

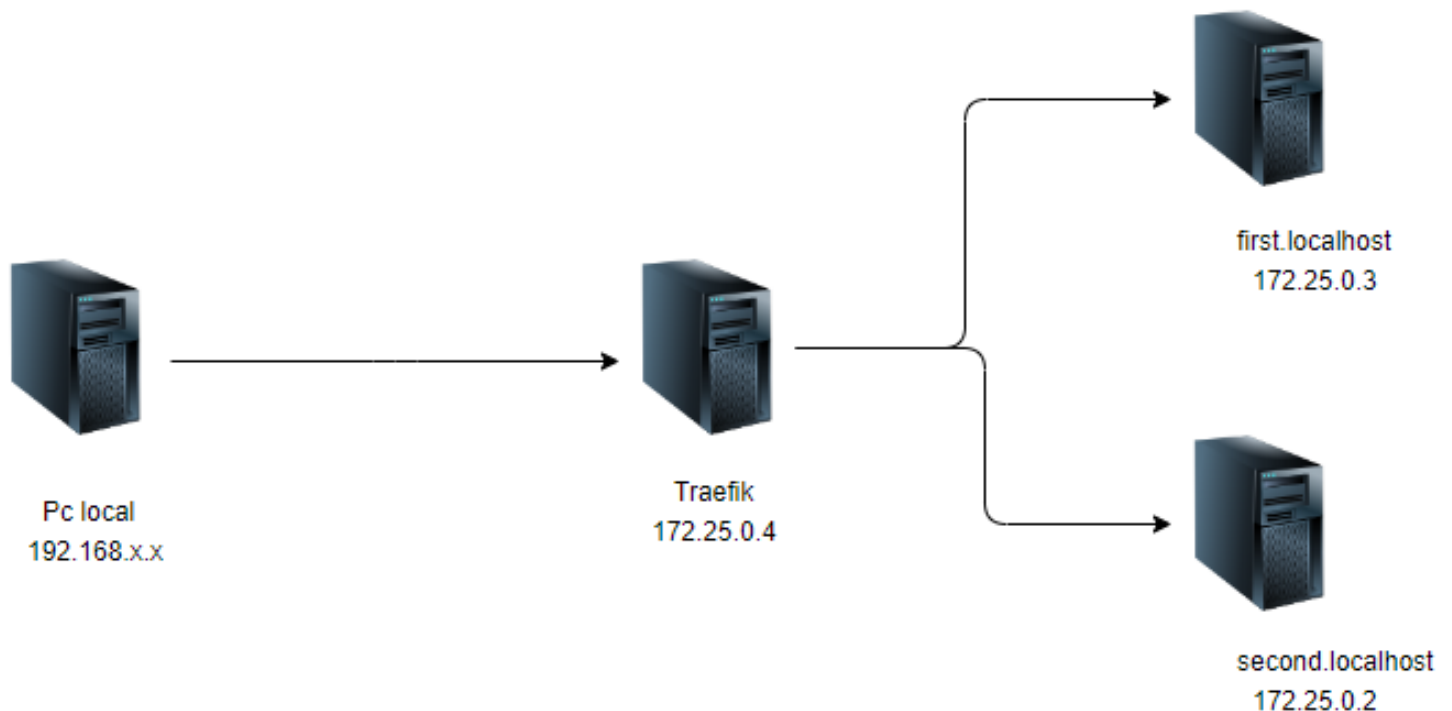
# Installation d'un Reverse proxy "Traefik"

