

# INSTALLATION D'UN SERVEUR PROXMOX SOUS RASPBERRY PI

Raspberry Pi - Debian Bullseye  
**Configuration de base**

Tutoriel **PROXMOX** - RASPBERRY PI

David GOÏTRÉ

## Table des matières

Introduction.....	1
1. Pré requis.....	1
2. Connexion au serveur.....	1
3a. Paramétrage Ethernet du serveur.....	2
3b. Paramétrage Wifi du serveur.....	3
4. Optimisation du système.....	3
5. Installation de Proxmox pour Raspberry PI 4.....	3
6. Installation de DietPi pour Raspberry PI 3B+.....	4
7. Installation de Proxmox por Raspberry PI 3B+.....	5
8. Accéder à l'interface WEB de Pimox.....	5
9. Créer une machine virtuelle.....	6
10. Commandes spécifiques DietPi.....	8
11. Commandes RaspberryPi.....	8
12. Liens annexes.....	8
13. Conclusion.....	8

## Introduction

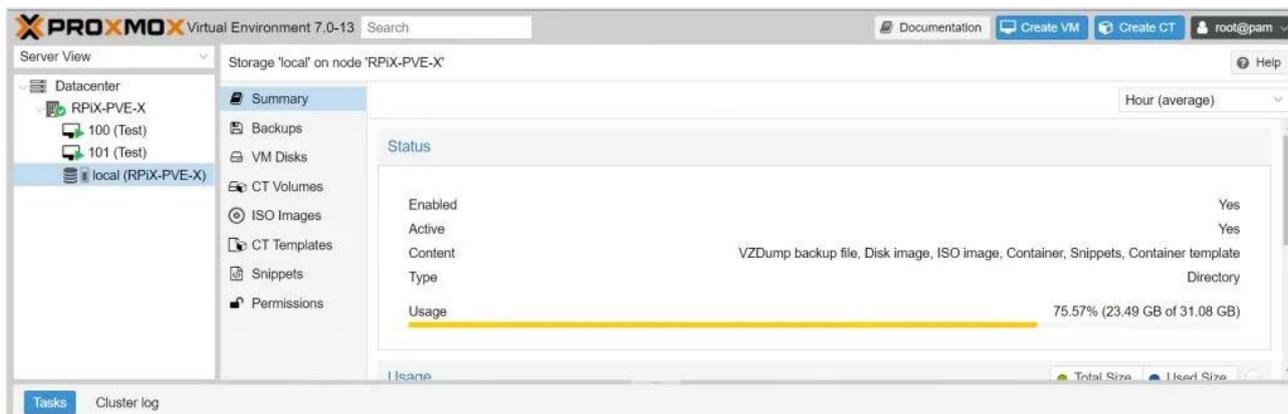
**PIMOX 2.0** est une plate-forme open source complète pour la virtualisation d'entreprise. Grâce à l'interface Web intégrée, on peut facilement gérer les machines virtuelles et les conteneurs, le stockage défini par logiciel et la mise en réseau, le clustering haute disponibilité et plusieurs outils prêts à l'emploi sur une seule solution.

## 1. Pré requis

On a besoin des différents matériels et logiciels pour la création d'un Serveur PIMOX avec un RaspberryPi.

- Un ou des PC client sous Windows
- Une Box (Free, Orange, Sfr...)
- Un Raspberry 3B+ avec l'OS Raspian Bullseye installé avec [Etcher](#)
- Le logiciel [Putty](#) pour se connecter en SSH au serveur
- Connaitre l'interface réseau (eth0, br0, ens3...) via la commande : `ip a`  
Pour notre test c'est l'interface **eth0** qui sera utilisée

Voici l'interface que l'on doit obtenir une fois connecter au serveur **PIMOX** mise en place



## 2. Connexion au serveur

a) Activer le **SSH** sur le serveur. Pour ce faire, ouvrir la carte SD du RaspberryPi via l'explorateur de Windows et créer un fichier **ssh** (sans extension) à sa racine.

b) Ouvrir **Putty** et se connecter au serveur avec les identifiants (par défaut **pi/raspberry**)

c) Mettre à jour les packages du système vers la dernière version. Exécuter la commande suivante pour mettre à jour et mettre à niveau les packages de votre système :

```
# apt-get update -y  
# apt-get upgrade -y
```

### 3a. Paramétrage Ethernet du serveur

Avant d'aller plus loin, il nous faut connaître l'interface réseau de notre serveur **RaspberryPI** et lui attribuer une adresse IP fixe.

