Traefik Reverse Proxy

Verzeichnisse und Dateien anlegen

```
mkdir -p /opt/containers/traefik/data
touch /opt/containers/traefik/data/acme.json
chmod 600 /opt/containers/traefik/data/acme.json
touch /opt/containers/traefik/data/traefik.yml
```

> Wichtig: Nicht vergessen, die Datei acme.json (oder acme_letsencrypt.json) anzulegen

./docker-compose.yml erstellen

```
#
# /opt/containers/trefik/docker-compose.yml
# Reverse Proxy Traefik installieren
#
version: '3.9'
services:
  traefik:
    container_name: traefik
    image: traefik:latest
    volumes:
      # Traefik mit der Zeit vom Server syncronisieren
      - /etc/localtime:/etc/localtime:ro
      # Lesezugriff auf den UNIX Docker socket
      - /var/run/docker.sock:/var/run/docker.sock:ro
      # Traefik-Konfiguration bereitstellen
      - ./data/traefik.yml:/traefik.yml:ro
      # Datei mit Zertifikaten bereitstellen
      - ./data/acme_letsencrypt.json:/acme_letsencrypt.json
      # Datei mit dynamischer Konfiguration bereitstellen
      - ./data/dynamic_conf.yml:/dynamic_conf.yml
    labels:
      # Traefik durch Watchtower aktualisieren lassen
```

```
      # https://goneuland.de/docker-images-automatisiert-aktualisieren-mit-watchtower/
      - "com.centurylinklabs.watchtower.enable=true"
      # Traefik für Traefik aktivieren
      - "traefik.enable=true"
      # Als Einstiegspunkt wählen wir direkt HTTPS
      - "traefik.http.routers.traefik.entrypoints=https"
      # Domain anpassen unter der Traefik erreichbar sein soll.
      - "traefik.http.routers.traefik.rule=Host(`traefik.die-steffens.eu`)"
      # Middlewares definieren, welche verwendet werden sollen.
      # Hier die Authentifizierung via htpasswd + alle die unter default definiert sind.
      - "traefik.http.routers.traefik.middlewares=traefikAuth@file,default@file"
      # SSL Zertifikat abrufen
      - "traefik.http.routers.traefik.tls=true"
      # Definierten Zertifikat Resolver aus traefik.yml wählen
      - "traefik.http.routers.traefik.tls.certresolver=http"
      - "traefik.http.routers.traefik.service=api@internal"
      - "traefik.http.services.traefik.loadbalancer.server.port=80"
      - "traefik.http.services.traefik.loadbalancer.sticky.cookie.httpOnly=true"
      - "traefik.http.services.traefik.loadbalancer.sticky.cookie.secure=true"
     # Traefik dem Proxy-Netzwerk hinzufügen.
      - "traefik.docker.network=proxy"
    restart: unless-stopped
    security_opt:
      - no-new-privileges:true
    networks:
      proxy:
    hostname: traefik
    ports:
      # Ports definieren, welche durch Traefik gemanaget werden.
      - "80:80"
      - "443:443"

networks:
  proxy:
    name: proxy
    driver: bridge
    attachable: true
```

./data/dynamic_conf.yml füllen

```
#
# /opt/containers/trafik/data/dynamic_conf.yml
#
# TLS
# Hier werden alle notwendigen Einstellungen für das Zertifikat getroffen.
# In Kombination mit den Einstellungen unter http.middlewares.default-security-headers
bekommen  wir ein A+ Zertifikat.
tls:
  options:
    default:
      minVersion: VersionTLS12
      cipherSuites:
        - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
        - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
        - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305
        - TLS_AES_128_GCM_SHA256
        - TLS_AES_256_GCM_SHA384
        - TLS_CHACHA20_POLY1305_SHA256
      curvePreferences:
        - CurveP521
        - CurveP384
      sniStrict: true

# Middlewares
# Optionale Optimierungen, die bei jeder Anfrage vorgenommen werden sollen bevor diese an den
Zielcontainer geleitet wird.
http:
  middlewares:
    # Eine grundlegende Authentifizierungs-Middleware, um das Traefik-Dashboard via htpasswd
zu schützen
    # Um die Authentifizierung für einen Container zu nutzen können wir
"traefik.http.routers.definierteRoute.middlewares=traefikAuth@file" nutzen
    traefikAuth:
      basicAuth:
        users:
          - "Kamikaze-Jim:$apr1$P/tG13aC$BdXjI6AtKysGWvRAr9aoI0"

    # Empfohlene Standard-Middleware für die meisten Dienste
    # Hinzufügbar via "traefik.http.routers.definierteRoute.middlewares=default@file"
    # Equivalent mit "traefik.http.routers.definierteRoute.middlewares=default-security-
```

```
headers@file,gzip@file"
    # Die Liste kann hier auch beliebig erweitert werden
    default:
      chain:
        middlewares:
          - default-security-headers
          - gzip

    # Kompatibilität zu alten Anleitungen. Damit kann auch wieder
"traefik.http.routers.definierteRoute.middlewares=secHeader@file"
    secHeaders:
      chain:
        middlewares:
          - default-security-headers
          - gzip

    # Standard Header
    default-security-headers:
      headers:
        browserXssFilter: true
        contentTypeNosniff: true
        forceSTSHeader: true
        frameDeny: true
#       Deprecated
#       sslRedirect: true
        #HSTS Configuration
        stsIncludeSubdomains: true
        stsPreload: true
        stsSeconds: 31536000
        customFrameOptionsValue: "SAMEORIGIN"
    # Gzip Kompression
    gzip:
      compress: {}
```

