? Réseau, reverse proxy avec Traefik

Maintenant, quand on veut pouvoir avoir plusieurs services sur un même port avec des noms de domaines différents. On doit utiliser un reverse-proxy, on peut par exemple utiliser `nginx` mais ce n'est pas pratique car cela sort de l'écosystème Docker.

Donc à la place on va utiliser `traefik`. Et on va faire ces configurations dans des docker-compose par soucis de simplicité et de lisibilité.

De plus traefik a l'avantage d'automatiquement renouveler les certificats HTTPS.

? HTTP (non sécurisé)

```
version: '3'

services:
# On crée le service traefik
  traefik:
    image: "traefik:latest"

    # On spécifie certains paramètres dans la commande
    command:
      # On défini que c'est docker que l'on utilise
      - "--providers.docker=true"
      # On dit à traefik de ne router des conteneurs que quand on dit qu'il faut le faire
      - "--providers.docker.exposedbydefault=false"
      # Et on défini un entrypoint (là où les utilisateurs vont se connecter) au port 80
appellé "web"
      - "--entrypoints.web.address=:80"

    # On expose le port 80
    ports:
      - "80:80"

    # Et on donne accès à traefik à docker en lecture seule
```

```
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock:ro"

  # Ici c'est un simple service de test
  whoami:
    image: "traefik/whoami"
    # Les labels permettent de mettre des "étiquettes" sur les conteneurs, c'est ceux ci que
traefik recherche
    labels:
      # On dit à traefik de s'occuper de ce conteneur
      - "traefik.enable=true"
      # On dit à traefik de router ceci vers "mon-nom-de-domaine.net"
      - "traefik.http.routers.whoami.rule=Host(`mon-nom-de-domaine.net`)"
      # Et on lui dit aussi d'utiliser l'entrypoint web défini plus tot
      - "traefik.http.routers.whoami.entrypoints=web"
```

Maintenant on peut faire `sudo docker-compose up` et aller sur `http://mon-nom-de-domaine.net` pour voir le site.

? HTTPS (sécurisé)

```
version: '3'

services:
  traefik:
    image: "traefik:latest"
    command:
      - "--api.insecure=true"
      - "--providers.docker=true"
      - "--providers.docker.exposedbydefault=false"
      - "--entrypoints.web.address=:80"

      # On ajoute un nouveau entrypoint sur le port 443 appellé "websecure"
      - "--entrypoints.websecure.address=:443"
      # On dit que l'on va ajouter un resolver de certificat en utilisant les challenge http
appellé "myresolver"
      # Par défault on va utiliser letsencrypt
      - "--certificatesresolvers.myresolver.acme.httpchallenge=true"
      # On lui donne comme entrypoint celui qui est sur le port 80
```

```
      - "--certificatesresolvers.myresolver.acme.httpchallenge.entrypoint=web"

    # On expose les ports 80 et 443
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock"
  whoami:
    image: "traefik/whoami"
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.whoami.rule=Host(`mon-nom-de-domaine.net`)"
      # On change le entrypoint pour être celui de 443 (websecure)
      - "traefik.http.routers.whoami.entrypoints=websecure"
      # Et on défini le resolver comme étant "myresolver"
      - "traefik.http.routers.whoami.tls.certresolver=myresolver"
```

Pareil ici en faisant `sudo docker-compose up` on va maintenant pouvoir aller sur `https://mon-nom-de-domaine.net` et y voir le site.

? Comment utiliser traefik pour plusieurs docker-compose différents ?

Quelque chose que je n'ai pas précisé c'est que quand on utilise docker-compose, tous les services de celui ci sont dans le même "réseau" docker, qui est isolé du reste.

Seulement on ne peut avoir qu'une seule fois traefik car les ports ne peuvent être bind qu'une seule fois. Mais traefik a aussi besoin que tous les services au quel il doit accéder soit dans le même réseau que lui.

Donc pour utiliser plusieurs docker-compose avec un seul traefik, on va avoir besoin de définir un réseau.

```
# Ici on crée un réseau appellé "traefik"
sudo docker network create traefik
```

Maintenant on peut définir ce réseau (externe car défini en avance) dans nos docker-compose

Premier (pour traefik) :

```
version: '3'

services:
  traefik:
    image: "traefik:latest"
    command:
      - "--api.insecure=true"
      - "--providers.docker=true"
      - "--providers.docker.exposedbydefault=false"
      - "--entrypoints.web.address=:80"

      - "--entrypoints.websecure.address=:443"
      - "--certificatesresolvers.myresolver.acme.httpchallenge=true"
      - "--certificatesresolvers.myresolver.acme.httpchallenge.entrypoint=web"

    ports:
      - "80:80"
      - "443:443"
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock"

    # On lie ce service au réseau "traefik"
    networks:
      - traefik

# On précise que le réseau traefik est externe et non pas interne au docker-compose
networks:
  traefik:
    external: true
```

Et le deuxième pour "whoami"

```
version: '3'

services:
  whoami:
    image: "traefik/whoami"
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.whoami.rule=Host(`mon-nom-de-domaine.net`)"
```

```
      - "traefik.http.routers.whoami.entrypoints=websecure"
      - "traefik.http.routers.whoami.tls.certresolver=myresolver"

    # On lie ce service au réseau "traefik"
    networks:
      - traefik

# On précise que le réseau "traefik" est externe et non pas interne au docker-compose
networks:
  traefik:
    external: true
```

Maintenant on peut lancer les deux docker-compose et tout devrait tout de même fonctionner comme avant □□

> Note: Par défault docker-compose crée un réseau pour les services qui y sont, mais quand vous définissez manuellement un réseau comme ici, ce n'est pas le cas. Donc si vous souhaitez que les services communiquent entre eux, mettez les également dans le réseau `traefik`.

? Petits ajouts

- Comment faire pour qu'un port 9000 d'un service devienne le port 80 par exemple ? Pour faire de la redirection de ports on peut utiliser ceci

```
- "traefik.http.services.NOMDUSERVICE.loadbalancer.server.port=9000" # On défini le port du service comme étant 9000
- "traefik.http.routers.NOMDUSERVICE.entrypoints=NOMDELENTRYPOINT" # On le lie à l'entrypoint de :80
```

- Comment lier un service à un hôte tel que `http://mon-nom-de-domaine.net/hello` (avec un préfixe) :

```
- "traefik.http.routers.NOMDUSERVICE.rule=(Host(`mon-nom-de-domaine.net`) && PathPrefix(`/hello`))"
```

- Comment faire une redirection du http vers https (http étant le port 80 donc l'entrypoint "web" et https étant le port 443 et donc l'entrypoint "websecure")

```
- "--entrypoints.web.http.redirections.entrypoint.to=websecure"
- "--entrypoints.web.http.redirections.entrypoint.scheme=https"
```

Revision #1
Created 26 April 2023 17:09:39 by SnowCode
Updated 26 April 2023 17:09:39 by SnowCode