a) Lister les interfaces

```
$ ip link | awk '{ print $2}' # liste les interfaces  
# ethtool <interface> | grep detected # détecte l'interface connectée
```

b) Définir une adresse IP fixe

```
# nano /etc/dhcpd.conf # ouvre le fichier de configuration réseau
```

c) Copier le texte ci-dessous à la fin du fichier **dhcpd.conf**

```
interface nom de l'interface réseau  
static ip_address=192.xxx.xxx.xxx/24  
static routers=192.xxx.xxx.xxx
```

d) Rebooter le serveur

```
# sudo reboot
```

e) Paramétrer le serveur

```
$ raspi-config # ouvre l'utilitaire, sélectionner le menu System Options
```

```
Raspberry Pi Software Configuration Tool (raspi-config)  
  
1 System Options          Configure system settings  
2 Display Options        Configure display settings  
3 Interface Options       Configure connections to peripherals  
4 Performance Options     Configure performance settings  
5 Localisation Options    Configure language and regional settings  
6 Advanced Options        Configure advanced settings  
8 Update                  Update this tool to the latest version  
9 About raspi-config      Information about this configuration tool
```

Sélectionner le menu **S3 Password** pour modifier le mot de passe et **S4 Hostname** pour modifier le nom du serveur.

```
Raspberry Pi Software Configuration Tool (raspi-config)  
  
S1 Wireless LAN          Enter SSID and passphrase  
S2 Audio                  Select audio out through HDMI or 3.5mm jack  
S3 Password              Change password for the 'pi' user  
S4 Hostname               Set name for this computer on a network  
S5 Boot / Auto Login     Select boot into desktop or to command line  
S6 Network at Boot       Select wait for network connection on boot  
S7 Splash Screen         Choose graphical splash screen or text boot  
S8 Power LED             Set behaviour of power LED
```

### 3b. Paramétrage Wifi du serveur

Par défaut le Wifi est désactivé. Il faut créer un fichier **wpa\_supplicant.conf** et le copier à la racine de la carte SD, permettant à Raspberry Pi OS de lire le fichier au prochain démarrage et d'appliquer la configuration directement.

a) ouvrir un éditeur de texte et copier le texte suivant

```
country=FR
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="NOM_RESEAU"
    scan_ssid=1 #nécessaire quand le ssid n'est pas diffuser
    pk="MOTDEPASSE"
    key_mgmt=WPA-PSK
}
```

b) Modifier les champs du **SSID** et **PSK**

c) Enregistrer le fichier sous le nom **wpa\_supplicant.conf** et copier-le à la racine de la carte SD

### 4. Optimisation du système

Si on utilise le Raspberry Pi pour DOCKER sans écran connecté, il est recommandé d'affecter le minimum de RAM à la partie vidéo. Il suffit de se connecter en SSH et d'éditer le fichier config

```
# sudo nano /boot/config.txt
```

c) Ajouter ou modifier les lignes du fichier config, comme ci-dessous :

```
gpu_mem=16
disable_l2cache=0
gpu_freq=250
```

d) Rebooter le Raspberry

### 5. Installation de Proxmox pour Raspberry Pi 4

Par défaut, le paquet **PROXMOX** n'existe dans le référentiel de Debian Bullseye. On doit donc utiliser une solution de contournement pour l'installer. **PIMOX** est un portage de PROXMOX créé spécialement pour le Raspberry Pi 3 et 4.

a) Télécharger le script d'installation de **Pimox pour Pi 4**

```
# sudo axel https://gdidees.eu/userfiles/file/scripts/RPiOS64autoinstall.sh
```

b) Editer le script et modifier premières lignes non commentées **RPI\_IP** et **GATEWAY**

```
# sudo nano RPiOS64autoinstall.sh
```

Résultat :

```
RPI_IP : 192.168.x.xxx (@IP du serveur)
GATEWAY : 192.168.x.xxx (@IP de la Box)
```

c) Rendre le script exécutable

```
# chmod +x RPiOS64autoinstall.sh
```

d) Installer **Pimox** à l'aide du script d'installation

```
# sudo ./RPiOS64autoinstall.sh
```

Il nous demandera immédiatement un mot de passe qui sera utiliser plus tard pour accéder à l'interface Web. Ensuite, il n'y a plus rien avoir d'autre à faire. Le script d'installation fera tout. Si une question vous est posée lors de l'installation, conserver les valeurs par défaut.

