

Installer et exécuter des modèles d'IA en local avec Ollama et WebUI



Bénévoles Numériques de Touraine

Ausmane MOHAMED

Aout 2026



AVERTISSEMENT

- Toutes les marques, noms de produits et de sociétés précisés dans ce document sont cités à des fins d'explications et sont la propriété de leurs détenteurs respectifs.
- Les logos appartiennent également à leurs propriétaires respectifs.
- Les modifications apportées sur ce document ainsi que les scripts sont la propriété de leurs auteurs.
- En aucun cas l'auteur, les distributeurs et toute personne ayant contribué à ce document ne peuvent être tenus responsables pour les dommages qui pourraient survenir par la mise en pratique des recommandations exposées dans ce texte.

Les prérequis pour installer Ollama et WebUI

Les prérequis pour installer Ollama sur un serveur Debian 13.04

Prérequis matériel pour la machine virtuelle (VM)

- CPU : 4 processeurs et 4 coeurs
- RAM : 20 Go
- Disque dur : 32 Go
- Carte réseau : 1

Prérequis logiciels

- Ubuntu Server Debian 13.04 LTS fraîchement installé
- Ollama
- Docker
- Open WebUI
- Achetez un nom de domaine chez un hébergeur
- Configurer le DNS (Domain Name Server) pour qu'il pointe vers votre adresse IP sur Internet
- Installer et configurer Nginx Proxy Manager
- Configurer votre box pour la redirection des ports (NAT/PAT)

Les différentes étapes

Les différentes étapes pour installer Paheko sur un serveur Ubuntu 24.04

- Mise à jour système
- Configuration d'adresse IP fixe du serveur Debian 13.04
- Installation de serveur Ollama
- Utilisation d'Ollama en ligne de commande
- Installation de Docker et Docker compose
- Installation d'Open WebUI
- Connecter Open WebUI à Ollama serveur
- Mise en place d'un certificat Let's Encrypt du serveur Ollama avec Nginx Proxy Manager
- Rendre Ollama accessible depuis internet

ETAPE 1 : Mise à jour du serveur debian

1 Mise à jour Debian

- Taper les commandes suivantes

```
apt update
apt upgrade
apt dist-upgrade
```

2 Configuration de l'environnement local UTF-8

- Taper la commande suivante

```
dpkg-reconfigure locales
```

3 installation agent QEMU pour Proxmox

- Taper la commande suivante

```
apt install qemu-guest-agent
```

ETAPE 2 : Configuration d'adresse IP fixe du serveur Debian server 13.04

1 Identification de l'interface réseau

- Taper la commande suivante pour obtenir le nom de l'interface réseau connectée à votre réseau local

```
ip a
```

Resultat de la commande ip a

```
2 : ens18 : <BROADCAST MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
inet 192.168.1.155/24 brd 192.168.1.255 scope global ens18
```

Le nom de l'interface réseau est **ens18**

2 Attribuer une adresse IP fixe au serveur Debian 13.04

- Editer le fichier `/etc/network/interfaces` pour configurer l'adresse IP fixe du serveur

```
vi /etc/network/interfaces

auto lo
iface lo inet loopback
allow-hotplug ens18
iface ens18 inet static
    address 192.168.1.78
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
    dns-nameservers 192.168.1.1
```



- 192.168.1.78 : l'adresse IP fixe du serveur
- 192.168.1.1 : l'adresse IP de la passerelle
- 192.168.1.1 : l'adresse IP du serveur DNS

3 Configurer la résolution du nom du serveur en local

- [Editer le fichier /etc/resolv.conf](#) pour configurer la résolution du nom du serveur en local

```
vi /etc/resolv.conf

nameserver 192.168.1.1
```

4 Modifier le fichier /etc/hostname

- [Editer le fichier /etc/hostname](#) pour configurer le nom du serveur en local

```
vi /etc/hostname

ollama-lab
```

