



c

TRAEFIK

Installer PortainerCE

SOMMAIRE

1. CREATION DU DOCKER-COMPOSE POUR PORTAINER CE
 - a. Préparation de l'environnement
 - b. Création du fichier « docker-compose.yml »
2. AJOUT DU ROUTEUR ET DU SERVICE « PORTAINER » 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 PORTAINER CE

Ce tutoriel présente l'installation d'une **stack Portainer CE** 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 Portainer

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

```
cd /  
mkdir -p /srv/portainer
```

- Créez un volume portainer_data dans Docker :

```
docker volume create portainer_data
```

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

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

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

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

```
---  
services:  
  portainer:  
    container_name: portainer  
    image: "portainer/portainer-ce:latest"  
    networks:  
      - traefik-proxy  
    restart: unless-stopped  
    volumes:  
      - /var/run/docker.sock:/var/run/docker.sock  
      - portainer_data:/data  
  
volumes:  
  portainer_data:  
  
networks:  
  traefik-proxy:  
    external: true
```

On commence par déclarer le nouveau service Portainer (conteneur) en s'appuyant sur la dernière version de Portainer CE.

On indique que le réseau utilisé par le conteneur sera celui de Traefik (nom du réseau = traefik-proxy).

On indique à Portainer qu'il faut monter le volume « portainer_data » créé dans Docker.

2 – AJOUT DU ROUTEUR ET DU SERVICE PORTAINER DANS TRAEFIK

Dans cette partie, nous allons ajouter notre stack Portainer 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 « portainer.yml » par exemple :

nano portainer.yml

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

http:

services:

portainer: # on déclare ici le service Portainer

loadBalancer:

servers:

- url: "http://portainer:9000" # nom du conteneur avec son port de fonctionnement 9000 par défaut

On définit, ici, le **SERVICE** Portainer qui renvoie vers l'URL du conteneur. Attention, pour Portainer CE, il faut utiliser le port « 9000 » par défaut (et non le 80).

routers:

portainer-secure:

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

entryPoints:

- "https"

middlewares:

- "SecureHeaders@file"

service: "portainer@file"

tls:

certResolver: letsencrypt

On définit, ici, le **ROUTEUR** Portainer qui renvoie vers l'URL du sous-domaine hébergé (le « host »). On applique le middleware de sécurité SecureHeaders et on demande à Let's Encrypt de nous fournir un certificat valide.

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 Portainer CE avec la commande « docker compose up -d ».

cd /srv/portainer

docker compose up -d

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

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