

# PIBOX OS sur Raspbian

## Table of Contents

- [PIBOX OS sur Raspbian](#)
  - [Préparation du système](#)
  - [Installation de Pibox-os](#)
    - [Prise en charge des ventilateurs PWM](#)
    - [Activer le pilote d'affichage](#)
    - [Installer K3S](#)
    - [Installer les binaires](#)
  - [Installation de OpenMediavault](#)
  - [Installation de Jellyfin](#)
    - [Installation depuis le dépôt officiel](#)
    - [Installation depuis le dépôt extrepo](#)
    - [Configuration](#)

Le **PiBox** est un stockage de type NAS, piloté par un Raspberry Pi Compute Module4.

Le référentiel <https://github.com/kubesail/pibox-os> fournit, outre une image disque prête à l'emploi, des scripts et des modules de noyau utilisés pour modifier le système d'exploitation Raspberry Pi afin de tirer pleinement parti du matériel **PiBox**.

Le présent tutoriel permet de personnaliser l'installation avec:

- **OpenMediaVault**, une solution de stockage en réseau (NAS) construite sur Debian et prenant en charge le Raspberry Pi.
- **Jellyfin**, un logiciel de serveur multimédia sous licence libre, conçu pour organiser, lire et partager des fichiers multimédias numériques sur des appareils en réseau.

## Préparation du système



L'installation de **OpenMediaVault** proscrit l'utilisation des variantes de système d'exploitation avec des bureaux graphiques. **OpenMediaVault** ne fonctionnera que sur les systèmes basés sur des terminaux. Dans le cas du Pi, il s'agit du Raspberry Pi OS Lite.

La première chose à faire est de mettre à jour et de mettre à jour le système Raspberry :

```
sudo apt update
sudo apt upgrade
```

Pour compiler les modules il faut installer le noyau RPI:

```
sudo apt install -y raspberrypi-kernel-headers
```

Dans le fichier `/boot/cmdline.txt`, ajouter `cgroup_enable=cgroup cpuset cgroup_enable=memory cgroup_memory=1` à la fin du fichier.

## Installation de Pibox-os



Le script suivant installe tout (PWM, écran, k3s et agent KubeSail)

```
curl -s  
https://raw.githubusercontent.com/kubesail/pibox-os/main/provision-  
os.sh | sudo bash
```

Récupérer le dépôt PiBox OS:

```
git clone https://github.com/kubesail/pibox-os.git
```

## Prise en charge des ventilateurs PWM

Pour rendre le ventilateur silencieux et tourner aussi vite que nécessaire, il faut activer le module **bcm2835-1** pour permettre à un service **pi\_fan\_hw pwm.service** d'envoyer le signal correct au ventilateur à l'aide du contrôleur PWM matériel du Pi.

```
cd pibox-os/pwm-fan  
tar zxvf bcm2835-1.68.tar.gz  
cd bcm2835-1.68  
./configure  
make  
sudo make install  
cd ..  
make  
sudo make install
```

## Activer le pilote d'affichage

Un service d'affichage qui affiche des statistiques et d'autres informations utiles est fourni. Pour l'installer:

```
pushd pibox-os/st7789_module  
curl -O  
https://raw.githubusercontent.com/torvalds/linux/master/drivers/staging/fbtf  
t/fbtf.h
```

```
make
xz fb_st7789v.ko
mv /lib/modules/"$(uname -r)"/kernel/drivers/staging/fbtf/fb_st7789v.ko.xz
/lib/modules/"$(uname -r)"/kernel/drivers/staging/fbtf/fb_st7789v.BACK
mv fb_st7789v.ko.xz /lib/modules/"$(uname -
r)"/kernel/drivers/staging/fbtf/fb_st7789v.ko.xz
popd
dtc --warning no-unit_address_vs_reg -I dts -O dtb -o /boot/overlays/drm-
minipitft13.dtbo pibox-os/overlays/minipitft13-overlay.dts
cat <<EOF >> /boot/config.txt
dtoverlay=spi0-lcs
dtoverlay=dwc2,dr_mode=host
hdmi_force_hotplug=1
dtoverlay=drm-minipitft13,rotate=0,fps=60
EOF
```

## Installer K3S

Les binaires fournis par Pibox-OS requièrent **K3s** une distribution kubernetes légère.

```
curl -sL https://get.k3s.io | sh -s - --write-kubeconfig-mode 644
```

Lors de l'installation de **K3S**, un fichier kubeconfig est écrit dans /etc/rancher/k3s/k3s.yaml. Le moyen le plus courant de se connecter au cluster Kubernetes est la commande **kubectl**, qui est automatiquement installée sur le nœud du cluster lors de l'installation de **K3S**.

