

2 - Proxmox VE



2. Proxmox VE

2.1 Une solution Open Source et gratuite



Proxmox Virtual Environment (VE) est une **plateforme open source** complète pour la **virtualisation d'entreprise** tout compris, qui intègre étroitement l'**hyperviseur KVM**, les **conteneurs LXC**, un **système de stockage distribué** ainsi qu'un **gestionnaire de réseau** sur une seule plateforme.

Cette solution gère facilement des clusters de haute disponibilité et les outils de reprise après sinistre grâce à une interface Web intuitive.

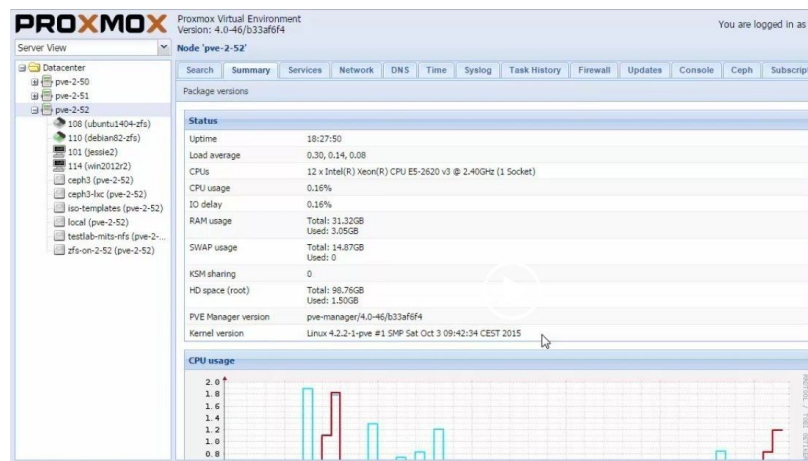


Figure 1 : Fenêtre principale de configuration d'un serveur Proxmox VE



Ses fonctionnalités de **niveau entreprises**, sa **gratuité** et sa **communauté** font de Proxmox VE un choix de taille pour **virtualiser une infrastructure informatique**, **optimiser les ressources existantes** et **augmenter l'efficacité** avec un minimum de frais.

2.2 KVM/Qemu sous le capot

Proxmox VE utilise l'**hyperviseur KVM** couplé avec le **logiciel Qemu** afin de gérer la couche de virtualisation :

- **KVM** pour **Kernel-based Virtual Machine** est un hyperviseur de type 1, libre sous licence GPL, fortement couplé au noyau Linux.
- **Qemu** (abréviation de **Quick Emulator**) est un logiciel open source qui émule un ordinateur physique. Du point de vue du système hôte, Qemu est un programme qui a accès à un certain nombre de ressources locales comme des partitions, des fichiers, des cartes réseaux qui sont ensuite transmises à un ordinateur virtuel qui les voit comme s'il s'agissait de véritables dispositifs.

Un système d'exploitation invité s'exécutant dans l'ordinateur virtuel accède à ces périphériques et fonctionne comme

s'il fonctionnait sur du matériel réel.



Par **exemple**, vous pouvez **passer une image ISO** comme **paramètre à Qemu**, et le **système d'exploitation** fonctionnant dans l'ordinateur émulé verra un **véritable CDROM** inséré dans un lecteur de CD.

Qemu peut émuler une grande variété de matériel d'ARM à Sparc, mais Proxmox VE ne s'intéresse qu'à l'émulation de processeurs x86 32 et 64 bits, puisqu'il représente l'écrasante majorité du matériel serveur.

2.3 Périphériques émulés et paravirtualisés

Le matériel PC émulé par Qemu comprend une carte mère, des contrôleurs réseau, des contrôleurs SCSI, IDE et SATA, des ports série et bien d'autres.



Tous ces **périphériques** sont l'**équivalent logiciel exact** des **périphériques matériels existants**, et si le système d'exploitation tournant dans le client a les **pilotes appropriés**, il utilisera les périphériques **comme s'il fonctionnait sur du matériel réel**.

Cela permet à Qemu d'exécuter des systèmes d'exploitation **non modifiés**.

Cela a cependant un **coût de performance**, car l'exécution dans le logiciel de ce qui était censé fonctionner dans le matériel implique **beaucoup de travail supplémentaire** pour le **CPU hôte**.



