qdevice setup 10.0.1.100

# Vérifier le fonctionnement
pvecm qdevice status
corosync-quorumtool -s
```

Configuration Corosync et exigences réseau

Corosync nécessite un réseau fiable et à faible latence pour le heartbeat entre les nœuds. La perte de paquets ou une latence excessive peut déclencher des faux positifs de détection de panne, entraînant des redémarrages intempestifs de VMs. L'architecture réseau du cluster est aussi critique que le matériel des nœuds. Pour une vue d'ensemble de la sécurité réseau, consultez notre guide Zero Trust et micro-segmentation.

Architecture réseau recommandée

```
# Configuration Corosync avec double lien (link0 + link1)
# /etc/pve/corosync.conf (géré automatiquement par pvecm)

totem {
    version: 2
    secauth: on
    cluster_name: production
    transport: knet

    interface {
        linknumber: 0
        # Réseau cluster primaire (dédié)
        knet_transport: udp
    }
    interface {
        linknumber: 1
        # Réseau cluster secondaire (redondance)
        knet_transport: udp
    }
}

# Vérifier la santé des liens
pvecm status
# Attention au token timeout (par défaut 1000ms)
# Ajuster si latence réseau > 2ms entre sites
```

> **Règle d'or du réseau cluster**
> Utilisez **toujours** deux liens Corosync sur des réseaux physiquement séparés (switches différents, cartes réseau différentes). Le lien primaire doit être un réseau dédié au cluster, jamais partagé avec le trafic VM ou le stockage. La latence maximale entre les nœuds ne doit pas dépasser **2 ms** pour un cluster sur un site unique, et **10 ms** pour un cluster étendu multi-site. Au-delà, les timeout Corosync doivent être ajustés, augmentant le temps de détection de panne et donc le RTO.

HA Manager : groupes, politiques, priorités

Le HA Manager de Proxmox gère le basculement automatique des VMs et conteneurs en cas de panne d'un nœud. Il surveille l'état des nœuds via le cluster membership de Corosync et redémarre les ressources marquées comme HA sur les nœuds disponibles. Le HA Manager est intégré nativement et ne nécessite aucun composant supplémentaire.

Configuration des groupes HA

```
# Créer un groupe HA pour les serveurs de base de données
ha-manager groupadd db-servers \
  --nodes node1,node2,node3 \
  --restricted 1 \
  --nofailback 0

# Créer un groupe HA avec affinité de site
ha-manager groupadd site-a-apps \
  --nodes node1,node2 \
  --restricted 1

# Ajouter une VM au HA Manager
ha-manager add vm:100 --group db-servers --state started --max-restart 3 --max-relocate 2

# Vérifier le statut HA
ha-manager status
```

Politiques de placement et priorités

L'option `--restricted 1` empêche la VM de s'exécuter sur un nœud non membre du groupe. L'option `--nofailback 0` permet à la VM de revenir automatiquement sur son nœud préféré quand celui-ci redevient disponible. En production, le nofailback devrait être activé (`--nofailback 1`) pour éviter les migrations intempestives pendant les heures de bureau — la migration retour sera planifiée lors de la prochaine fenêtre de maintenance.

Fencing et STONITH

Le fencing est le mécanisme qui garantit qu'un nœud défaillant est effectivement isolé avant de redémarrer ses VMs sur un autre nœud. Sans fencing, un nœud partiellement défaillant pourrait continuer à écrire sur les disques partagés pendant qu'un second nœud lance les mêmes VMs, causant une corruption de données irréversible. Proxmox intègre un watchdog software (softdog) par défaut, mais les environnements de production exigent un fencing matériel.

```
# Vérifier le watchdog actif
ha-manager status
cat /etc/default/pve-ha-manager

# Configuration du watchdog hardware IPMI
apt install ipmitool fence-agents-ipmilan -y

# Test du fencing IPMI
ipmitool -I lanplus -H 10.0.0.101 -U admin -P password chassis power status
ipmitool -I lanplus -H 10.0.0.101 -U admin -P password chassis power off

# Pour les serveurs Dell : iDRAC fencing
# Pour les serveurs HP : iLO fencing
# Pour les environnements cloud : API fencing (APC PDU, etc.)
```

Live migration et storage migration

La migration à chaud est l'une des fonctionnalités les plus utilisées au quotidien dans un cluster Proxmox. Elle permet de déplacer une VM en cours d'exécution d'un nœud à un autre sans interruption de service, typiquement pour vider un nœud avant une mise à jour ou rééquilibrer la charge.

```
# Migration live (stockage partagé requis)
qm migrate 100 node2 --online

# Migration live avec stockage local (utilise la réplication)
qm migrate 100 node2 --online --with-local-disks

# Storage migration (changer le backend de stockage)
qm move-disk 100 scsi0 ceph-pool --delete 1

# Migration en masse (vidange d'un nœud)
for VMID in $(qm list | awk 'NR>1 {print $1}'); do
  qm migrate $VMID node2 --online
done
```

Intégration Ceph pour le stockage distribué

Ceph fournit un stockage distribué répliqué qui élimine le besoin d'un SAN externe. Intégré nativement dans Proxmox VE, il offre un pool de stockage accessible par tous les nœuds du cluster, condition nécessaire pour la live migration et la haute disponibilité. Ceph est aussi le backend de stockage qui offre le meilleur compromis performance/résilience/coût pour les clusters de 3 nœuds et plus. Pour des benchmarks comparatifs, consultez notre architecture cluster 3 nœuds de référence.