## 6. Installation de DietPi pour Raspberry PI 3B+

Grâce au **DietPi Automation\_Custom\_Script.sh**, il est très facile d'installer un port de Proxmox Virtual Environment 7 - PiMox7 sur Raspberry PI 3B+.

a) Télécharger les fichiers nécessaires pour l'installation

```
# sudo axel https://gdidees.eu/userfiles/file/scripts/pimox-script.zip
```

b) Télécharger l'image [DietPi](#)

c) Flasher la carte SD avec Etcher

d) Remplacer les fichiers **dietpi.txt** et **config.txt** d'origine et ajouter le script d'installation **Automation\_Custom\_Script.sh** à la racine de la carte SD.

e) Ajuster le réseau et la configuration locale dans le fichier **dietpi.txt** pour correspondre à votre configuration (voir l'exemple ci-dessous).

```
#####  
#---- PVE-CONFIGURATION ---- edit by Tux.feat.Mac | J.T. ----- Ver.: 0.0.2 - 14.11.2021 -----#  
#####  
#### Adjust your HOSTNAME, IP, MASK, GATEWAY and DNS underneath.  
AUTO_SETUP_NET_HOSTNAME=RPi3B-PVE-01  
AUTO_SETUP_NET_STATIC_IP=192.168.178.123  
AUTO_SETUP_NET_STATIC_MASK=255.255.255.0  
AUTO_SETUP_NET_STATIC_GATEWAY=192.168.178.1  
AUTO_SETUP_NET_STATIC_DNS=192.168.178.1  
  
#### ADJUST LOCALES AND COUNTRY TO MATCH YOUR LOCATION ####  
#### Locale eg : "en_GB.UTF-8" / "de_DE.UTF-8" | One entry and UTF-8 ONLY!  
AUTO_SETUP_LOCALE=en_GB.UTF-8  
#### Keyboard Layout eg : "gb" / "us" / "de" / "fr"  
AUTO_SETUP_KEYBOARD_LAYOUT=de  
#### WiFi country code eg: GB US DE JP): https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2  
AUTO_SETUP_NET_WIFI_COUNTRY_CODE=DE  
#### Timezone eg : "Europe/London" https://en.wikipedia.org/wiki/List_of_tz_database_time_zones  
AUTO_SETUP_TIMEZONE=Europe/Berlin  
  
#### CHANGE YOUR DEFAULT PASSWD ####  
AUTO_SETUP_GLOBAL_PASSWORD=raspberry  
#####  
#---- END-PVE-CONFIGURATION - ! NO TOUCHI BELOW THIS LINE ! - UNLESS YOU KNOW WHAT YOU ARE DOING ----#  
#####
```

## 7. Installation de Proxmox por Raspberry PI 3B+

a) Ouvrir putty et se connecter avec les identifiants de Dietpi. Par **défaut** les identifiants sont :

- Nom d'utilisateur : root
- Mot de passe : dietpi (si différent, saisir le mot de passe du fichier dietpi.txt)

b) Installer **Pimox**

```
# cd /boot
# sudo chmod +x Automation_Custom_Script.sh
# sudo ./Automation_Custom_Script.sh
```

## 8. Accéder à l'interface WEB de Pimox

a) Une fois le serveur installer, **lancer le à partir du navigateur** via l'url **@IP:8006**, saisir les identifiants de Dietpi et cliquer sur **Login** pour démarrer le processus de configuration.



Proxmox VE Login

User name: root

Password: .....

