

Installer Pimox, hyperviseur de type 1, sur Raspberry Pi 4 (ARM64)

Table of Contents

- [Installer Pimox, hyperviseur de type 1, sur Raspberry Pi 4 \(ARM64\)](#)
 - [Préparer Raspberry Pi OS 64bit](#)
 - [Télécharger et flasher l'image de Pimox](#)
 - [Configuration de Pi](#)
 - [Configuration de Pimox](#)
 - [Installation from scratch](#)
 - [Installation manuelle](#)
 - [Accéder à l'interface Web Pimox](#)
 - [Mise en place du conteneur LXC/LXD](#)
 - [Installation du conteneur](#)
 - [Configuration du conteneur](#)
 - [Démarrer le conteneur](#)
 - [Ajout de machines virtuelles ARM64](#)
 - [Création de la machine virtuelle](#)
 - [Installation d'Ubuntu sur la machine virtuelle](#)

Pimox est un port de **Proxmox Virtual Environment**, un serveur logiciel open-source pour la gestion de la virtualisation, vers le Raspberry Pi permettant de construire un cluster **Proxmox** de Raspberry Pi ou même un cluster hybride de Pis et de matériel x86/x86_64.

Préparer Raspberry Pi OS 64bit

Télécharger et flasher l'image de Pimox

Télécharger l'image à partir de la page de téléchargement de Raspberry Pi et la flasher sur une carte SD. Alternativement, on peut utiliser le Raspberry Pi Imager et flasher la version Lite 64 bits. Sélectionner "Raspberry Pi OS (autres)" et sélectionner "Raspberry Pi OS Lite (64-bit)"

Configuration de Pi

Pour la première exécution, il faut se connecter un moniteur physique et un clavier pour configurer le Pi. Pour que le Raspberry Pi fonctionne Headless, il faut activer SSH lors de la configuration.

La connexion par défaut pour pi est indiquée ci-dessous :

- **nom d'utilisateur:** pi

- **mot de passe:** raspberry

Après s'être connecté, taper **sudo raspi-config** pour démarrer l'outil de configuration et modifier le mot de passe Pi. Pour activer SSH, accéder à **3 Options d'interface** et **2 SSH** dans le menu.

Configuration de Pimox

À partir de ce moment, on peut se connecter à distance via SSH.

Installation from scratch

Utiliser la commande ci-dessous pour exécuter le programme d'installation interactif sur RPiOS64bit.

Passer en tant que root

```
sudo su
```

mettre à jour et upgrader le système

```
apt update && apt upgrade -y
```

Télécharger et exécuter le script d'installation

```
curl  
https://raw.githubusercontent.com/pimox/pimox7/master/RPiOS64-IA-Install.sh  
| sudo bash
```

Suivre l'invite. Une fois l'installation lancée, cela prendra environ 10 à 20 minutes.

Installation manuelle

Vérifier les prérequis suivants:

- Système d'exploitation 64 bits basé sur Debian Bullseye préinstallé (pas 32 bits)
- Dans `/etc/network/interfaces`, donner au Pi une adresse IP statique. On ne peut pas utiliser DHCP.
- Dans `/etc/network/interfaces`, supprimer toutes les adresses IPv6.
- Dans `/etc/hostname`, définir un nom pour le Pi.
- Dans `/etc/hosts`, s'assurer que ce nom d'hôte correspond à l'adresse IP statique définie précédemment.
- S'assurer que les en-têtes du noyau sont installés.

Procéder à l'installation:

```
echo "deb https://raw.githubusercontent.com/pimox/pimox7/master/ dev/" >
/etc/apt/sources.list.d/pimox.list
curl https://raw.githubusercontent.com/pimox/pimox7/master/KEY.gpg | apt-key
add -
apt update
apt install proxmox-ve
```



Utiliser une console locale attachée ! Les connexions réseau seront perdues/réinitialisées pendant la progression de l'installation



Les sources originales de Proxmox peuvent être trouvées sur <https://git.proxmox.com>

Les sources (très minimes) corrigées peuvent être trouvées sur <https://github.com/pimox>

Accéder à l'interface Web Pimox

L'URL par défaut est [https://\[pimox ip\]:8006](https://[pimox ip]:8006). Entrer **root** comme nom d'utilisateur et le mot de passe **root** pour se connecter.

L'interface web est similaire au Proxmox VE.

Mise en place du conteneur LXC/LXD



Seul **aarch64/arm64** est confirmé comme fonctionnant pour les conteneurs VM et LXC/LXD.

Installation du conteneur

Pour le conteneur LXC/LXD, il faut télécharger manuellement les images de conteneur sur le Pimox.

Actuellement, la fonction "télécharger depuis l'URL" dans Pimox téléchargera les conteneurs x8664 et x8664, mais cela ne fonctionnera pas dans Pimox.

On peut accéder à la liste des images de conteneur LXD sur <https://us.lxd.images.canonical.com/images/> ou <https://jenkins.linuxcontainers.org/view/Images/>.

Télécharger l'image Ubuntu 20.04 LTS arm64 lxd à partir de <https://us.lxd.images.canonical.com/images/ubuntu/focal/arm64/cloud/>. Le fichier image à télécharger est nommé root.tar.xz. Une fois téléchargé sur le PC, renommer le fichier avec un meilleur nom, tel

que ubuntu-20.04.tar.xz.

Une fois cela fait, cliquez sur le bouton "Télécharger" et sélectionner le fichier.



Alternativement, on peut utiliser la fonctionnalité "télécharger depuis l'URL". Cependant, Il faut modifier le nom du fichier dans un terminal et en tant que root dans le dossier /var/lib/vz/template/cache.

