

Mise en place d'un serveur Traefik (Reverse Proxy)

Prérequis :

Premièrement, à des fins de tests, nous allons générer l'image docker suivante :

Dockefile:

```
# Docker Who display him name
#
# Created by : Néhémie Barkia
# Last Update : 26/12/2023
#
# PRÉREQUIS
FROM debian:latest
RUN apt-get update -y
RUN apt-get install apache2 php php-pdo php-mbstring php-mysql php-json php-common git -y
RUN apt-get clean
# INSTALLATION WEB
RUN rm /var/www/html/* -r
# Insertion des informations de connexion dans les fichiers de Fix
RUN echo "<?php" >> /var/www/html/index.php
RUN echo "\$h = getenv('HOSTNAME');" >> /var/www/html/index.php
RUN echo "echo \$h" >> /var/www/html/index.php
RUN echo "?>" >> /var/www/html/index.php
#
WORKDIR /app
VOLUME /app/logs
EXPOSE 80
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

Création de l'image :

```
docker build -t web_name .
```

Nous venons de créer une image nommée web_name. C'est un serveur web qui renvoie simplement le nom du conteneur qui l'exécute.

Modification du fichier HOSTS de linux :

```
127.0.1.1    traefik.localhost web.traefik.localhost
```

Installation et configuration de traefik :

/srv/traefik.toml :

```
[entryPoints]
  [entryPoints.http]
    address = ":80"
  [entryPoints.https]
    address = ":443"

[api]

[providers.docker]
  endpoint = "unix:///var/run/docker.sock"
```

docker-compose.yml :

```
version: '3'

services:
  reverse-proxy:
    image: traefik:v2.9
    command: --api.insecure=true --providers.docker
    ports:
      - "80:80"
      - "8080:8080"
    volumes:
      - /srv/traefik.toml:/etc/traefik/traefik.toml
```

```
- /var/run/docker.sock:/var/run/docker.sock
```

labels:

```
- "traefik.http.routers.api.rule=Host(`traefik.localhost`)"  
- "traefik.http.routers.api.service=api@internal"  
- "traefik.http.routers.api.entrypoints=http"
```

web:

```
image: web_name
```

ports:

```
- "8081-8181:80"
```

labels:

```
- "traefik.enable=true"  
- "traefik.http.routers.web.rule=Host(`web.traefik.localhost`)"  
- "traefik.http.routers.web.entrypoints=http"
```

Création de l'infrastructure :

```
docker-compose up
```

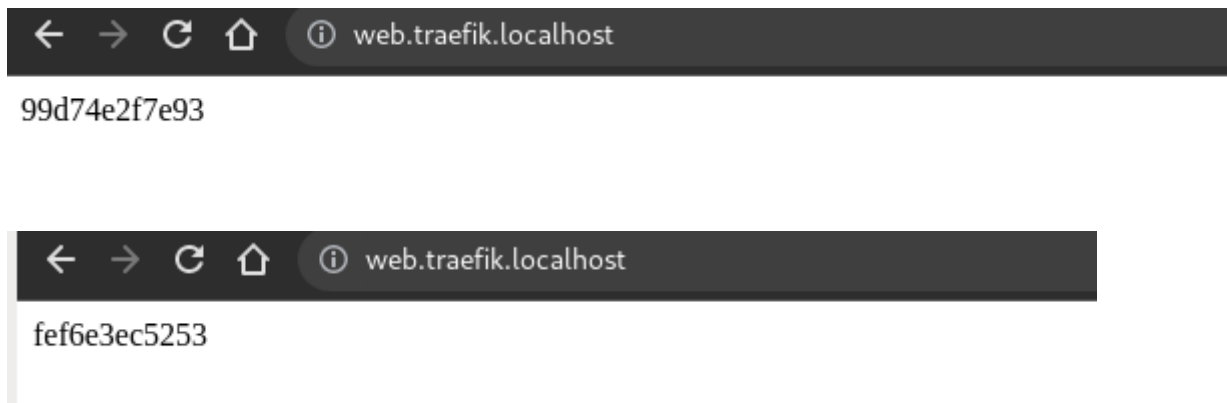
Augmentation du nombre de serveurs :

