

# INSTALLATION D'UN SERVEUR JEEEDOM SOUS RASPBERRY PI

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

Tutoriel **JEEEDOM** - 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. Installation de JEEDOM .....	3
5. Optimisation du système.....	3
6. Commandes RaspberryPi.....	4
7. Liens annexes.....	4
8. Conclusion .....	4

## Introduction

**JEEDOM** est un logiciel Open-Source gratuit qui peut s'installer sur n'importe quel système Linux. Il est basé sur un noyau avec de multiples fonctionnalités : gestion de scénarios, interaction avec l'installation domotique de manière textuelle et sonore, visualisation d'historiques et génération de courbes et graphiques, mise en relation de tous les équipements et objets connectés, personnalisation de l'interface... Son interface claire et intuitive permet de mettre en place une solution complète sans connaissance en développement.

## 1. Pré requis

On a besoin des différents matériels et logiciels pour la création d'un Serveur JEEDOM 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 VPN
- Connaître 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 **JEEDOM** 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 JEEDOM 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/network/interfaces # ouvre le fichier des interfaces
```

c) Copier le texte ci-dessous dans le fichier **interfaces**

```
# Interface reseau de bouclage
auto lo
iface lo inet loopback
# Interface reseau principale
allow-hotplug eth0
iface eth0 inet static
address 192.xxx.xxx.xxx
netmask 255.255.255.0
gateway 192.xxx.xxx.xxx
```

d) Rebooter le serveur

```
# /etc/init.d/networking restart
# 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
```

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
```

### 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
    psk="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. Installation de JEEDOM

Par défaut, le paquet **JEEDOM** n'est pas disponible dans le référentiel de Debian 11. Il faut :

a) Installer les packages nécessaires avec la commande suivante

```
# wget -O- https://raw.githubusercontent.com/jeedom/core/master/install/install.sh |
sudo bash
```

b) Se connecter à **JEEDOM** en saisissant l'**@IP** du serveur dans le navigateur  
Les identifiants par défaut sont : **admin/admin**

### 5. Optimisation du système

Si on utilise le Raspberry Pi pour Jeedom 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

## 6. Commandes RaspberryPi

a) Liste des commandes basique à la gestion du serveur RaspberryPi

```
# 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
```

## 7. Liens annexes

Liste de contenu à télécharger pour Docker et Portainer

- [Installation Jeedom](#)

- [Guides Jeedom](#)

## 8. Conclusion

**JEEDOM** est installé et configuré avec succès sur le serveur **RaspberryPi Debian 11**. On peut désormais accéder à Internet en toute sécurité et protéger son identité.

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

Pour **Debian** : [Installation de Docker Linux Debian](#)