



c

TRAEFIK

Installer GLPI 10

SOMMAIRE

1. **CREATION DU DOCKER-COMPOSE POUR GLPI 10**
 - a. Préparation de l'environnement
 - b. Création du fichier « docker-compose.yml »
2. **AJOUT DU ROUTEUR ET DU SERVICE « GLPI 10 » DANS TRAEFIK (avec un fichier de configuration dynamique)**
 - a. Préparation du fichier de configuration dynamique
 - b. Ajout d'un service
 - c. Ajout d'un routeur
 - d. Tests

© tutos-info.fr - 05/2023

DIFFICULTE



UTILISATION COMMERCIALE INTERDITE

1 – CREATION DU DOCKER-COMPOSE POUR GLPI 10

Ce tutoriel présente l'installation d'une **stack GLPI 10** 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>).

1^{ère} étape : préparation de l'environnement GLPI 10

- Commencez par créer 1 dossier « glpi » dans l'arborescence Docker initiale (pour rappel nous avons créé une arborescence Docker dans « /srv ») :

```
cd /  
mkdir -p /srv/glpi
```

- Créez un fichier « mariadb.env » dans le dossier glpi, ajoutez les variables d'environnement ci-dessous (vous pouvez, bien entendu, modifier les réponses) et enregistrez le fichier:

```
nano /srv/glpi/mariadb.env
```

```
MARIADB_ROOT_PASSWORD=passroot  
MARIADB_DATABASE=glpidb  
MARIADB_USER=glpi  
MARIADB_PASSWORD=passglpi
```

2^{ème} étape : création du fichier « docker-compose.yml » pour GLPI

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

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

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

```
---  
services:  
# Conteneur MARIADB  
mariadb:  
  image: mariadb:latest  
  hostname: mariadb  
  networks:  
  - backend  
  volumes:  
  - /var/lib/mysql:/var/lib/mysql  
  env_file:  
  - ./mariadb.env  
  restart: always  
# Conteneur GLPI 10  
glpi:  
  image: diouxx/glpi  
  container_name: glpi  
  hostname: glpi
```

On commence, ici, par créer un 1^{er} conteneur pour MariaDB et on connecte MariaDB sur le réseau interne (non exposé sur Internet) nommé « backend ».

Ici, on stipule que MariaDB doit utiliser le fichier contenant les variables d'environnement (utilisateurs et mots de passe pour la base de données).

Ici, on crée le conteneur GLPI basé sur une image bien connue sur le Docker Hub (diouxx/glpi).

networks:

- **traefik-proxy**
- **backend**

Ici, on connecte le conteneur GLPI sur le réseau Traefik (« traefik-proxy ») mais aussi sur le réseau interne « backend » afin que l'on puisse communiquer avec le serveur MariaDB.

volumes:

- **/etc/timezone:/etc/timezone:ro**
- **/etc/localtime:/etc/localtime:ro**
- **/var/www/html/glpi:/var/www/html/glpi**

Ici, on a monté les volumes utiles pour le serveur web Apache qui fonctionne avec GLPI.

environment:

- **TIMEZONE=Europe/Paris**

restart: always

networks:

traefik-proxy:

external: true

backend:

2 – AJOUT DU ROUTEUR ET DU SERVICE GLPI DANS TRAEFIK

Dans cette partie, nous allons ajouter notre stack GLPI 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 :

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

L'inconvénient principal d'utiliser la 1^{ère} 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 2^{ème} 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 2^{ème} 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 « glpi.yml » par exemple :

nano glpi.yml

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

http:

services:

glpi: # on déclare ici le service

loadBalancer:

servers:

- url: "http://glpi:80" # nom du conteneur Docker

Ici, on ajoute un nouveau **SERVICE** « glpi » en spécifiant l'adresse du conteneur GLPI (le port 80 est exposé par défaut pour GLPI).

routers:

glpi-secure:

rule: "Host(`glpittest.sio-ndlp.fr`)"

entryPoints:

- "https"

middlewares:

- "SecureHeaders@file"

service: "glpi@file"

tls:

certResolver: letsencrypt

Ici, on ajoute un nouveau **ROUTEUR** « glpi-secure » en spécifiant l'adresse URL du conteneur GLPI (notre sous-domaine hébergé) et on applique le middleware de sécurité SecureHeaders et on demande l'obtention d'un certificat par 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 GLPI avec la commande « docker compose up -d ».

cd /srv/glpi

docker compose up -d

L'environnement GLPI se crée 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(`glpittest.sio-ndlp.fr`)**") : la page d'installation de GLPI 10 s'affiche ! Il ne reste plus qu'à suivre les étapes d'installation de l'assistant GLPI 10 (voir autres tutoriels sur notre site).

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