**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

### docker-compose.yml

```
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
13     image: "traefik:v2.8.1" #"traefik:latest"
14     networks:
15       - traefik-benlab
16     container_name: "traefik"
17     restart: unless-stopped #cela permet que le conteneur ne se lance pas à l'exécution de docker desktop si le service est arrêté.
18     command:
19       - "--log.level=DEBUG"
20       # Les commandes dessous permette de bypass le fichier traefik.yml
21       #- "--api.insecure=true" #accès au dashboard
22       #- "--providers.docker=true" #déclaration de docker pour le backend (docker provider)
23       #- "--providers.docker.exposedbydefault=false"
24       #- "--entrypoints.web.address=:80" #point d'entrée http
25     ports:
26       - "80:80"
27       - "8181:8080"
28     volumes:
29       - "/var/run/docker.sock:/var/run/docker.sock:ro"
30       - "./config_dynamique:/etc/traefik/config_dynamique/:ro" #dossier des configs dynamiques
31       - "./traefik.yml:/etc/traefik/traefik.yml:ro" #config static ne sera pas uti
32
33
34     whoami1:
35       image: "containous/whoami"
36       networks:
37         - traefik-benlab
38       container_name: "whoami1"
39       labels:
40         - "traefik.enable=true"
41         - "traefik.http.routers.whoami1.rule=Host('first.localhost')" #déclaration du nom pour le référencement DNS
42         - "traefik.http.routers.whoami1.entrypoints=web" #Point d'entrée pour la redirection
43
44     whoami2:
45       image: "containous/whoami"
46       networks:
47         - traefik-benlab
48       container_name: "whoami2"
49       labels:
50         - "traefik.enable=true"
51         - "traefik.http.routers.whoami2.rule=Host('second.localhost')"
52         - "traefik.http.routers.whoami2.entrypoints=web"
53
54     networks:
55       traefik-benlab:
```

## traefik.yml (fichier de configuration statique)

19 lines (14 sloc) | 426 Bytes

Raw Blame



```
1  api:
2    insecure: true
3
4  entryPoints:
5    web:
6      address: ':80'
7
8
9  providers:
10   docker:
11     endpoint: "unix:///var/run/docker.sock"
12     exposedByDefault: false
13     network: "traefik-benlab"
14     watch: true
15   file:
16     filename: "/etc/traefik/config_dynamique/config.yml" #Fichier de config dynamique
17     watch: true
18
19
```

## config.yml (fichier de configuration dynamique)

```
1  http:
2    routers:
3      api:
4        rule: "Host(`traefik.localhost`)"
5        service: api@internal
6        middlewares:
7          - basic-auth
8    middlewares:
9      basic-auth:
10        basicAuth:
11          users:
12            - "ben:$apr1$yN5SkU9i$/.72ViNpEiXNikGnMFNBV0" #ben 123456
13        removeHeader: true
```