Realm: Linux PAM standard authentication

Language: English

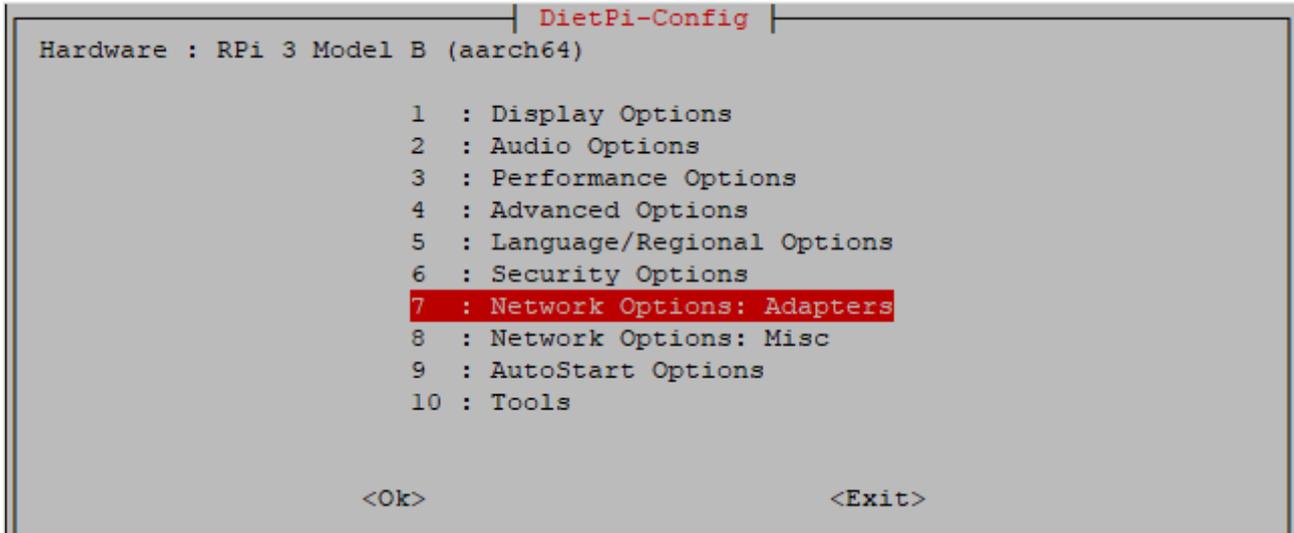
Save User name:  Login

b) Si l'interface web ne s'affiche pas, activer la connexion Ethernet avec la commande **dietpi-config**, puis sélectionner la **ligne 7**

```
DietPi-Config
Please select an option to change:
    ●- Adapter Options -●
Ethernet : Available | [On] | Connected
WiFi    : Not Found | [Off] | Disconnected
Onboard WiFi : [Off]
    ●- Additional Options -●
IPv6    : [Off]
Proxy   : [Off]
Test    : Run internet connection test

    <Ok>                               <Back>
```

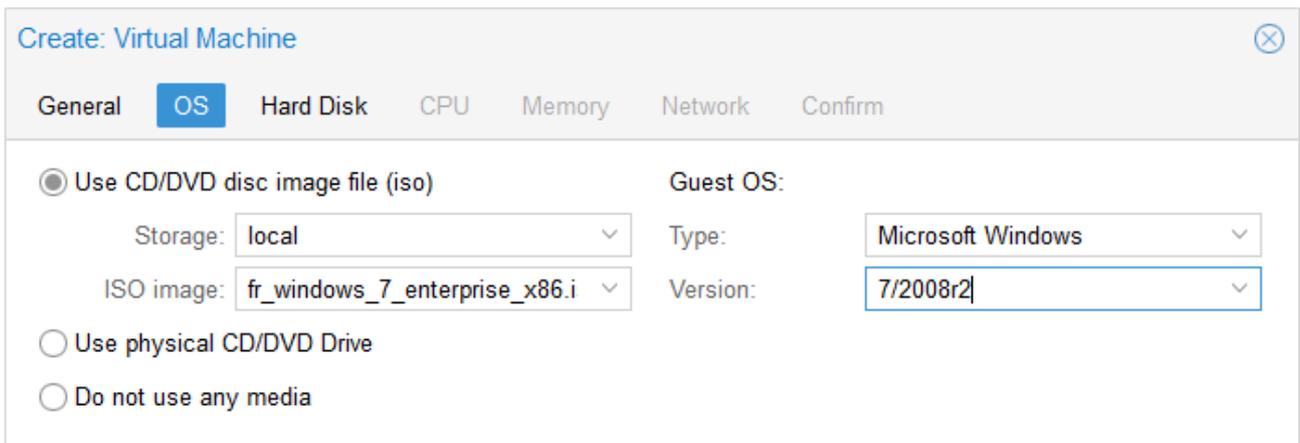
c) Sélectionner la ligne **Ethernet...** et valider avec le bouton **Ok**



