

NOUVEAU

TUTORIEL PRATIQUE



EXPOSER WORDPRESS DERRIERE TRAEFIK 3.5 (DOCKER COMPOSE)

De l'installation de base aux premières manipulations

POINTS CLES DU TUTORIEL



Installation pas à pas du CMS WORDPRESS avec Docker Compose



Exposition de WORDPRESS derrière Traefik 3.5



1^{ère} connexion à WORDPRESS et manipulations POST-INSTALLATION

Note :

Ce tutoriel suppose que vous ayez une machine Debian 13 ou un conteneur LXC Proxmox à disposition et que vous ayez mis en place le reverse proxy © Traefik 3.5, sur une autre machine, conformément au tutoriel présent sur notre site : <https://tutos-info.fr>.

L'objectif de ce tutoriel est d'exposer le CMS © WORDPRESS derrière le reverse proxy © Traefik. WordPress est un système de gestion de contenu (SGC ou content management system (CMS) en anglais) .

© Wordpress est gratuit et libre. Ce logiciel écrit en PHP repose sur une base de données MySQL ou MariaDB, et est distribué par la fondation WordPress sur le site wordpress.org. Les fonctionnalités de © WordPress permettent de créer et gérer différents types de sites web, de sites vitrine, site de vente en ligne, de sites applicatifs, blog, portfolio, sites institutionnels, sites d'enseignement, etc.

Pour réaliser ce tutoriel il vous faut :

- Une machine Debian 13 fonctionnelle avec 1 Go de RAM et 20 Go d'espace disque
- Ou un conteneur LXC Debian avec les mêmes caractéristiques
- Le reverse proxy © Traefik 3.5.0 installé sur une machine (voir tutoriel sur notre site : <https://tutos-info.fr>)

1 – MISE A JOUR DES PAQUETS

- Connectez-vous à votre nouvelle machine Debian (ou conteneur LXC) avec un utilisateur disposant des droits « sudo » et mettez à jour les paquets avec la commande suivante :

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

2 – INSTALLATION DE DOCKER ET DOCKER-COMPOSE AVEC LE SCRIPT OFFICIEL – SAUF SI DÉJÀ FAIT

En « mode laboratoire », il est possible d'installer Docker et docker Compose avec le script officiel. L'utilisation du script rend l'installation rapide et simple. Pour cela, saisissez les commandes suivantes :

```
apt install curl -y
curl -fsSL https://get.docker.com | sh
```

Note : Si vous travaillez en « production », il est recommandé d'ajouter votre utilisateur système au groupe docker. Ainsi, il ne sera plus nécessaire de saisir "sudo" avant toute commande. Pour cela, réalisez les étapes suivantes :

```
# Création du groupe "docker" :
groupadd docker
```

```
# Attribution du groupe à un utilisateur :
usermod -aG docker votre_nom_user
```

Déconnectez votre session Debian et reconnectez-vous.

Note : vous pouvez également installer Docker en vous aidant du site officiel et des sources officielles. Lien vers le site officiel : [Debian](#) | [Docker Docs](#)

Pour vérifier que Docker est bien installé sur votre machine Debian, saisissez la commande suivante :

```
sudo docker version
```

L'environnement Docker installé s'affiche :

```
debian@CT-DEBIAN-DOCKER:~$ sudo docker version
[sudo] password for debian:
Client: Docker Engine - Community
 Version:      28.3.3
 API version:  1.51
 Go version:   go1.24.5
 Git commit:   980b856
 Built:        Fri Jul 25 11:34:00 2025
 OS/Arch:     linux/amd64
 Context:      default

Server: Docker Engine - Community
 Engine:
  Version:      28.3.3
  API version:  1.51 (minimum version 1.24)
  Go version:   go1.24.5
  Git commit:   bea959c
  Built:        Fri Jul 25 11:34:00 2025
  OS/Arch:     linux/amd64
  Experimental: false
 containerd:
  Version:      1.7.27
  GitCommit:    05044ec0a9a75232cad458027ca83437aae3f4da
 runc:
  Version:      1.2.5
  GitCommit:    v1.2.5-0-g59923ef
 docker-init:
  Version:      0.19.0
  GitCommit:    de40ad0
```

Pour afficher la version Docker Compose installée, saisissez la commande suivante :

sudo docker compose version

```
debian@CT-DEBIAN-DOCKER:~$ sudo docker compose version
Docker Compose version v2.39.1
```

