Installation

Traefik lässt sich bequem als Docker Container bereitstellen. Es wird nur ein Container für Traefik selbst benötigt, keine weiteren Zusatzdienste sind notwendig.

Eine einfache voll funktionsfähige Konfiguration als Docker Compose sieht z. B. wie folgt aus. Diese ist sehr gut für Testumgebungen geeignet, um neue Images auszuprobieren und diese auf einem Host unter einem Port (hier 80) bereitzustellen.

> Für ein produktives Setup sollte unbedingt noch Let's Encrypt mit konfiguriert werden. Auch wenn diese Konfiguration bereits reicht, um Dienste anzubieten, so währen diese jetzt noch unverschlüsselt und somit unsicher. Außerdem ist in diesem Fall mit api.insecure die Konfiguration über Port 8080 einsehbar, was ein Sicherheitsrisiko bedeutet.

```
services:
  traefik:
    image: "traefik"
    container_name: "traefik"
    command:
      #- "--log.level=DEBUG"
      - "--api.insecure=true"
      - "--providers.docker=true"
      - "--providers.docker.exposedbydefault=false"
      - "--entrypoints.web.address=:80"
    ports:
      - "80:80"
      - "8080:8080"
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock:ro"
```

Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach nach gewählter Installation `sudo docker-compose up -d` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegeben Optionen. Sollten die Container bereits mit dieser Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

Damit wäre Traefik nun einsatzbereit und unter http://localhost:8080 die Konfigurationsoberfläche erreichbar.

Reverse Proxy Netzwerk konfigurieren

Um nun Dienste über den Traefik erreichbar zu machen, muss dieser die Dienste innerhalb des Docker Hosts erreichen können. Da Docker für jede Compose Datei ein eigenes Default Netzwerk konfiguriert, kann Traefik diese nicht erreichen. Auch die Option alle Dienste zusammen mit Traefik in einer Compose Datei zu beschreiben, ist keine gute Idee.

Somit erstellen wir uns nun ein Netzwerk, welches unabhängig von den Compose Dateien verwaltet wird. In diesem Beispiel wird das Netzwerk mit dem Namen *traefik* erstellt. Da keine weiteren Optionen angegeben werden, ist es ein Bridge Netzwerk, welches für unseren Fall gut geeignet ist.

```
sudo docker network create traefik
```