Pour **atténuer ce problème**, Qemu peut présenter au système d'exploitation invité des **dispositifs paravirtualisés**, où l'OS invité reconnaît qu'il fonctionne à l'**intérieur de Qemu** et **coopère** avec l'hyperviseur KVM.

Create: Virtual Machine

General OS CD/DVD **Hard Disk** CPU Memory Network Confirm

Bus/Device:	SCSI	0	Cache:	Default (No cache)
Storage:	local-lvm		No backup:	<input type="checkbox"/>
Disk size (GB):	32		Discard:	<input type="checkbox"/>
Format:	Raw disk image (raw)		IO thread:	<input type="checkbox"/>

Help Back Next

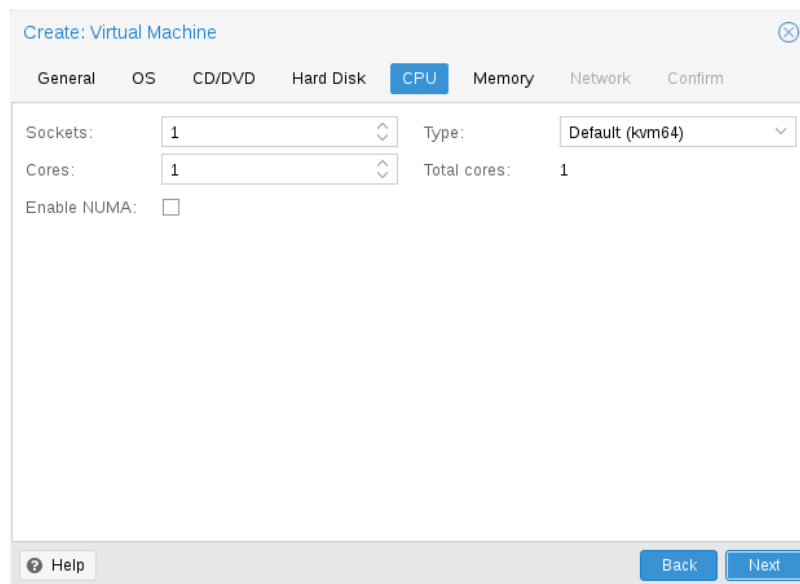


Figure 2 : Fenêtres de configuration d'une VM

Qemu s'appuie sur la norme de **virtualisation virtio**, et est donc capable de présenter des dispositifs virtio paravirtualisés, qui incluent un **contrôleur de disque générique** paravirtualisé, une **carte réseau** paravirtualisé, un **port série** paravirtualisé, un **contrôleur SCSI** paravirtualisé, etc....



Sous Proxmox VE, il est **fortement recommandé** d'utiliser les **appareils virtio** chaque fois que vous le pouvez, car ils fournissent une **grande amélioration des performances**.

L'utilisation du contrôleur de disque générique virtio par rapport à un contrôleur IDE émulé doublera le débit d'écriture séquentielle.

L'utilisation de l'interface réseau virtio peut fournir jusqu'à **trois fois le débit** d'une carte réseau Intel E1000 émulée.

2.4 Notion de cluster

Proxmox VE peut être configuré en **mode cluster**.



Un **cluster** est un **ensemble de serveurs** reliés en **réseau** dans le but d'offrir plus de fonctionnalités.

Dans la plupart des cas, un cluster est composé d'**un ou plusieurs nœuds maîtres** ainsi que d'un **grand nombre de nœuds esclaves**.

Si l'on croit la documentation de Proxmox VE, vous pouvez monter jusqu'à 32 nœuds physiques sans problèmes.

Dans notre cas, les fonctionnalités ci-dessous ne seront accessibles qu'en mode cluster sur Proxmox VE.

- Création de clusters multi-master
- Proxmox Cluster file system (pmxcfs): Système de fichier orienté base de donnée permettant de stocker des petits fichiers (30Mo maximum) de configuration, répliquée en temps réel sur tous les nœuds utilisant « corosync ».
- Migration de Machines Virtuelles entre plusieurs hôtes
- Migration de Conteneur LXC entre plusieurs hôtes
- Fonction d'enregistrement de log à l'échelle du cluster
- API Web RESTful
- Déploiement plus rapide
- Pare-feu à l'échelle du cluster