Pour se connecter au cluster, il faut indiquer à **Kubectl** où trouver le **kubeconfig**, en définissant une variable d'environnement ou en le copiant dans ~/.kube/config:

```
mkdir ~/.kube
sudo k3s kubectl config view --raw | tee ~/.kube/config
chmod 600 ~/.kube/config
```



Pour configurer la variable d'environnement **KUBECONFIG** utiliser la commande suivante:

```
export KUBECONFIG=~/.kube/config
```

Et pour rendre ce paramétrage persistant, ajouter cette commande dans ~/.profile ou ~/.bashrc

Vérifier que le cluster Kubernetes est bien en cours d'exécution avec la commande :

```
kubectl get pods -A
```

## Installer les binaires

```
curl -s https://raw.githubusercontent.com/kubesail/pibox-os/main/setup.sh |  
sudo bash  
curl -s  
https://raw.githubusercontent.com/kubesail/pibox-os/main/update-framebuffer.  
sh | sudo bash
```

## Installation de OpenMediaVault

Exécuter la commande suivante pour télécharger le script d'installation d'**OpenMediaVault** et le diriger directement vers bash.

Ce script installera et configurera tout le nécessaire pour exécuter **OpenMediaVault** sur le Raspberry Pi.

```
wget -O -  
https://raw.githubusercontent.com/OpenMediaVault-Plugin-Developers/installSc  
ript/master/install | sudo bash
```

Ce processus peut prendre un certain temps car le script doit installer de nombreux packages requis par le logiciel **OpenMediaVault**.



**k3s** est livré avec un contrôleur d'entrée **traefik** préinstallé qui se lie aux ports 80, 443 et 8080 sur l'hôte, on ne peut donc utiliser ces ports pour un serveur tiers.

Pour modifier le port d'**OpenMediaVault** (80 par défaut), exécuter `omv-firstaid`, sélectionner Option-2 "configure web control panel".

Une fois le processus d'installation terminé, il est recommandé de redémarrer le Raspberry Pi.

```
sudo reboot
```

## Installation de Jellyfin

### Installation depuis le dépôt officiel



Par défaut, le gestionnaire de packages **apt** ne prend pas en charge les référentiels exécutés derrière HTTPS.

\\Pour contourner ce problème, il faut installer le package **apt-transport-https** en

exécutant la commande suivante:



```
sudo apt install apt-transport-https lsb-release
```

Ce package ajoute la prise en charge du protocole de transport HTTPS au gestionnaire de packages **apt**.

Une fois le Raspberry Pi à jour, il faut installer certains fichiers pour accéder au référentiel de packages **Jellyfin**.

- importer la clé de signature GPG:

```
curl https://repo.jellyfin.org/debian/jellyfin_team.gpg.key | gpg --dearmor  
| sudo tee /usr/share/keyrings/jellyfin-archive-keyring.gpg >/dev/null
```

- ajouter le référentiel **Jellyfin**:

```
echo "deb [signed-by=/usr/share/keyrings/jellyfin-archive-keyring.gpg  
arch=$( dpkg --print-architecture )] https://repo.jellyfin.org/debian $(  
lsb_release -c -s ) main" | sudo tee /etc/apt/sources.list.d/jellyfin.list
```



Cette ligne récupère automatiquement l'architecture système actuelle et sa version actuelle et remplit les espaces vides.

- mettre à jour la liste des packages:

```
sudo apt update
```



Lorsqu'on modifie les fichiers sources, il faut mettre à jour la liste des packages. Sans mise à jour, le gestionnaire de packages ne sera pas au courant des modifications.

- installer le package **jellyfin**:

```
sudo apt install -y jellyfin
```

## Installation depuis le dépôt extrepo

Pour commencer, ajouter le dépôt de **Jellyfin**:

```
sudo apt install extrepo -y
```

```
sudo extrepo enable jellyfin
```

Rafraîchir les dépôts et installer jellyfin

```
apt update  
apt install jellyfin -y
```

## Configuration

Se rendre à l'adresse : <http://ip-du-serveur:8096>

- Sélectionner la langue souhaitée
- Créer un compte utilisateur
- Ensuite on peut ajouter les dossiers en local ou sur un dossier réseaux comme un NAS où sont stockés les films, séries, etc ..
- Sélectionner la Langue et pays pour les paramètre du serveur



Activer uniquement "Autoriser le mapping automatique de port" si le routeur supporte l'UPnP et si on souhaite accéder au serveur sur internet

From:

<http://vps101364.serveur-vps.net/> - **dwndoc**

Permanent link:

<http://vps101364.serveur-vps.net/doku.php?id=prive:pibox>

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