En cas de problème, vous pouvez désinstaller Docker et Docker Compose avec les commandes suivantes :

systemctl stop docker

apt purge docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

1^{ère} étape : création des dossiers et sous-dossiers

Dans cette étape, nous créons les dossiers et sous-dossiers utiles à la création de notre « stack » WORDPRESS avec les commandes suivantes (nous mettons les dossiers dans « /opt/containers » :

```
sudo mkdir -p /opt/containers/wordpress/{data/html,data/db,backup,php}
```

On se place ensuite dans le dossier adéquat avec la commande suivante :

```
cd /opt/containers/wordpress
```

2^{ème} étape : création d'un fichier « .env » (fichier caché avec les variables d'environnement)

- Créez le fichier « .env » dans « /opt/containers/wordpress » à l'aide de la commande suivante :

```
sudo nano /opt/containers/wordpress/.env
```

- Collez ou saisissez le contenu suivant :

```
# DB (changer les mots de passe)
WP_DB_NAME=wordpress
WP_DB_USER=wpuser
WP_DB_PASSWORD=ChangeMe-DB!
DB_ROOT_PASSWORD=ChangeMe-Root!
```

```
# Fuseau
TZ=Europe/Paris
```

```
# Domaine (à adapter)
WP_DOMAIN=wp.votre_domaine.fr
```

- Quittez et sauvegardez le fichier avec **CTRL + O** et pressez la touche **Entrée**.

3^{ème} étape : création du fichier « uploads.ini » pour limiter les uploads PHP

- Créez le fichier « uploads.ini » à l'aide des commandes suivantes :

```
sudo mkdir -p /opt/containers/wordpress/php
sudo nano /opt/containers/wordpress/php/uploads.ini
```

- Collez ou saisissez le contenu suivant :

```
file_uploads = On
memory_limit = 256M
upload_max_filesize = 128M
post_max_size = 128M
max_execution_time = 120
```

- Quittez et sauvegardez le fichier avec **CTRL + O** et pressez la touche **Entrée**.

4^{ème} étape : création du fichier « docker-compose.yml » avec la dernière version de WORDPRESS

- Créez le fichier « **docker-compose.yml** » dans « **/opt/containers/wordpress** » à l'aide de la commande suivante :

sudo nano /opt/containers/wordpress/docker-compose.yml

- Collez ou saisissez le contenu suivant (**le fichier commence par 3 tirets**) :

```
---

services:
  db:
    image: mariadb:10.11
    restart: unless-stopped
    environment:
      MARIADB_ROOT_PASSWORD: "${DB_ROOT_PASSWORD}"
      MARIADB_DATABASE: "${WP_DB_NAME}"
      MARIADB_USER: "${WP_DB_USER}"
      MARIADB_PASSWORD: "${WP_DB_PASSWORD}"
      TZ: "${TZ}"
    command: ["--character-set-server=utf8mb4","--collation-server=utf8mb4_unicode_ci"]
    volumes:
      - ./data/db:/var/lib/mysql
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "127.0.0.1", "-u", "${WP_DB_USER}", "--password=${WP_DB_PASSWORD}"]
      interval: 5s
      timeout: 3s
      retries: 30
      start_period: 20s

  wordpress:
    image: wordpress:php8.3-apache # (on peut prendre une version précise si besoin)
    restart: unless-stopped
    depends_on:
      db:
        condition: service_healthy
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: "${WP_DB_USER}"
      WORDPRESS_DB_PASSWORD: "${WP_DB_PASSWORD}"
      WORDPRESS_DB_NAME: "${WP_DB_NAME}"
      TZ: "${TZ}"
      # Ajouts auto dans wp-config.php pour le HTTPS derrière reverse proxy & cron réel
      WORDPRESS_CONFIG_EXTRA: |
        define('WP_HOME', 'https://${WP_DOMAIN}');
        define('WP_SITEURL', 'https://${WP_DOMAIN}');
        define('DISABLE_WP_CRON', true);
        define('FS_METHOD', 'direct');
        $_SERVER['HTTPS'] = 'on';
    volumes:
      - ./data/html:/var/www/html
      - ./php/uploads.ini:/usr/local/etc/php/conf.d/uploads.ini
    ports:
      - "8801:80" # ← accessible depuis Traefik avec port 8801 exposé
```

```
# Déclencheur cron "réel" toutes les 5 minutes
wp-cron:
  image: busybox:1.36
  restart: unless-stopped
  depends_on:
    wordpress:
      condition: service_started
  entrypoint: ["/bin/sh","-c","while true; do wget -q -O - http://wordpress/wp-cron.php?doing_wp_cron
>/dev/null 2>&1; sleep 300; done"]
```

- Quittez et sauvegardez le fichier avec **CTRL + O** et pressez la touche **Entrée**.