5 Modifier le fichier /etc/hosts

- [Editer le fichier /etc/hosts](#) pour configurer le nom du serveur en local

```
vi /etc/hosts

127.0.0.1 localhost
127.0.0.1 ollama-lab
192.168.1.78 ollama-lab ollama-lab.votrenomdedomaine.fr
```

ETAPE 3 : Installation de serveur Ollama (sans docker)

1 Installer la commande curl

► Taper la commande suivante

```
sudo apt install curl -y
```

2 Lancer le script d'installation de Ollama

► Taper la commande suivante

```
curl -fsSL https://ollama.com/install.sh | sh
```



- **curl** télécharge le script d'installation depuis le site officiel d'Ollama
- **sh** exécute le script d'installation
- Le script détecte automatiquement votre distribution Linux et installe Ollama

3 Vérifier que le démon tourne correctement

► Taper la commande suivante

```
systemctl status ollama
```

4 Vérifier la version installée

► Taper la commande suivante

```
ollama --version
```

5 Vérifier l'état du service

► Taper la commande suivante

```
systemctl status ollama
```

6 Démarrer / Arrêter / Redémarrer

► Taper la commande suivante

```
systemctl start ollama  
systemctl stop ollama  
systemctl restart ollama
```

7 Afficher les logs du service pour diagnostiquer

➤ Taper la commande suivante

```
sudo journalctl -u ollama -f
```

ETAPE 4 : Utilisation du serveur Ollama en ligne de commande

1 Vérifier l'état du service Ollama

➤ Taper la commande suivante

```
systemctl status ollama
```

2 Redémarrer manuellement le service Ollama si nécessaire

➤ Taper la commande suivante

```
ollama serve
```

Tester Ollama directement en ligne de commande avant de passer à Open WebUI

3 Télécharger et exécuter votre premier modèle LLM Mistral

➤ Taper la commande suivante

```
ollama pull mistral
```

4 Démarrer une session interactive avec le modèle LLM Mistral

➤ Taper la commande suivante

```
ollama run mistral
```

5 Saisir vos questions ou vos instructions, et le modèle générera des réponses

➤ Saisir vos questions ou vos instructions

Expliquez le réseau de neurones artificiels

► Taper la commande suivante pour quitter

```
/bye
```

ETAPE 5 : Installation de Docker et Docker compose sous debian 13.04

1 Mise à jour Debian

► Taper la commande suivante

```
apt update  
apt upgrade
```

2 Installer les packages nécessaires

► Taper la commande suivante

```
apt install -y apt-transport-https ca-certificates curl wget gnupg2
```

3 Ajoutez la clé GPG au référentiel Docker officiel

► Taper la commande suivante

```
apt install -y apt-transport-https ca-certificates curl wget gnupg2
```

4 Ajouter le dépôt officiel de Docker

► Taper la commande suivante

```
sudo echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb_release -cs) stable" |  
sudo tee /etc/apt/sources.list.d/docker.list
```

5 Mettre à jour la base de données de packages

► Taper la commande suivante

```
sudo apt update
```

6 Installation du Docker Engine, containerd, et Docker Compose

➤ Taper la commande suivante

```
sudo apt install docker-ce docker-ce-cli containerd.io
```

7 Vérifiez que le service Docker est en cours d'exécution

➤ Taper la commande suivante

```
systemctl status docker
```

8 Ajouter votre nom d'utilisateur au groupe docker

➤ Taper la commande suivante

```
sudo usermod -aG docker ausmane
```

9 Activer le démarrage automatique de Docker

➤ Taper la commande suivante

```
systemctl enable docker
```

10 Démarrer le service Docker

➤ Taper la commande suivante

```
systemctl start docker
```

11 Afficher la version de Docker

➤ Taper la commande suivante

```
docker -v
```

12 Visualiser les conteneurs actifs et inactifs

➤ Taper la commande suivante

```
docker ps -a
```

13 Afficher la version de docker compose

➤ Taper la commande suivante

ETAPE 6 : Installation de Open WebUI avec Docker

1 Récupérer l'image Open WebUI et démarrer le conteneur

► Taper la commande suivante

```
docker run -d --network=host -v open-webui :/app/backend/data -e OL-  
LAMA_BASE_URL=http://127.0.0.1:11434 --name open-webui --restart always  
ghcr.io/open-webui/open-webui:main
```



