

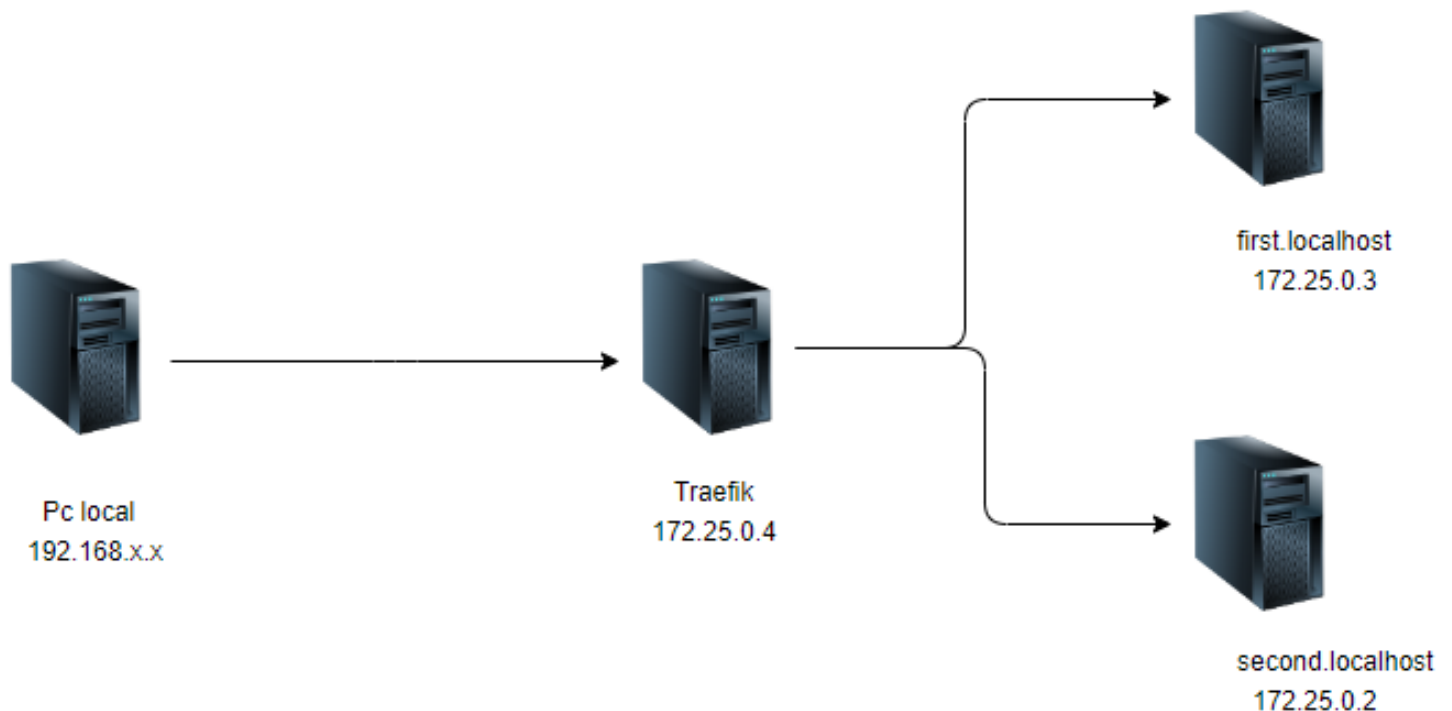
Installation d'un Reverse proxy "Traefik"

Mission : déployer des conteneurs docker et traefik:

- déploiement de traefik en tant que reverse proxy
- au moins 2 conteneurs "containers/whoami" doivent être déployés
- les containers viennent s'enregistrer auprès de traefik
- mapping des ports dynamiques entre traefik et les containers whoami
- chaque conteneur doit être capable de répondre à son propre domaine (ex: container 1 doit répondre à first.localhost et container 2 à second.localhost)

reverse-proxy : est un type de serveur, habituellement placé en frontal de serveurs web. Le reverse-proxy permet à un utilisateur d'Internet d'accéder à des serveurs internes. Une des applications courantes du proxy inverse est la répartition de charge (load-balancing). L'utilisateur du Web passe par son intermédiaire pour accéder aux applications de serveurs internes.

Quand le Pc local fait une requête sur un conteneur, il est redirigé sur le reverse proxy, et le reverse-proxy fait la requête



Images Utilisées

Nous utiliserons Traefik en tant que reverse-proxy et 2 serveurs TinyGo “whoami” qui renvoient les requêtes en html.