Note : Pourquoi WORDPRESS_CONFIG_EXTRA ?

- Force WP_HOME/SITEURL en https (sinon WordPress peut rester en http derrière un reverse proxy).
- Désactive le pseudo-cron et on le remplace par un vrai cron via le service wp-cron.
- \$_SERVER['HTTPS']='on' pour éviter les boucles/redirections.

5^{ème} étape : création du fichier de configuration dynamique sur Traefik

On crée maintenant le fichier de configuration dynamique (service et routeur) dans notre machine Traefik à l'aide de la commande suivante :

```
sudo nano /etc/traefik/dynamic/wordpress.yml
```

- Copiez ou saisissez le contenu suivant :

```
http:
  routers:
    wp-https:
      rule: Host(`wordpress.votre_domaine.fr`)
      entryPoints: [websecure]
      service: wp-svc
      tls:
        certResolver: le
      middlewares: [wp-sec]

    wp-http:
      rule: Host(`wordpress.votre_domaine.fr`)
      entryPoints: [web]
      middlewares: [redirect-to-https]
      service: wp-svc

  middlewares:
    redirect-to-https:
      redirectScheme:
        scheme: https
    wp-sec:
      headers:
        stsSeconds: 63072000
        stsIncludeSubdomains: true
        stsPreload: true
        frameDeny: true
        contentTypeNosniff: true
        referrerPolicy: no-referrer-when-downgrade
```

services:

wp-svc:

loadBalancer:

servers:

- url: "**http://ip_machine_wordpress:8801**"

passHostHeader: true

- Quittez et sauvegardez le fichier avec **CTRL + O** et pressez la touche **Entrée**.

5^{ème} étape : lancement de la création de la stack « WORDPRESS »

Sur votre machine GLPI, effectuez les manipulations suivantes :

```
cd /opt/containers/wordpress
```

```
sudo docker compose up -d
```

Patientez pendant la création de la stack complète WORDPRESS :

```
[+] Running 16/27
  wordpress [#####] Pulling
    396b1da7636e Extracting [=====] 29.77MB/2...
    ✓ 28c25be817a8 Download complete
    ✓ b9d62849df79 Download complete
    ✓ e7993c6bc3e9 Download complete
    ✓ 137dc0a47992 Download complete
    ✓ d532d4395865 Download complete
    ✓ 01b50763a979 Download complete
    ✓ 21d6cbbfe929 Download complete
```

Une fois la stack créée, vous devez obtenir ceci :

```
✓ Network wordpress_default          Created
✓ Container wordpress-db-1           Healthy
✓ Container wordpress-wordpress-1    Started
✓ Container wordpress-wp-cron-1      Started
```

Portainer affiche les nouveaux conteneurs en lien avec notre stack WORDPRESS :

☐	wordpress-db-1	healthy	   	wordpress	mariadb:10.11	2025-08-17 08:19:44	172.19.0.2	-	administrators
☐	wordpress-wordpress-1	running	   	wordpress	wordpress:php8.3-apache	2025-08-17 08:19:45	172.19.0.3	8801:80	administrators
☐	wordpress-wp-cron-1	running	   	wordpress	busybox:1.36	2025-08-17 08:19:45	172.19.0.4	-	administrators

5 – TEST DE CONNEXION A WORDPRESS

- Lancez votre navigateur et saisissez, dans la barre d'adresses, votre nom de domaine (ou votre WAN). Si vous avez tout configuré correctement, WORDPRESS affiche un assistant d'installation avec une première fenêtre demandant la sélection de la langue.
- Sélectionnez « **Français** » et cliquez le bouton bleu « **Continuer** »



- Complétez la fenêtre et cliquez le bouton bleu « **Installer Wordpress** » :



Bienvenue

Bienvenue dans la très célèbre installation en 5 minutes de WordPress ! Vous n'avez qu'à remplir les informations demandées ci-dessous et vous serez prêt à utiliser la plus extensible et puissante plateforme de publication de contenu au monde.

Informations nécessaires

Veuillez renseigner les informations suivantes. Ne vous inquiétez pas, vous pourrez les modifier plus tard.