- **d** exécute le conteneur en arrière-plan
- **v open-webui :/app/backend/data** crée un volume Docker pour pérenniser vos données
- **-restart always** relance le conteneur s'il s'arrête ou si votre machine redémarre
- **-name open-webui** attribue un nom explicite au conteneur

2 Visualiser les conteneurs actifs

► Taper la commande suivante

```
docker ps -a
```

3 Premier lancement d'Open WebUI

► Ouvrir votre navigateur Web

<http://192.168.1.78:8080/auth?redirect=%2F>



Au premier accès, Open WebUI vous demande de créer un compte administrateur. Renseignez votre nom, votre e-mail et un mot de passe

Cette fenêtre s'ouvre (Figure 1)

Démarez avec Open WebUI

① Open WebUI n'établit aucune connexion externe et garde vos données en sécurité sur votre serveur local.

Nom d'utilisateur
Ausmane Bénévoles numériques de Touraine

Courriel
benoit@benoit.com

Mot de passe

Créer un compte administrateur

FIGURE 1 – Première connexion

4 Page d'accueil d'open WebUI

Cette fenêtre s'ouvre (Figure 2)

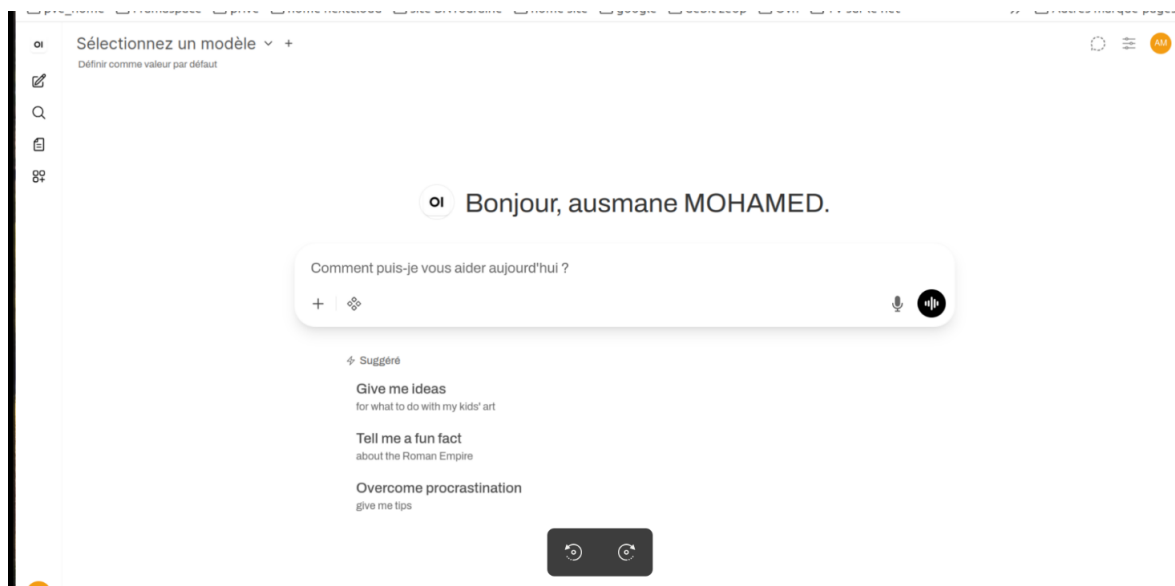


FIGURE 2 – Page d'accueil

ETAPE 7 : Connecter Open WebUI avec Ollama

1 Ouvrir un terminal et redémarrer manuellement le service Ollama si nécessaire

► Taper la commande suivante

```
ollama serve
```

2 Verifier que le modèle LLM Mistral a bien été téléchargé

► Taper la commande suivante

```
ollama list
```

Si la liste est vide, télécharger le modèle mistral avec la commande suivante : `ollama pull mistral`