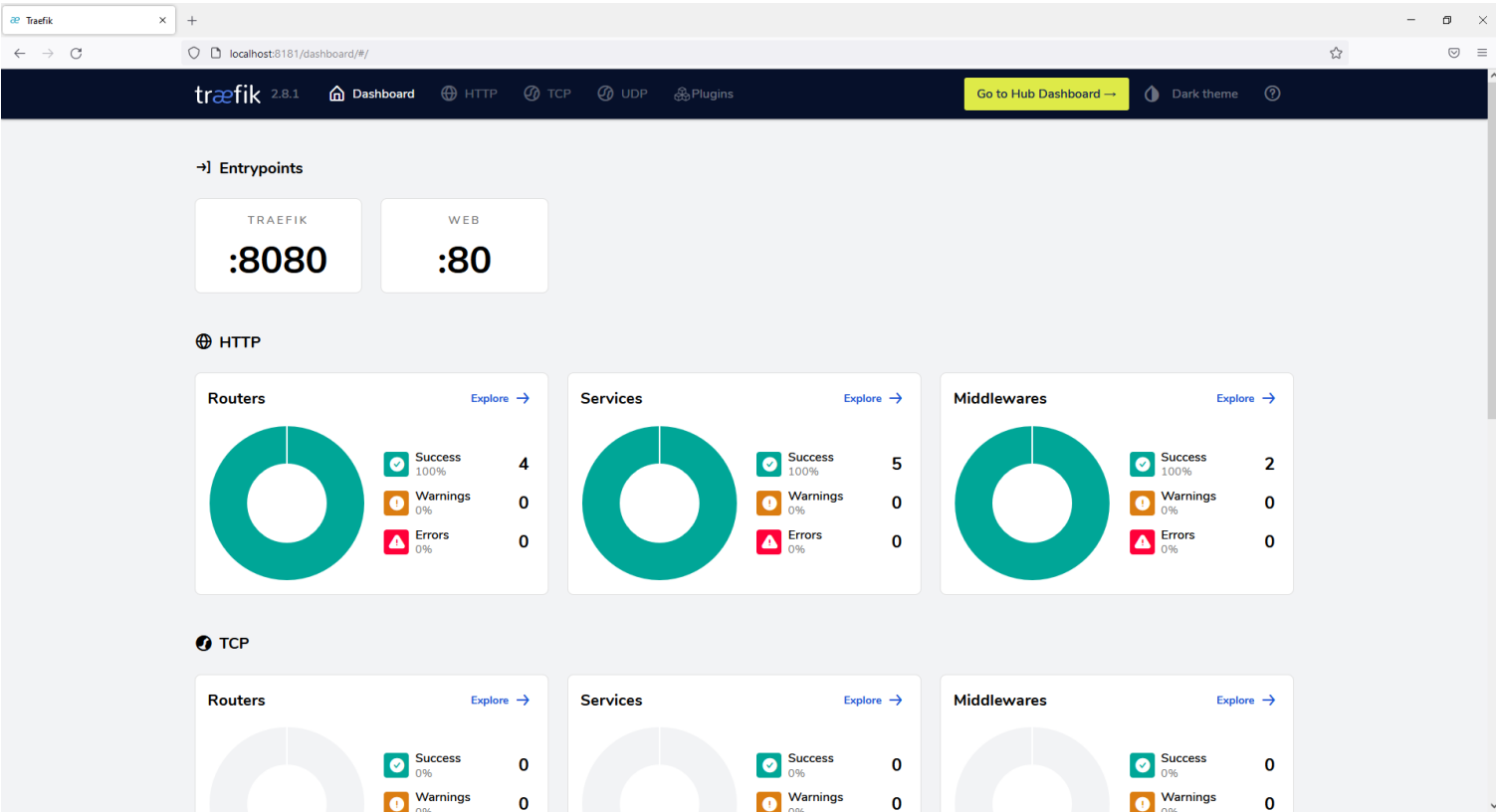
Ce fichier permet d'ajouter un middleware pour l'authentification et autre ajouts comme le TLS, etc

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

```
ben@DESKTOP-ACBFNCF:~/traefik$ docker ps
```

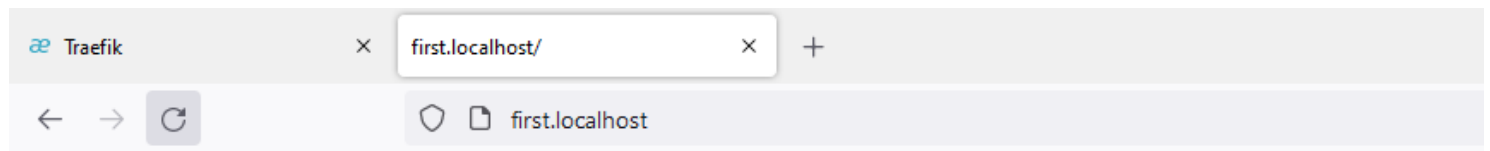
| CONTAINER ID | IMAGE             | COMMAND                  | CREATED           | STATUS           | PORTS                                      | NAMES   |
|--------------|-------------------|--------------------------|-------------------|------------------|--|---------|
| 4d0fb2b05f18 | traefik:v2.8.1    | "/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.

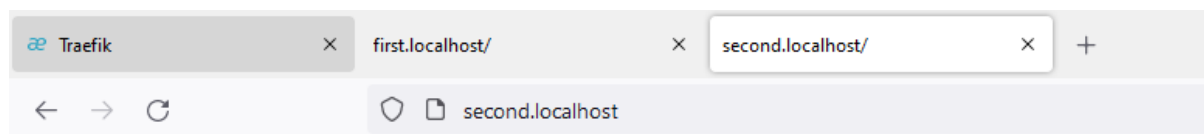


All Status Success Warnings Errors Search

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



```
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)