Titre du site

Mon portfolio

Identifiant

Administrateur

Les identifiants ne peuvent utiliser que des caractères alphanumériques, des espaces, des tirets bas ("_"), des traits d'union ("-"), des points et le symbole @.

Mot de passe

●●●●●●●●●●●●●●●●

Forte

 Afficher

Important : Vous aurez besoin de ce mot de passe pour vous connecter. Pensez à le stocker dans un lieu sûr.

Votre e-mail

bts.sio@ik.me

Vérifiez bien cette adresse e-mail avant de continuer.

Visibilité par les moteurs de recherche

☐ Demander aux moteurs de recherche de ne pas indexer ce site

Certains moteurs de recherche peuvent décider de l'indexer malgré tout.

Installer Wordpress

Une nouvelle fenêtre affiche la réussite de l'installation ; cliquez le lien bleu « **Se connecter** » :



Quel succès !

WordPress est installé. Merci et profitez bien !

Identifiant

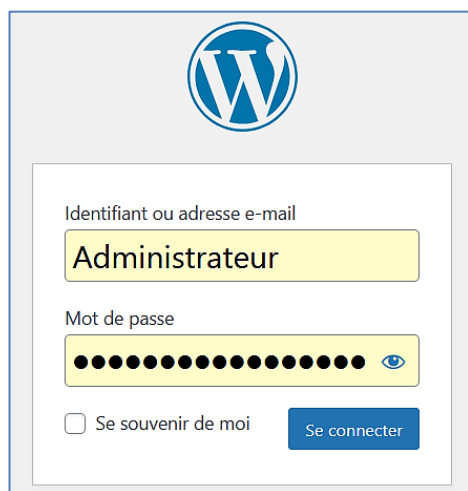
Administrateur

Mot de passe

Le mot de passe que vous avez choisi.

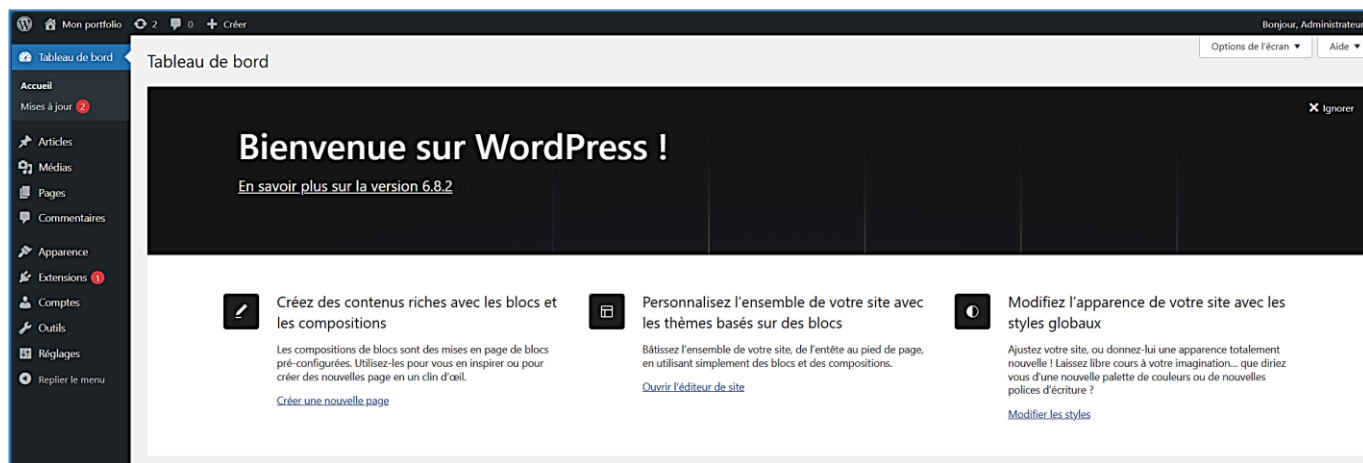
[Se connecter](#)

- Authentifiez-vous avec les indications saisies lors de l'étape précédente afin d'accéder au tableau de bord de Wordpress puis cliquez « **Se connecter** » :

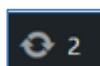


WordPress login form. At the top is the WordPress logo. Below it, a text input field labeled "Identifiant ou adresse e-mail" contains the text "Administrateur". Below that is a password input field labeled "Mot de passe" with a series of dots and an eye icon. At the bottom left is a checkbox labeled "Se souvenir de moi". At the bottom right is a blue button labeled "Se connecter".