3 Accéder à Open WebUI avec votre navigateur

<http://192.168.1.78:8080/auth?redirect=%2F>

Cette fenêtre s'ouvre (Figure 3)

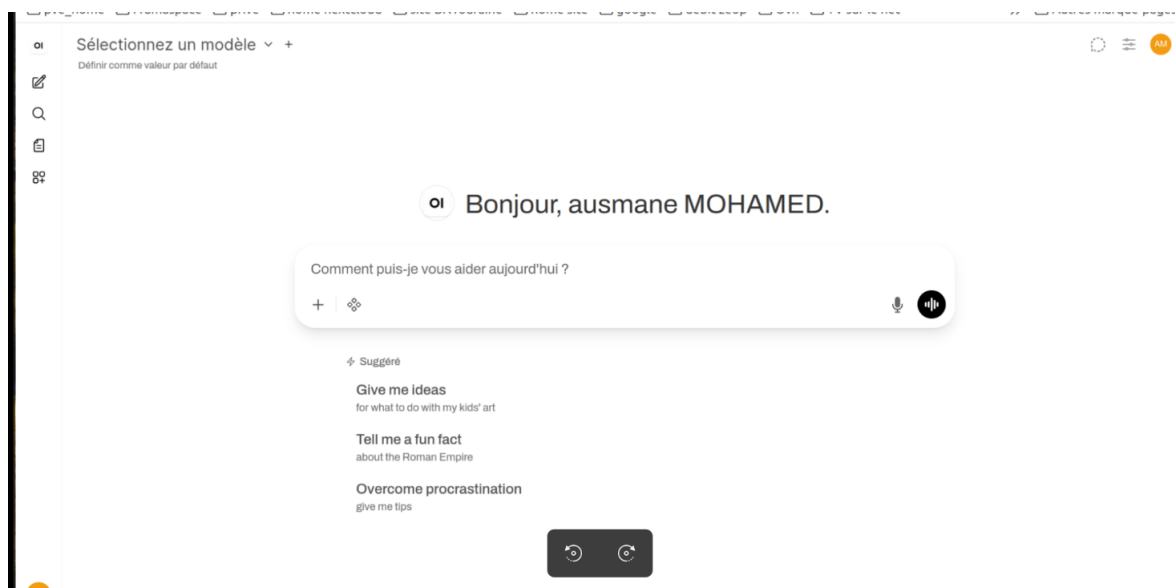


FIGURE 3 – Page d'accueil

4 Configurer l'URL de l'API Ollama

1. Cliquer sur → Réglages → Générales → Connexions

Cette fenêtre s'ouvre (Figure 4)

