

Mettre en place Traefik avec Docker

Traefik est un Reverse Proxy qui écoutera les ports 80 et 443. Par conséquent les serveurs web tels que Apache ou Nginx ne pourront pas fonctionner en même temps.

Prérequis :



- Avoir un nom de domaine
 - Dans notre cas nous utiliserons le nom de domaine : `miraceti.net`

Configuration nom de domaine

Grâce à notre nom de domaine, nous allons pouvoir créer des sous domaine (en CNAME `A`).

Sur le site de gestion de notre nom de domaine respectif, dans la section DNS, ajouter l'entrée suivante :



- sub.domaine.ext Type ip.adress.server
 - Soit
- traefik.miraceti.net A 12.34.56.78

Préparation des fichiers de configuration

Création du fichier `docker-compose.yml` dans un dossier **traefik** :

```
networks:  
  traefik-network:
```

external: true

volumes:

traefik-certificates:

services:

traefik:

image: \${TRAEFIK_IMAGE_TAG}

container_name: traefik

command:

- "--log.level=\${TRAEFIK_LOG_LEVEL}"
- "--accesslog=true"
- "--api.dashboard=true"
- "--api.insecure=true"
- "--ping=true"
- "--ping.entrypoint=ping"
- "--entryPoints.ping.address=:8082"
- "--entryPoints.web.address=:80"
- "--entryPoints.websecure.address=:443"
- "--providers.docker=true"
- "--providers.docker.endpoint=unix:///var/run/docker.sock"
- "--providers.docker.exposedByDefault=false"
- "--certificatesresolvers.letsencrypt.acme.tlschallenge=true"
- "--certificatesresolvers.letsencrypt.acme.email=\${TRAEFIK_ACME_EMAIL}"
- "--certificatesresolvers.letsencrypt.acme.storage=/etc/traefik/acme/acme.json"
- "--metrics.prometheus=true"
- "--metrics.prometheus.buckets=0.1,0.3,1.2,5.0"
- "--global.checkNewVersion=true"
- "--global.sendAnonymousUsage=false"

volumes:

- /var/run/docker.sock:/var/run/docker.sock
- traefik-certificates:/etc/traefik/acme

networks:

- traefik-network

ports:

- "80:80"
- "443:443"

healthcheck:

test: ["CMD", "wget", "http://localhost:8082/ping", "--spider"]

```

interval: 10s
timeout: 5s
retries: 3
start_period: 5s
labels:
  - "traefik.enable=true"
  - "traefik.http.routers.dashboard.rule=Host(`${TRAEFIK_HOSTNAME}`)"
  - "traefik.http.routers.dashboard.service=api@internal"
  - "traefik.http.routers.dashboard.entrypoints=websecure"
  - "traefik.http.services.dashboard.loadbalancer.server.port=8080"
  - "traefik.http.routers.dashboard.tls=true"
  - "traefik.http.routers.dashboard.tls.certresolver=letsencrypt"
  - "traefik.http.services.dashboard.loadbalancer.passhostheader=true"
  - "traefik.http.routers.dashboard.middlewares=authtraefik"
  - "traefik.http.middlewares.authtraefik.basicauth.users=${TRAEFIK_BASIC_AUTH}"
  - "traefik.http.routers.http-catchall.rule=HostRegexp(`${host:.+}`)"
  - "traefik.http.routers.http-catchall.entrypoints=web"
  - "traefik.http.routers.http-catchall.middlewares=redirect-to-https"
  - "traefik.http.middlewares.redirect-to-https.redirectscheme.scheme=https"
restart: unless-stopped

```

Création du fichier `.env` dans le même dossier :

```

# Traefik Variables
TRAEFIK_IMAGE_TAG=traefik:3.1
TRAEFIK_LOG_LEVEL=WARN
TRAEFIK_ACME_EMAIL=my.adress@mail.com
TRAEFIK_HOSTNAME=traefik.miraceti.net
# Basic Authentication for Traefik Dashboard
# Username: traefikadmin
# Password:
# Passwords must be encoded using MD5, SHA1, or BCrypt https://hostingcanada.org/htpasswd-generator/
TRAEFIK_BASIC_AUTH='traefikadmin:$2y$10$smzJfirKC75x/hVpiINeZ0iSm.Jkity9cn4KwNkRv07hSQVfc5a
ze'

```

Création du network pour Traefik

Nous allons créer un network "traefik-network" pour lier tous les conteneurs/services qui auront besoin d'écouter les ports 80 et 443.

```
docker network create traefik-network
```

Déploiement du docker compose

```
docker compose up -d
```

Accéder au Dashboard

Revision #4

Created 19 July 2024 07:13:09 by gpatruno

Updated 19 July 2024 07:38:28 by gpatruno