Configuration du conteneur

Configurer le conteneur en cliquant sur le bouton bleu "Créer CT" en haut à droite de la page. Exécuter l'outil de création comme on le ferait normalement dans n'importe quel Proxmox VE. Il faudra environ 40 secondes à 2 minutes pour créer un conteneur, après tout, il s'agit d'un Raspberry Pi.

Démarrer le conteneur

Démarrer le conteneur et accéder à l'onglet de la console, l'adresse IP est 192.168.1.8/24

Ajout de machines virtuelles ARM64



Pour les machines virtuelles, il faut télécharger l'image ISO arm64.

Création de la machine virtuelle

Télécharger d'abord l'image ubuntu-20.04.1-live-server-arm64.iso à partir de :

<https://mirror.yandex.ru/ubuntu-cdimage/ubuntu/releases/20.04.1/release/>

Ensuite, il faut créer une machine virtuelle avec le support EFI, tous les autres paramètres peuvent être enregistrés dans l'état par défaut, mais il faudra indiquer une image ISO et probablement changer la taille du disque dur, de 32 Go à quelque chose de plus grand.



Proxmox pour les machines virtuelles EFI veut créer à chaque fois un lecteur de 128 Ko, ce qui déclenche à l'étape de démarrage de la VM, une erreur du type :

gemu-system-aarch64 : le périphérique nécessite 67108864 octets, le backend de bloc fournit 131072 octets

Il faut recréer le lecteur EFI, accéder à l'onglet Matériel et supprimer la partition EFI, puis



cliquer sur le bouton Ajouter et la recréer.

Ce message signifie que la VM a besoin d'un lecteur de 64 Mo pour EFI.

Pour activer l'émulation ARM64, après avoir créé via l'interface Web, créer un fichier de configuration sur la VM (par exemple, l'ID de la VM sera 110). Cette configuration se trouve dans le dossier suivant : `/etc/pve/nodes/vm/qemu-server/`.

```
sudo mcedit /etc/pve/nodes/vm/qemu-server/110.conf
```

Ensuite, il faut ajouter au moins `arch : aarch64`, puis supprimer la chaîne telle que : `vmgenid : 00000000000000000000`.

Exemple de configuration :

```
arch: aarch64
bios: ovmf
boot: dcn
bootdisk: scsi0
cores: 1
efidisk0: local:vm-110-disk-1,size=64M
memory: 2048
name: k3s-server
net0: virtio=00:00:00:00:00:00,bridge=vbr0,firewall=1
numa: 0
ostype: l26
scsi0: local:vm-110-disk-0,size=40G
scsi1: local:iso/ubuntu-20.04.1-live-server-arm64.iso,media=cdrom,size=881M
scsihw: virtio-scsi-pci
serial0: socket
smbios1: uuid=ffffffff-ffff-ffff-ffff-ffffffffffffff
sockets: 1
vga: serial0
```



Il est possible de créer une machine virtuelle via l'interface utilisateur **proxmox**:

- **General** : donner un nom à la machine virtuelle. On peut utiliser les valeurs par défaut pour les autres champs. Cliquer sur Suivant pour accéder à l'onglet OS.
- **OS** : Linux et n'utiliser aucun support
- **System** : changer le **BIOS** en **OVMF** et décocher Ajouter un disque EFI.
- **HD** : choisir un disque de stockage, peu importe celui choisi, Il faut s'assurer de lui donner suffisamment de place en Go. Vérifier l'option **Discard**, cela garantira qu'une action **TRIM** dans le système d'exploitation invité supprimera l'espace de stockage utilisé par les fichiers supprimés. On a également la possibilité de choisir l'image disque brute (**raw**) ou le format d'image QEMU (**qcow2**). **Raw** est un peu plus rapide que qcow2 car il a très peu de surcharge et aucune métadonnée associée. **qcow2** offre des fonctionnalités supplémentaires telles que la compression, le cryptage AES et les sauvegardes incrémentielles. Choisir en fonction du cas d'utilisation.

- **CPU** : 1 cœur, on peut le changer plus tard une fois qu'il est installé
- **Memory** : 2048, on peut la modifier plus tard une fois installée
- **Network**: laisser les paramètres par défaut et cliquer sur Suivant

Confirmer, Aller dans l'onglet Matériel:

- Ajouter un lecteur de cd scsi avec l'iso ARM. Retirer le lecteur ide.
- Ajouter un port série sur 0
- Changer l'affichage en terminal série 0



Avec cela, on a un modèle qu'on peut modifier dans la console.

Accéder à `/etc/pve/qemu-system/`, ouvrir le fichier conf de la VM que l'on viens de créer:

- Changer l'arc en **aarch64**
- Ajouter un démarrage : **dcn**

Sortir, puis exécuter

```
z qm set <VMID> -efidisk0 local-lvm:1,format=raw
```

Installation d'Ubuntu sur la machine virtuelle

Ensuite, démarrer la VM et ouvrir la console **xterm.js**, car **noVNC** par défaut ne peut pas fonctionner avec VGA via le port série (chaque pression sur n'importe quelle touche enverra Entrée à la VM).

Après le démarrage d'Ubuntu, attendre un peu (le démarrage peut prendre environ 10 à 20 minutes), puis suivre les instructions d'installation.



Lorsque l'installation est presque terminée et que l'on voit l'étape des mises à jour de sécurité, cliquer sur **Annuler**, car pour une raison quelconque, la machine virtuelle est bloquée à cette étape (onglet d'état ouvert, on verra 100 % d'utilisation du processeur et de la mémoire).

Après l'installation, Ubuntu devrait recevoir l'adresse IP via DHCP, et on peut enfin se connecter.



Le chemin complet du début à la fin de l'installation peut prendre environ quelques heures, alors il faut être patient.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:rpi-pimox>

Last update: **2025/02/19 10:59**

