

Cockpit (lokale Anwendung) hinter Traefik erreichbar machen

Um die Anwendung "Cockpit" über Traefik (läuft als Docker-Container) erreichbar zu machen sind folgende Schritte notwendig:

Zunächst muss eine Sub-Domain eingerichtet werden und diese über einen [File-Provider](#) konfiguriert werden. Diese Datei wird über Docker-Yaml-Datei in die Traefik-Konfiguration mit eingebunden:

```
services:
  traefik:
    image: traefik:latest
    container_name: traefik
    restart: always
    command:
      # Experimentell, um das Dashboard ohne Zugriffsschutz aufzurufen
      - "--api.insecure=true"
      - "--api.dashboard=true"
      - "--log.level=warn"
      - "--providers.docker"
      # File-Provider einbinden:
      - "--providers.file.directory=/etc/traefik/"
      - "--providers.file.watch=true"
      #####
      - "--providers.docker.exposedByDefault=false"
      - "--providers.docker.network=traefik_web"
      - "--entrypoints.http.address=:80"
      - "--entrypoints.https.address=:443"
      - "--entrypoints.http.http.redirections.entrypoint.to=https"
      - "--entrypoints.http.http.redirections.entrypoint.scheme=https"
      - "--entrypoints.https.address=:443"
      # Vermeidet, dass wir den resolver in jedem container mit
      "traefik.http.routers.https.tls.certresolver=le" angeben muessen
      - "--entrypoints.https.http.tls.certResolver=le"
      - "--certificatesresolvers.le.acme.tlschallenge=true"
      # Staging-Server von Let's encrypt nutzen (für Testzwecke, größere Limits)
```

```

    #- "--certificatesresolvers.le.acme.caserver=https://acme-staging-
v02.api.letsencrypt.org/directory"
    - "--certificatesresolvers.le.acme.email=thomas@atj-krueger.de"
    - "--certificatesresolvers.le.acme.storage=/letsencrypt/acme.json"
    - "--certificatesresolvers.le.acme.tlschallenge=true"

ports:
  - "80:80"
  - "443:443"
  - "8080:8080"

volumes:
  - /var/run/docker.sock:/var/run/docker.sock:ro
  - /etc/docker/container/traefik:/letsencrypt
  # Konfigurationsdatei für File-Provider einbinden:
  # Hinweis: In diesem Fall muss die Datei auf dem Host links vom Doppelpunkt stehen.
  # Die Zielfdatei im Traefik-Container steht entsprechende rechts von Doppelpunkt.
  -
/etc/docker/container/traefik/dynamic_conf/dynamic_conf.yml:/etc/traefik/dynamic_conf.yml

networks:
  - web
extra_hosts:
  - "host.docker.internal:host-gateway"

networks:
  web:
    []external: true
    name: traefik_web

```

Inhalt der Datei `dynamic_conf.yml` (Korrekte Sub-Domain und lokale IP-dresse des Servers müssen noch angepasst werden):

```

# As YAML Configuration File
http:
  serversTransports:
    mytransport:
      insecureSkipVerify: true

```

```
routers:
  cockpit:
    service: cockpit
    rule: "Host(`cockpit.domain.de`)"

services:
  cockpit:
    loadBalancer:
      servers:
        - url: "http://ip-des-Servers:9090"
```

Darüber hinaus muss Cockpit selber auch noch etwas umkonfiguriert werden. Dazu legt man eine Datei mit Namen "[cockpit.conf](#)" im Pfad /etc/cockpit an. Deren Inhalt sieht bei mir so aus:

```
[WebService]
Origins = https://cockpit.domain.de wss://cockpit.domain.de
ProtocolHeader = X-Forwarded-Proto
AllowUnencrypted=true
```

Anschließend müssen der Traefik-Container neu erstellt und Cockpit neu gestartet werden:

```
sudo docker stop traefi
sudo docker rm traefik
sudo docker-compose -f traefik.yml up --force-recreate --build -d

sudo systemctl stop cockpit
```

Cockpit startet sich automatisch, wenn ein Aufruf erfolgt.

Mit dieser Konfiguration ist Cockpit nur noch über Traefik lauffähig. Ruft man Cockpit direkt über IP-Adresse und Port auf, kann man sich zwar anmelden, aber im weiteren Verlauf erscheint eine Fehlermeldung.

Revision #1

Created 2024-10-14 08:26:47 UTC by Thomas Krueger

Updated 2024-10-14 08:30:28 UTC by Thomas Krueger