Le tableau de bord de Wordpress s'affiche :



Il faudra mettre à jour les extensions et faire les mises à jour de Wordpress pour éviter toutes failles de sécurité. Pour cela commencez par faire la mise à jour de sécurité en cliquant sur ce symbole en haut du tableau de bord :



Note : pour l'utilisation de Wordpress, consultez notre tutoriel présent sur notre site : <https://tutos-info.fr>

FICHIER « .env » - SUR MACHINE WORDPRESS

```
# DB (changer les mots de passe)
WP_DB_NAME=wordpress
WP_DB_USER=wpuser
WP_DB_PASSWORD=ChangeMe-DB!
DB_ROOT_PASSWORD=ChangeMe-Root!
```

```
# Fuseau
TZ=Europe/Paris
```

```
# Domaine (à adapter)
WP_DOMAIN=wp.votre_domaine.fr
```

FICHIER « uploads.ini » - SUR MACHINE WORDPRESS

```
file_uploads = On
memory_limit = 256M
upload_max_filesize = 128M
post_max_size = 128M
max_execution_time = 120
```

FICHIER « docker-compose.yml » - SUR MACHINE WORDPRESS

```
---
services:
  db:
    image: mariadb:10.11
    restart: unless-stopped
    environment:
      MARIADB_ROOT_PASSWORD: "${DB_ROOT_PASSWORD}"
      MARIADB_DATABASE: "${WP_DB_NAME}"
      MARIADB_USER: "${WP_DB_USER}"
      MARIADB_PASSWORD: "${WP_DB_PASSWORD}"
      TZ: "${TZ}"
    command: ["--character-set-server=utf8mb4","--collation-server=utf8mb4_unicode_ci"]
    volumes:
      - ./data/db:/var/lib/mysql
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "127.0.0.1", "-u", "${WP_DB_USER}", "--password=${WP_DB_PASSWORD}"]
      interval: 5s
      timeout: 3s
      retries: 30
      start_period: 20s

  wordpress:
    image: wordpress:php8.3-apache # (on peut prendre une version précise si besoin)
    restart: unless-stopped
    depends_on:
      db:
        condition: service_healthy
    environment:
      WORDPRESS_DB_HOST: db:3306
```

```

WORDPRESS_DB_USER: "${WP_DB_USER}"
WORDPRESS_DB_PASSWORD: "${WP_DB_PASSWORD}"
WORDPRESS_DB_NAME: "${WP_DB_NAME}"
TZ: "${TZ}"
# Ajouts auto dans wp-config.php pour le HTTPS derrière reverse proxy & cron réel
WORDPRESS_CONFIG_EXTRA: |
    define('WP_HOME', 'https://${WP_DOMAIN}');
    define('WP_SITEURL', 'https://${WP_DOMAIN}');
    define('DISABLE_WP_CRON', true);
    define('FS_METHOD', 'direct');
    $$_SERVER['HTTPS'] = 'on';
volumes:
- ./data/html:/var/www/html
- ./php/uploads.ini:/usr/local/etc/php/conf.d/uploads.ini
ports:
- "8801:80" # ← accessible depuis Traefik avec port 8801 exposé

# Déclencheur cron "réel" toutes les 5 minutes
wp-cron:
  image: busybox:1.36
  restart: unless-stopped
  depends_on:
    wordpress:
      condition: service_started
  entrypoint: ["bin/sh", "-c", "while true; do wget -q -O - http://wordpress/wp-cron.php?doing_wp_cron
>/dev/null 2>&1; sleep 300; done"]

```

FICHER « `wordpress.yml` » - SUR MACHINE TRAEFIK

```

http:
  routers:
    wp-https:
      rule: Host(`wordpress.votre_domaine.fr`)
      entryPoints: [websecure]
      service: wp-svc
      tls:
        certResolver: le
      middlewares: [wp-sec]

    wp-http:
      rule: Host(`wordpress.votre_domaine.fr`)
      entryPoints: [web]
      middlewares: [redirect-to-https]
      service: wp-svc

  middlewares:
    redirect-to-https:
      redirectScheme:
        scheme: https
    wp-sec:
      headers:
        stsSeconds: 63072000
        stsIncludeSubdomains: true
        stsPreload: true
        frameDeny: true
        contentTypeNosniff: true

```

referrerPolicy: no-referrer-when-downgrade

services:

wp-svc:

loadBalancer:

servers:

- url: "**http://ip_machine_wordpress:8801**"

passHostHeader: true