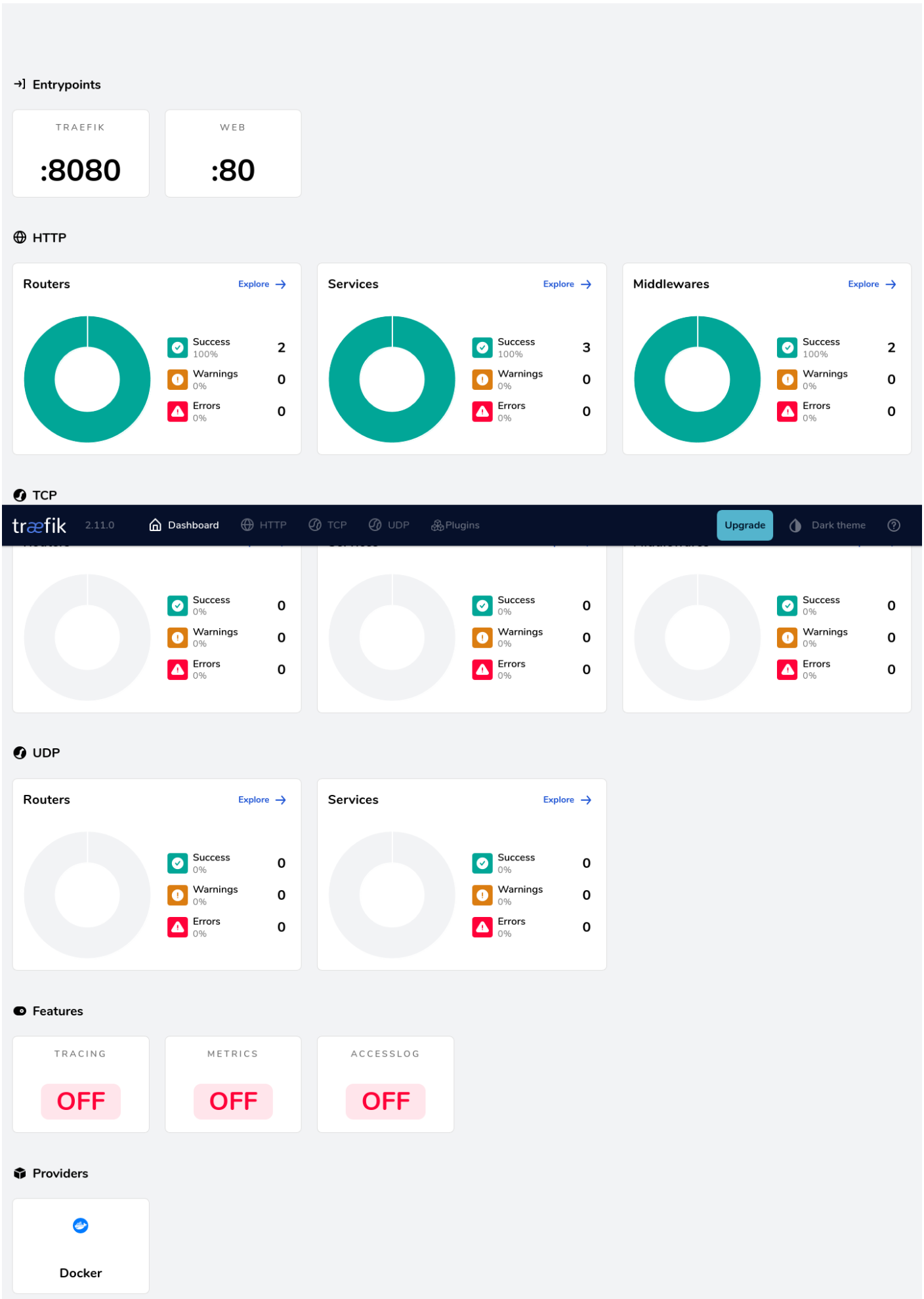
Nun wird die bisherige Compose Konfiguration für den Container Traefik um folgende Zeilen erweitert, damit Traefik in das Netzwerk aufgenommen wird.

```
    command:
      - "--providers.docker.network=traefik"
    networks:
      - traefik
networks:
  traefik:
    name: traefik
    external: true
```

Zum Schluss den Container mit *docker compose up -d* neu erstellen lassen und fertig. Ab jetzt können einfach weitere Container in dieses Netzwerk aufgenommen werden, damit Traefik sie erreichen und anbieten kann.

Konfiguration einsehen

Über den Port 8080 lässt sich die Konfiguration anzeigen: http://localhost:8080. Es wird ein Dashboard angezeigt, welches alle erkannten Dienste, Regeln usw. anzeigt. Hier kann nun erkundet werden, ob alles richtig übernommen wurde. Für produktive Umgebungen, die öffentlich erreichbar sind, sollte dieses Dashboard deaktiviert werden, da die gesamte Konfiguration gezeigt wird und alles offen legt.


Entrypoints
TRAEFIK
:8080
WEB
:80
HTTP
Routers
Explore
Success
100%
2
Warnings
0%
0
Errors
0%
0
Services
Explore
Success
100%
3
Warnings
0%
0
Errors
0%
0
Middlewares
Explore
Success
100%
2
Warnings
0%
0
Errors
0%
0
TCP
traefik
2.11.0
Dashboard
HTTP
TCP
UDP
Plugins
Upgrade
Dark theme
Success
0%
0
Warnings
0%
0
Errors
0%
0
Success
0%
0
Warnings
0%
0
Errors
0%
0
Success
0%
0
Warnings
0%
0
Errors
0%
0
UDP
Routers
Explore
Success
0%
0
Warnings
0%
0
Errors
0%
0
Services
Explore
Success
0%
0
Warnings
0%
0
Errors
0%
0
Features
TRACING
OFF
METRICS
OFF
ACCESSLOG
OFF
Providers
Docker

Version #7
Erstellt: 2024-03-14 19:43:46 CET von Marcel
Zuletzt aktualisiert: 2024-03-17 10:56:08 CET von Marcel