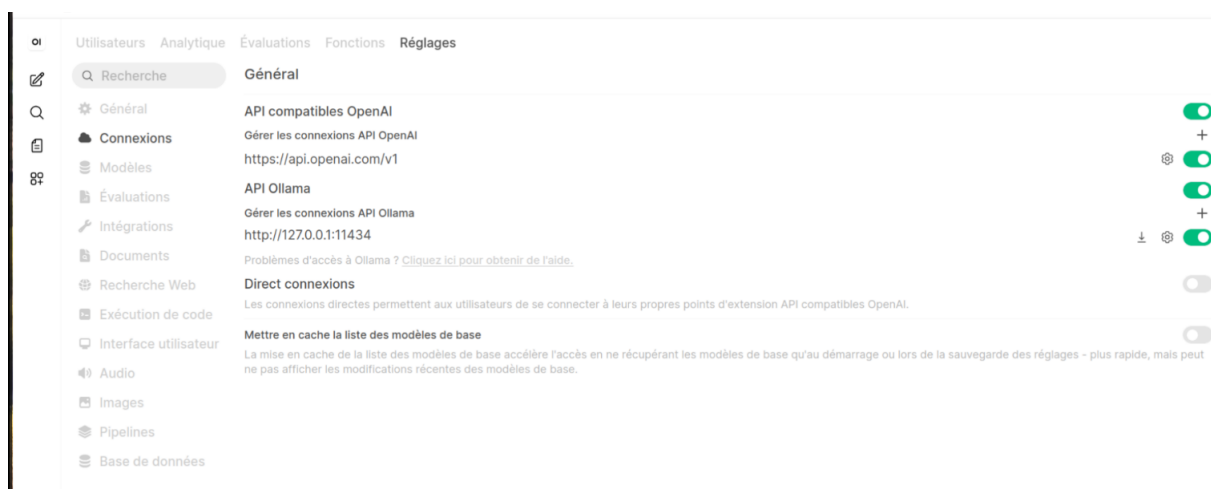


FIGURE 4 – Connexions

5 Modifier l'URL de l'API Ollama

Cette fenêtre s'ouvre (Figure 5)

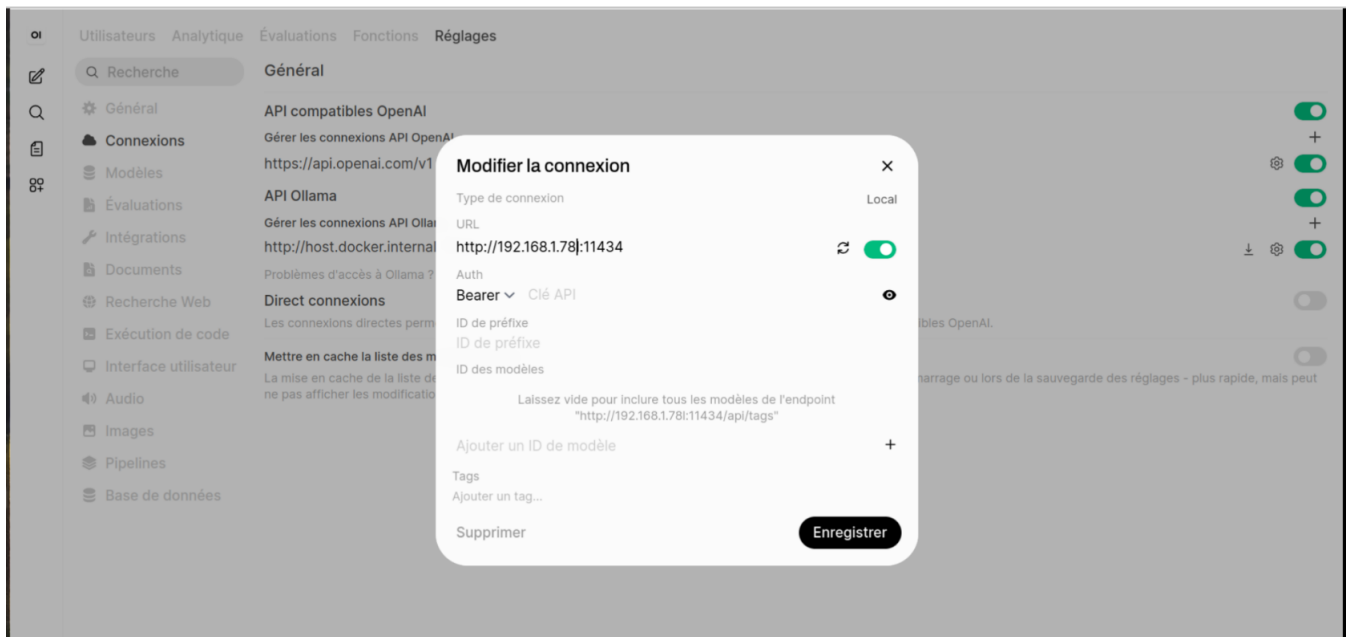


FIGURE 5 – Connexion



- Saisir l'adresse IP et le port d'écoute du serveur Ollama
- `http ://192.168.1.78 :11434`