Configuration des conteneurs

55 lines (48 sloc) | 1.9 KB

Raw

Blame

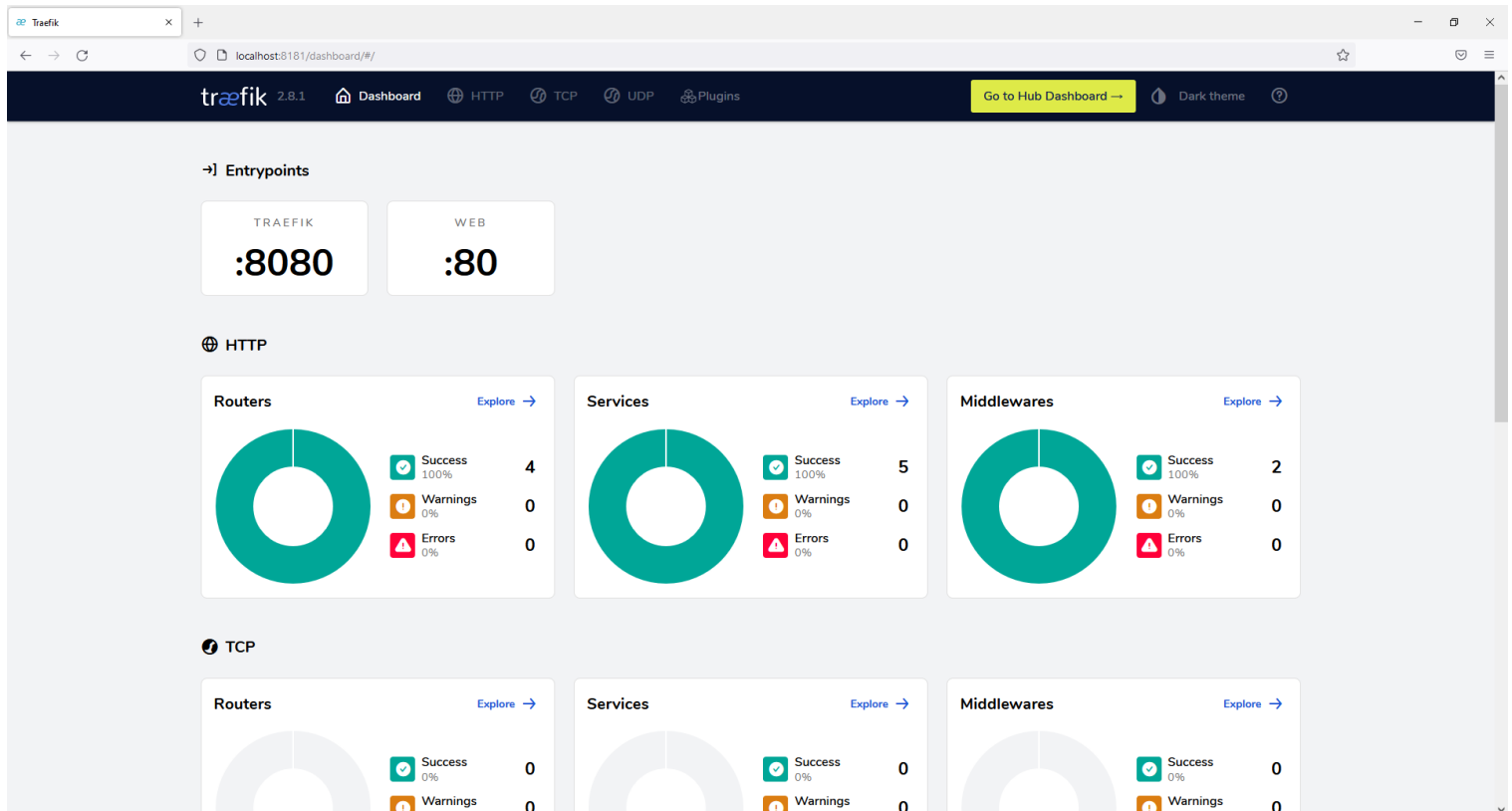


```
1 version: "3.8"
2
3 services:
4   traefik:
5     deploy:
6       labels:
7         traefik.http.routers.traefik.rule: "Host(`traefik.localhost`)" #référencement DNS
8         traefik.docker.network: "traefik-benlab"
9         traefik.enable: "true"
10        traefik.http.routers.traefik.entrypoints: "http"
11        traefik.http.routers.traefik.service: "api@internal"
12    image: "traefik:v2.8.1" #"traefik:latest"
13    networks:
14      - traefik-benlab
15    container_name: "traefik"
16    restart: unless-stopped #cela permet que le conteneur ne se lance pas à l'exécution de docker desktop si le service est arrêté.
17    command: #alternative est la création d'un fichier de config
18      #- "--log.level=DEBUG"
19      - "--api.insecure=true" #accès au dashboard
20      - "--providers.docker=true" #déclaration de docker pour le backend (docker provider)
21      - "--providers.docker.exposedbydefault=false"
22      - "--entrypoints.web.address=:80" #point d'entrée http
23    ports:
24      - "80:80"
25      - "8181:8080" #modification de la redirection déjà un conteneur sur 8080 en local.
26    volumes:
27      - "/var/run/docker.sock:/var/run/docker.sock:ro"
28      - "./config.yml:/etc/traefik/config.yml:ro" #config dynamique
29      #- "./traefik.yml:/etc/traefik/traefik.yml" #config static
30
31
32    whoami1:
33      image: "containous/whoami"
34      networks:
35        - traefik-benlab
36      container_name: "whoami1"
37      labels:
38        - "traefik.enable=true"
39        - "traefik.http.routers.whoami1.rule=Host(`first.localhost`)" #déclaration du nom pour le référencement DNS
40        - "traefik.http.routers.whoami1.entrypoints=web" #Point d'entrée pour la redirection
41
42    whoami2:
43      image: "containous/whoami"
44      networks:
45        - traefik-benlab
46      container_name: "whoami2"
47      labels:
48        - "traefik.enable=true"
49        - "traefik.http.routers.whoami2.rule=Host(`second.localhost`)"
50        - "traefik.http.routers.whoami2.entrypoints=web"
51
52    networks:
53      traefik-benlab:
```

```
ben@DESKTOP-ACBFNCF:~/traefik$ docker compose up -d
```

```
ben@DESKTOP-ACBFNCF:~/traefik$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
4d0fb2b05f18   traefik:v2.8.1 "/entrypoint.sh --ap..." About an hour ago Up About an hour 0.0.0.0:80->80/tcp, 0.0.0.0:8181->8080/tcp traefik
d34d3f33d8bd   containous/whoami "/whoami"              About an hour ago Up About an hour 80/tcp                             whoami2
f6a721db0762   containous/whoami "/whoami"              About an hour ago Up About an hour 80/tcp                             whoami1
```