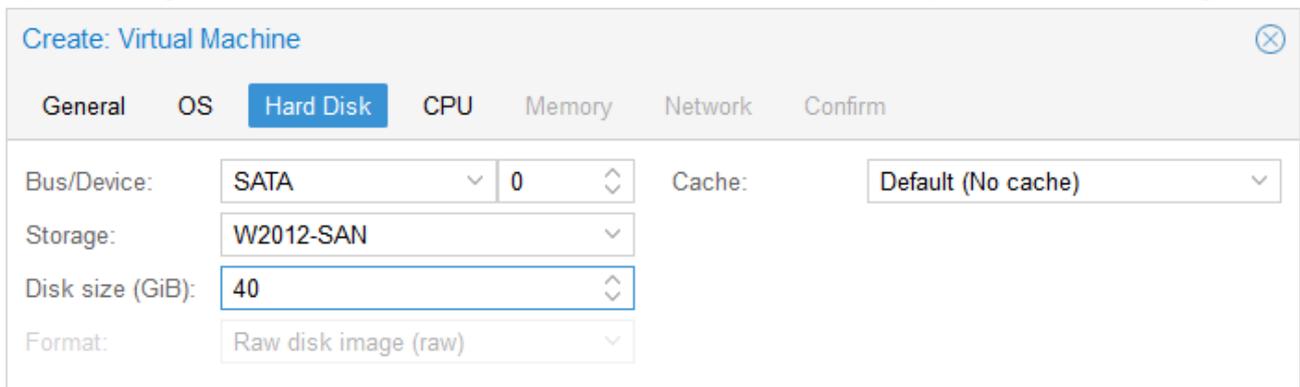
## 9. Créer une machine virtuelle

Avant d'installer une machine virtuelle, il faut copier l'image ISO de l'OS dans le stockage local dans le répertoire `/var/lib/vz/template/iso/` de notre Proxmox. Cliquer ensuite sur le bouton **Create VM**.

a) Dans l'onglet **OS**, sélectionner l'image ISO et spécifier le type et la version du système que l'on souhaite installer



b) Dans l'onglet **Hard Disk**, mettre le bus en mode **SATA** et choisir l'emplacement de stockage



c) Dans l'onglet **CPU**, choisir le nombre de processeur et de cœurs à allouer

Create: Virtual Machine ⊗

General OS Hard Disk **CPU** Memory Network Confirm

Sockets:  Type:

Cores:  Total cores: 2

d) Dans l'onglet **Memory**, choisir la taille de mémoire RAM à allouer

Create: Virtual Machine ⊗

General OS Hard Disk CPU **Memory** Network Confirm

Memory (MiB):

e) Une fois la configuration terminée, il faut donner un nom à notre machine

Create: Virtual Machine ⊗

**General** OS Hard Disk CPU Memory Network Confirm

Node:  Resource Pool:

VM ID:

Name:

e) Dans l'onglet **Confirm**, valider la configuration

Create: Virtual Machine ⊗

General OS Hard Disk CPU Memory Network **Confirm**

Key ↑	Value
cores	2
ide2	local:iso/fr_windows_7_enterprise_x86.iso,media=cdrom
memory	2048
name	Win7a
net0	e1000,bridge=vbr0
nodename	proxmoxa
numa	0
ostype	win7
sata0	W2012-SAN:40
scsihw	virtio-scsi-pci

## 10. Commandes spécifiques DietPi

a) Pour mettre à jour, nous devons d'abord désinstaller Portainer puis installer la mise à jour

```
# dietpi-software # Installe les logiciels sélectionnés
# dietpi-config # configure le réseau de Dietpi
# dietpi-update # vérifie les mises à jour de Dietpi
# dietpi-backup # sauvegarde le système
# dietpi-launcher # affiche le menu des commandes de Dietpi
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-
ce
```

## 11. Commandes RaspberryPi

Liste des commandes basiques à la gestion du serveur RaspberryPi

```
# sudo -i # passe en mode root
# shutdown -h now # éteint le serveur en toute sécurité
# shutdown -r now # redémarre le serveur en toute sécurité
# apt install openssh-server # installe le SSH
# systemctl enable sshd.service # active le service SSH au démarrage
##### Désactive la mise en veille #####
# systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target
```

## 12. Liens annexes

Liste de contenu à télécharger pour Docker et Portainer

- [Support Debian](#)

- [Support Dietpi](#)

## 13. Conclusion

**PIMOX** est installé et configuré avec succès sur le serveur **RaspberryPi Debian 11**. On peut désormais créer des machines virtuelles.

Destiné au RaspberryPi (Raspbian), **PIMOX** fonctionne aussi parfaitement sur une distribution Ubuntu, Debian...

Pour **Debian** :

```
# echo "deb [arch=amd64] http://download.proxmox.com/debian/pve bullseye pve-no-
subscription" > /etc/apt/sources.list.d/pve-install-repo.list
# sudo wget https://enterprise.proxmox.com/debian/proxmox-release-bullseye.gpg -O
/etc/apt/trusted.gpg.d/proxmox-release-bullseye.gpg
# sudo apt update && apt full-upgrade
# sudo apt install pve-kernel-5.15
```