➤ Saisir l'adresse IP et le port d'écoute du serveur Ollama `http ://192.168.1.78 :11434`

➤ Cliquer sur → Enregistrer

6 Vous pouvez maintenant utiliser Ollama avec Open WebUI

Cette fenêtre s'ouvre (Figure 4)

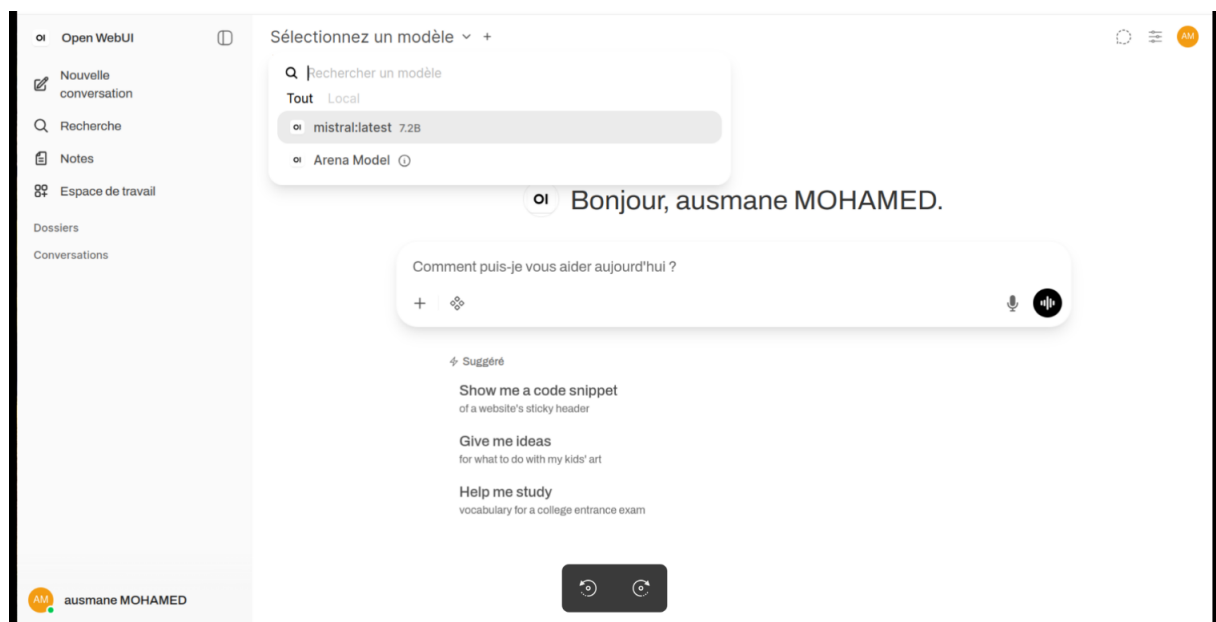


FIGURE 6 – Connexion Ollama

ANNEXE

Liens vers les logiciels libres et open source utilisés dans ce tutoriel

- **Proxmox** (Hyperviseur) : <https://www.proxmox.com/en/>
- **Debian** (système d'exploitation) : <https://www.debian.org/index.fr.html>
- **Docker** (Docker est une plateforme de conteneurisation) : <https://docs.docker.com/engine/install/>
- **Ollama** (est un outil open-source qui exécute de grands modèles de langage (LLM) en local) : <https://ollama.com/>
- **Open WebUI** (Open WebUI est une interface web open source qui permet d'exploiter des modèles de langage (LLM) sur votre instance Ollama) : <https://openwebui.com/>
- **Letsencrypt** (Une autorité de certification) : <https://letsencrypt.org/fr/>
- **Nginx Proxy Manager** (est un reverse proxy open source qui permet de rediriger le trafic Web et de gérer les certificats letsencrypt) : <https://nginxproxymanager.com/>