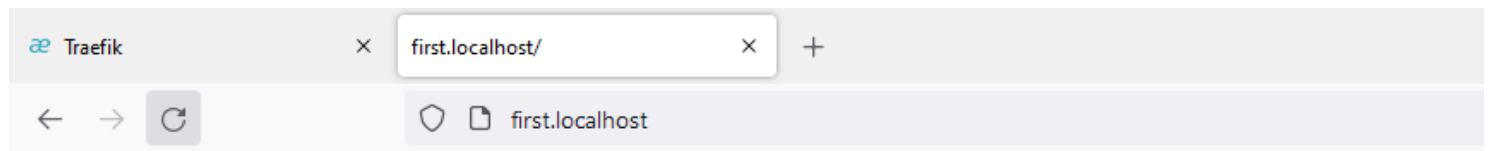
Les conteneurs sont bien présents.



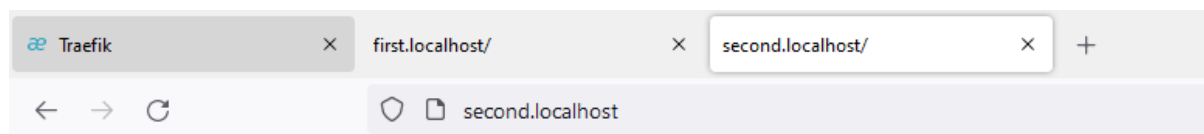
Le portail traefik est accessible sur l'adresse <http://localhost:8181>

HTTP Routers 4 HTTP Services 5 HTTP Middlewares 2

All Status						
Status	TLS	Rule	Entrypoints	Name	Service	Provider
✓		PathPrefix(`/api`)	traefik	api@internal	api@internal	traefik
✓		PathPrefix(`/`)	traefik	dashboard@internal	dashboard@internal	traefik
✓		Host(`first.localhost`)	web	whoami1@docker	whoami1-traefik	docker
✓		Host(`second.localhost`)	web	whoami2@docker	whoami2-traefik	docker



```
Hostname: f6a721db0762
IP: 127.0.0.1
IP: 172.25.0.3
RemoteAddr: 172.25.0.4:45954
GET / HTTP/1.1
Host: first.localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 172.25.0.1
X-Forwarded-Host: first.localhost
X-Forwarded-Port: 80
X-Forwarded-Proto: http
X-Forwarded-Server: 4d0fb2b05f18
X-Real-Ip: 172.25.0.1
```



```
Hostname: d34d3f33d8bd
IP: 127.0.0.1
IP: 172.25.0.2
RemoteAddr: 172.25.0.4:36984
GET / HTTP/1.1
Host: second.localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 172.25.0.1
X-Forwarded-Host: second.localhost
X-Forwarded-Port: 80
X-Forwarded-Proto: http
X-Forwarded-Server: 4d0fb2b05f18
X-Real-Ip: 172.25.0.1
```

On peut bien sûr vérifier sur les requêtes affichées par les conteneurs whoami sont émises par le reverse-proxy Traefik avec l'adresse 172.25.0.4 et non avec l'ip de mon pc local (192.168.x.x)