User und Passwort festlegen

```
#
# Erzeugt das für obige Datei erforderliche USR/PWD-Gespann
#
echo $(htpasswd -nb <user> '<password>')
```

```
#
# Zum Beispiel:
# echo $(htpasswd -nb jonathan 'supersicher')
# Ausgabe: jonathan:$$apr1$$J8vBlNIm$$lSwLv9iGa8KcCct3EyLD41
# Sollte es an dieser Stelle zu Problemen kommen kann man sich auch das Passwort auf Webseiten
# wie zum Beispiel: https://www.redim.de/blog/passwortschutz-mit-htaccess-einrichten
generieren lassen.
```

./data/traefik.yml füllen

```
#
# /opt/containers/traefik/data/traefik.yml
#
# Statische Traefik-Konfigurationsdatei
# https://doc.traefik.io/traefik/getting-started/configuration-overview/#the-static-
configuration
# https://doc.traefik.io/traefik/reference/static-configuration/cli/

api:
  dashboard: true                           # Aktivieren des Dashboard

# Certificate Resolver
# Diese sind für den Abruf von Zertifikaten von einem ACME-Server zuständig
# https://doc.traefik.io/traefik/https/acme/#certificate-resolvers
certificatesResolvers:
  http:
    acme:
      email: "ralf@rastef.de"               # E-Mail-Adresse für die Registrierung
      storage: "acme_letsencrypt.json"      # Datei für die Speicherung von Zertifikate (Ich
weiche hier bewusst von dem "Standard": acme.json ab)
      httpChallenge:
        entryPoint: http

# EntryPoints
# EntryPoints sind die Netzwerk-Eingangspunkte in Traefik. Sie definieren den Port, der die
Pakete empfängt.
# https://doc.traefik.io/traefik/routing/entrypoints/
entryPoints:
  http:
    address: ":80"                          # Erstellen des Einstiegspunkt für HTTP (Port
```

```
80)
    http:
      redirections:                          # Weiterleitung von HTTP auf HTTPS (Port 80 zu
Port 443).
        entryPoint:
          to: "https"                        # Das Ziel
          scheme: "https"                    # Umleitungszielschema
  https:
    address: ":443"                          # Erstellen des Einstiegspunkt für HTTPS (Port
443)



global:
  checknewversion: true                      # In regelmäßigen Abständen prüfen, ob eine neue
Version veröffentlicht wurde.
  sendanonymoususage: false                  # Regelmäßige Übermittlung anonymer
Nutzungsstatistiken.

providers:
  docker:
    endpoint: "unix:///var/run/docker.sock"  # Den UNIX Docker socket beobachten
    exposedByDefault: false                  # Nur Container ausstellen, die explizit
aktiviert sind (mit dem Label traefik.enabled)
    network: "proxy"                         # Standardnetzwerk, das für Verbindungen zu
allen Containern verwendet wird.
  file:
    filename: "./dynamic_conf.yml"           # Link zur dynamischen Konfiguration
    watch: true                              # Achten auf Änderungen
  providersThrottleDuration: 10              # Frequenz in welchen Abständen die
Konfiguration nachgeladen wird
```

Die eigentliche Anleitung stammt von hier: https://goneuland.de/traefik-v2-reverse-proxy-mit-crowdsec-einrichten/

Revision #1
Created 2026-09-26 13:20:52 UTC by ralf
Updated 2026-09-26 13:20:52 UTC by ralf