TUTOS-INFO
L'informatique par l'exemple

c

TUTORIEL

TRAEFIK
Installer Wordpress

SOMMAIRE

EXPERT
DIFFICULTE

1 – CREATION DU DOCKER-COMPOSE POUR WORDPRESS

Ce tutoriel présente l'installation d'une **stack Wordpress** qui fonctionnera **derrière le reverse proxy Traefik** (version 2.10). ***Nous supposons, ici, que votre reverse proxy Traefik 2.10 est déjà installé*** et fonctionnel (voir tutoriel « Installer le reverse proxy Traefik » : https://cloud.tutos-info.fr/s/LFYRJi87srdR4Kj).

1ère étape : préparation de l'environnement Wordpress

- Commencez par créer 1 dossier « wpress » avec deux sous-dossiers « db » et « html » dans l'arborescence Docker initiale (pour rappel nous avons créé une arborescence Docker dans « /srv ») :

```
cd /
mkdir -p /srv/wpress /srv/wpress/db /srv/wpress/html
```

- Créez un fichier « uploads.ini » dans le dossier « wpress » :

```
cd /srv/wpress
nano uploads.ini
```

- Saisissez ces 2 lignes dans le fichier « uploads.ini » et enregistrez les modifications dans le fichier :

```
upload_max_filesize = 512M
post_max_size = 512M
```

2ème étape : création du fichier « docker-compose.yml » pour Wordpress

- Créez un fichier « docker-compose.yml » dans le dossier « wpress » :

```
cd /srv/wpress
nano docker-compose.yml
```

- Saisissez le contenu suivant dans le fichier « docker-compose.yml » et enregistrez-le :

```
---
services:
  wordpress-db:
    container_name: wordpress-db
    image: mysql:latest
    networks:
      - backend
    volumes:
      - /srv/wpress/db:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: passroot
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: passwordpress

  wordpress:
    container_name: wordpress
    depends_on:
      - wordpress-db
```

Ici, on créé un conteneur pour la base de données nommé « wordpress-db » basé sur la dernière version de mysql. On indique que le réseau utilisé sera le « backend » (un réseau interne et non exposé sur Internet).

Ici, on « monte » le dossier /srv/wpress/db dans le conteneur mysql.

Ici, on indique les variables d'environnement (à modifier !) qui seront utilisées pour se connecter à la base de données lors de l'installation du CMS.

Ici, on créé le conteneur « wordress » avec un lien avec la base de données (« depends_on »).

```
    image: wordpress:latest
    networks:
      - traefik-proxy
      - backend
    restart: always
    volumes:
      - /srv/wpress/uploads.ini:/usr/local/etc/php/conf.d/uploads.ini
      - /srv/wpress/html:/var/www/html
    environment:
      WORDPRESS_DB_HOST: wordpress-db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: passwordpress
      WORDPRESS_DB_NAME: wordpress

networks:
  traefik-proxy:
    external: true

  backend:
```

Ici, on indique que le conteneur Wordpress aura deux cartes réseau : une sera exposée sur Internet derrière le reverse proxy Traefik et l'autre utilisera le réseau interne « backend » pour communiquer avec le serveur mysql.

2 – AJOUT DU ROUTEUR ET DU SERVICE WORDPRESS DANS TRAEFIK

Dans cette partie, nous allons ajouter notre stack Wordpress derrière le reverse proxy Traefik 2.10 sans utiliser les « labels » mais **en utilisant un fichier de configuration dynamique**.

Quand on utilise Traefik comme reverse proxy et que l'on souhaite ajouter un service, il est possible d'utiliser 2 méthodes :

- 1ère méthode : ajouter des « labels » dans le fichier « docker-compose.yml » de Traefik
- 2ème méthode : utiliser un fichier de configuration dynamique pour le nouveau service

L'inconvénient principal d'utiliser la 1ère méthode dite des « labels » est que vous devez ajouter ces labels dans le fichier « docker-compose.yml » original de Traefik ce qui, à la longue, va surcharger le fichier et, surtout, vous devrez relancer la stack via la commande « docker compose up -d ».

En utilisant la 2ème méthode, nous allons créer un fichier « yml » dans lequel nous allons déclarer le routeur et le service pour la stack en question sans avoir à ajouter un label et relancer la stack. Traefik va instantanément interpréter ce fichier dynamique et appliquer les règles de routage définies !

Nous présentons cette 2ème méthode car il s'agit, pour nous, de la meilleure façon d'utiliser Traefik compte tenu du fait qu'il a été écrit en Go et qu'il est capable d'interpréter « à la volée » les changements qu'on lui impose.

- Placez-vous dans le dossier « /srv/traefik/dynamic » initialement créé dans l'arborescence originale de Traefik (voir tutoriel précédent)

cd /srv/traefik/dynamic

- Créez un fichier « wpress.yml » par exemple :

nano wpress.yml

- L'éditeur s'ouvre : copiez le contenu suivant (voir page suivante) :

```
http:
 services:
  wordpress: # on déclare ici le service
   loadBalancer:
    servers:
     - url: "http://wordpress:80" # nom du conteneur Docker avec le port 80 exposé par défaut

 routers:
  wordpress-secure:
   rule: "Host(`wptest.sio-ndlp.fr`)"
   entryPoints:
    - "https"
   middlewares:
    - "SecureHeaders@file"
   service: "wordpress@file"
   tls:
    certResolver: letsencrypt
```

Ici, on commence par déclarer le **SERVICE** « wordpress » en indiquant que ce service correspond au conteneur Docker nommé « wordpress » et dont le port « 80 » du conteneur est exposé par défaut.

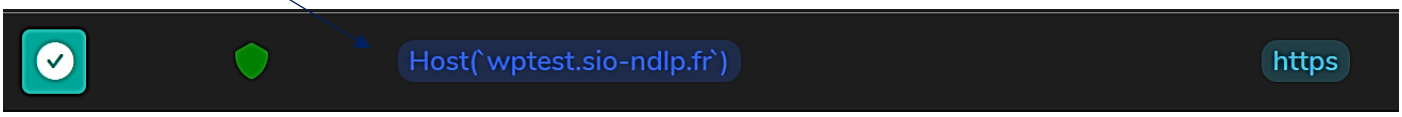
Ici, on déclare le **ROUTEUR** « wordpress-secure » en indiquant le sous-domaine à utiliser, le point d'entrée « https » (443) et l'application du middleware « SecureHeaders » et la génération d'un certificat Let's Encrypt.

Une fois le contenu copié, quittez et enregistrez le fichier : la configuration de ce nouveau service et de ce nouveau routeur dans Traefik est déjà prise en compte !

Il est temps de lancer la création de notre environnement Wordpress avec la commande « docker compose up -d ».

```
cd /srv/wpress
docker compose up -d
```

L'environnement Wordpress se créé et si l'on se connecte au tableau de bord de Traefik, on constate que le routeur est bien apparu :



Le service est, lui aussi, apparu :

Avec cette méthode dite « dynamique » on constate les attraits de ce reverse proxy des temps modernes ! Un fichier de configuration permet, à lui seul, d'ajouter les routeurs et les services derrière Traefik en quelques lignes.

Lancez votre navigateur et connectez-vous avec l'adresse définie au niveau du routeur (***"Host(`wptest.sio-ndlp.fr`)"*** : la page d'installation de Wordpress s'affiche ! Il ne reste plus qu'à suivre les étapes d'installation de l'assistant Wordpress !

Conservez une copie du fichier « wpress.yml » qui vous a servi pour configurer le nouveau service et son routeur ; il pourra être adapté pour une autre configuration !