```
docker-compose scale web=10
```

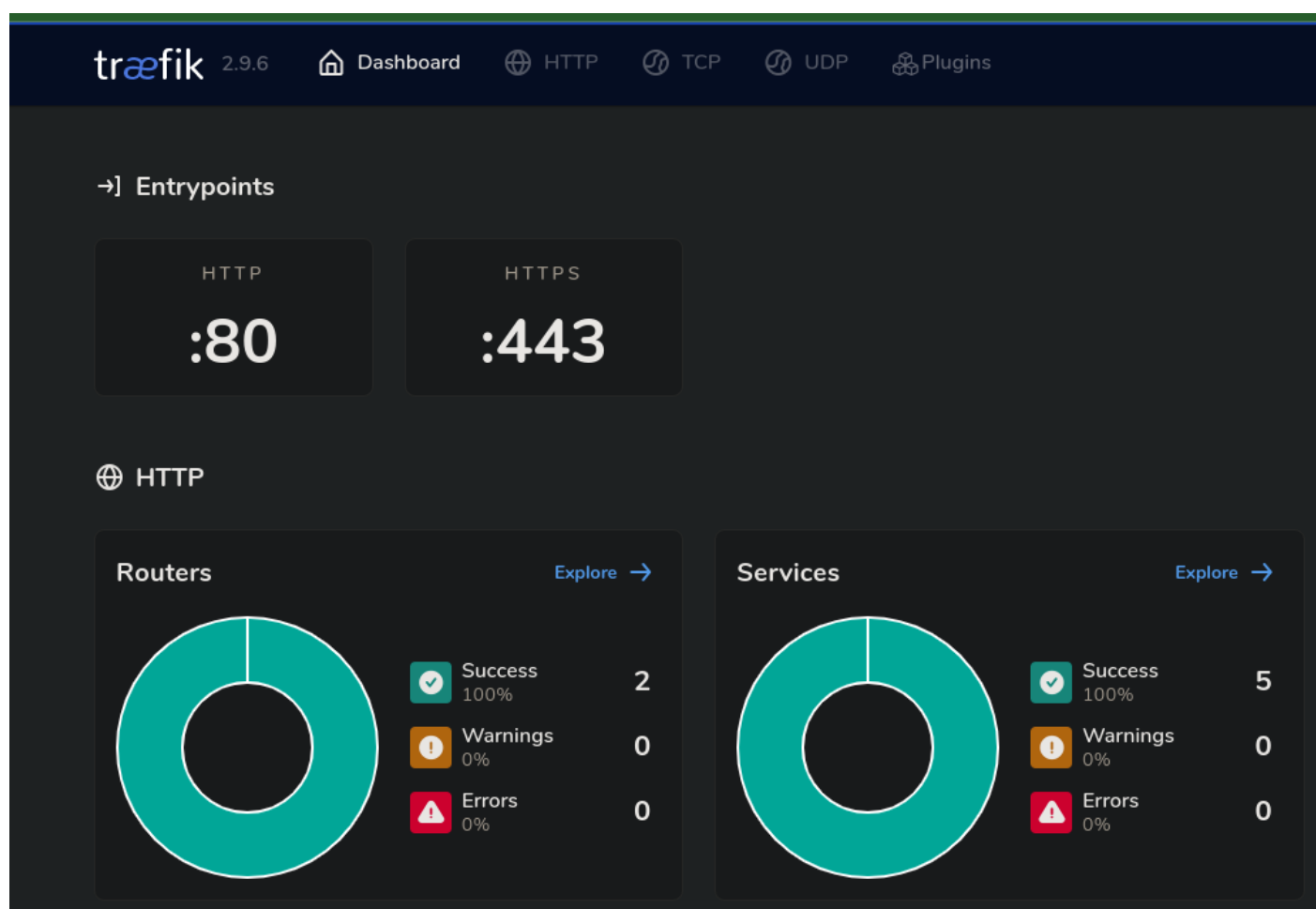
```
WARNING: The scale command is deprecated. Use the up command with the --scale flag instead.  
WARNING: The "web" service specifies a port on the host. If multiple containers for this service are created on a single host, the port will clash.  
Creating traefik_web_2 ... done  
Creating traefik_web_3 ... done  
Creating traefik_web_4 ... done  
Creating traefik_web_5 ... done  
Creating traefik_web_6 ... done  
Creating traefik_web_7 ... done  
Creating traefik_web_8 ... done  
Creating traefik_web_9 ... done  
Creating traefik_web_10 ... done
```

Ici nous dupliquons neuf fois le serveur web. Vu que leurs ports sont compris dans le fichier de config du service. Ceux-ci vont être intégrés dans le cluster et ainsi répondre aux requêtes envoyées à [web.traefik.localhost](#) (au travers du proxy traefik évidemment).

Résultat côté utilisateur :



Dans l'interface d'administration :



On peut afficher la liste des serveurs rattachés à un service :

Status	Name	Type	Servers	Provider
	api@internal		0	
	dashboard@internal		0	
	noop@internal		0	
	reverse-proxy-traefik@docker	loadbalancer	1	
	web-traefik@docker	loadbalancer	100	

On constate qu'il y a cent serveur rattaché à mon cluster (**si vous voulez scaler + de cent serveurs, modifier la pool de port dans le docker-compose**).

Revision #8

Created 26 December 2022 16:20:42 by Nehemie

Updated 26 December 2022 16:52:22 by Nehemie