```
# Installation de Ceph (intégré dans Proxmox)
pveceph install --version reef

# Création du cluster Ceph
pveceph init --network 10.0.3.0/24 --cluster-network 10.0.4.0/24

# Ajout de monitors (un par nœud, minimum 3)
pveceph mon create
# Exécuter sur chaque nœud

# Ajout des OSDs (un par disque dédié)
pveceph osd create /dev/sdb --db_dev /dev/nvme0n1p1
pveceph osd create /dev/sdc --db_dev /dev/nvme0n1p2

# Création du pool RBD pour les VMs
pveceph pool create vm-pool --size 3 --min_size 2 --pg_autoscale_mode on

# Ajout comme stockage Proxmox
pvesm add rbd ceph-pool --pool vm-pool --monhost 10.0.3.1,10.0.3.2,10.0.3.3
```

Prévention du split-brain

Le split-brain survient quand une partition réseau divise le cluster en deux groupes qui se croient chacun légitimes. Sans mécanisme de protection, les deux groupes pourraient démarrer les mêmes VMs, causant une corruption de données. Proxmox prévient le split-brain via le quorum Corosync, mais plusieurs configurations additionnelles renforcent la protection. La sécurité des infrastructures virtualisées est couverte en détail dans notre analyse des menaces actuelles.

Mesures de prévention

```
# 1. Toujours un nombre impair de votants (nœuds + QDevice)
pvecm expected 3  # Ne JAMAIS utiliser cette commande en production normale

# 2. Configurer les self-fencing timeouts
# Si un nœud perd le quorum, il se redémarre automatiquement après le timeout
# /etc/pve/datacenter.cfg
ha: shutdown_policy=conditional

# 3. Superviser le statut du quorum
corosync-quorumtool -l

# 4. Alerter si le nombre de nœuds actifs passe sous le seuil critique
pvecm status | grep "Quorate:"
```

Architectures HA réelles

Architecture 2 nœuds + QDevice (PME)

Configuration minimale pour de la haute disponibilité. Les deux nœuds partagent un stockage Ceph à 2 replicas (size=2, min_size=1 — avec les risques associés) ou utilisent un NAS externe en NFS/iSCSI. Le QDevice fournit le troisième vote pour le quorum.

Architecture 3 nœuds (recommandée)

Configuration idéale pour la majorité des PME et ETI. Chaque nœud est à la fois hyperviseur et nœud Ceph (hyper-convergé). Trois monitors Ceph, trois managers, et un minimum de 3 OSDs par nœud pour un pool en réplication triple. Cette architecture tolère la perte d'un nœud complet sans interruption de service.

Architecture 5 nœuds multi-site

Pour les organisations nécessitant une continuité d'activité inter-sites : 3 nœuds sur le site principal, 2 nœuds sur le site secondaire, avec un Ceph stretch cluster et un QDevice sur un troisième site (ou en cloud). Cette architecture tolère la perte d'un site complet.

Monitoring de la santé du cluster

```
# Script de monitoring cluster complet
#!/bin/bash
# cluster_health_check.sh

echo "=== Cluster Status ==="
pvecm status

echo "=== HA Status ==="
ha-manager status

echo "=== Ceph Health ==="
ceph health detail
ceph osd tree

echo "=== Node Resources ==="
for node in node1 node2 node3; do
  echo "--- $node ---"
  pvesh get /nodes/$node/status --output-format json | jq '{cpu: .cpu, memory_used:
(.memory.used/.memory.total*100|round), uptime: .uptime}'
done

echo "=== Replication Status ==="
pvesr status
```

FAQ — Questions fréquentes

Un cluster Proxmox à 2 nœuds sans QDevice est-il viable en production ?

Non. Sans QDevice, un cluster à 2 nœuds perd le quorum dès qu'un nœud tombe. Le nœud survivant refuse de démarrer les VMs de l'autre nœud car il ne peut pas garantir que le nœud défaillant est réellement hors service (risque de split-brain). Vous seriez obligé d'intervenir manuellement avec `pvecm expected 1`, ce qui prend du temps et annule le bénéfice de la haute disponibilité. Investissez dans un QDevice — une Raspberry Pi ou une petite VM dans un cloud public suffit.

Peut-on mélanger des nœuds avec des configurations matérielles différentes dans un cluster ?

Oui, c'est supporté et courant en production. Proxmox gère des nœuds hétérogènes sans problème. Cependant, pour la live migration, les CPU doivent être compatibles au niveau des instructions. Utilisez le type CPU `x86-64-v2-AES` ou `kvm64` pour les VMs qui doivent migrer entre des générations de processeurs différentes. Pour Ceph, des disques de performances inégales entre nœuds peuvent créer des hotspots — utilisez les device classes et les CRUSH rules pour gérer l'hétérogénéité.

Comment mettre à jour un cluster Proxmox sans interruption de service ?

La mise à jour rolling est la procédure standard. Videz un nœud en migrant toutes ses VMs vers les autres nœuds (`qm migrate` en masse), appliquez la mise à jour sur le nœud vide, redémarrez-le, vérifiez qu'il rejoint le cluster correctement, puis passez au nœud suivant. Le HA Manager gère automatiquement les migrations si le nœud est mis en mode maintenance. L'ensemble de la procédure est transparent pour les utilisateurs finaux.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

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