Traefik

Requerimientos previos

- Tener instalado Docker

Pasos

1. Crear una red en Docker para Traefik
2. Crear una contraseña para Tablero de control de Traefik □□
3. Crear archivos de configuración:
   1. Archivo de Docker Compose: `compose.yaml`
   2. Configuración principal de Traefik: `traefik.yaml`
   3. Configuración dinámica (para middlewares y otras configuraciones): `dynamic.yaml`
4. Levantar contenedor
5. Acceder al tablero de control

Configuración previa

Crear una red para Traefik

```
sudo docker network create nombre_de_la_red
```

Archivos de configuración

.env

```
MAIN_DOMAIN=ejemplo.com
VOLUME_PATH=/var/www/dashboard.ejemplo.com
```

compose.yaml

```
services:
  traefik:
    image: traefik
    container_name: traefik
    ports:
      - "80:80"
      - "443:443"
      - "8082:8082"
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock:ro"
      - ./traefik.yaml:/etc/traefik/traefik.yaml:ro
      - ./dynamic.yaml:/etc/traefik/dynamic.yaml:ro
      - traefik-certs:/certs
    networks:
      - traefik
    restart: always
    labels:
      - "traefik.enable=true"
      - "traefik.http.middlewares.traefik-auth.basicauth.removeheader=true"
      - "traefik.http.middlewares.traefik-auth.basicauth.users=user:password"
      - "traefik.http.routers.traefik.rule=Host(`dashboard.${MAIN_DOMAIN}`)"
      - "traefik.http.routers.traefik.service=api@internal"
      - "traefik.http.routers.traefik.tls=true"
      - "traefik.http.routers.traefik.tls.certresolver=tls"
      - "traefik.http.routers.traefik.entrypoints=https"
      - "traefik.http.routers.traefik.middlewares=traefik-
auth,secHeaders@file,autodetectContenttype@file"
    healthcheck:
      test: ["CMD", "traefik", "healthcheck", "--ping"]
      interval: 60s
      timeout: 5s
      retries: 3
      start_period: 5s
volumes:
  traefik-certs:
    name: traefik-certs
networks:
  traefik:
    external: true
```

traekif.yaml

```
providers:
  docker:
    exposedByDefault: false
    endpoint: "unix:///var/run/docker.sock"
    network: traefik
  file:
    filename: /etc/traefik/dynamic.yaml
    watch: true
api:
  dashboard: true
entryPoints:
  http:
    address: :80
  https:
    address: :443
  metrics:
    address: :8082
    transport: # Opcional
      respondingTimeouts:
        readTimeout: 60m #  Amplia el tiempo antes de desconectar una conexión. Permite la
subida de archivos grandes al servidor.
ping:
  entryPoint: metrics
accessLog:
  filePath: ./logs/access.log
  bufferingSize: 50
certificatesResolvers:
  tls:
    acme:
      email: correo@electronico.com
      storage: /certs/acme.json
      tlsChallenge: {}
      httpChallenge:
        entryPoint: http
```

dynamic.yaml

```
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
http:
  middlewares:
    # agregar middlewares aquí
```

Middlewares de ejemplo

Los middlewares se pueden configurar el segundo archivo de configuración, dyamic.yaml, que permite actualizar la configuración en tiempo de ejecución, sin reiniciar el contenedor.

Más información en https://doc.traefik.io/traefik/v2.5/middlewares/http/headers/

Encabezados seguros (secHeaders)

```
http:
  middlewares:
    secHeaders:
      headers:
        browserXssFilter: true
        contentTypeNosniff: true
        frameDeny: true
        stsIncludeSubdomains: true
        stsPreload: true
        stsSeconds: 31536000
        customFrameOptionsValue: "SAMEORIGIN"
        customResponseHeaders:
```

```
          server: ""
          x-powered-by: ""
```

Política de permisos (permissionsPolicy)

```
http:
  middlewares:
    permissionsPolicy:
      headers:
        permissionsPolicy: "autoplay=(self), fullscreen=(self), picture-in-picture=(self),
screen-wake-lock=(self), sync-xhr=(self), web-share=(self)"
```

Política de referidos (referrerPolicy)

```
http:
  middlewares:
    referrerPolicy:
      headers:
        referrerPolicy: "no-referrer-when-downgrade"
```

Autodetección de content-type (autoDetectContentType)

```
http:
  middlewares:
    autoDetectContentType:
      contentType: {}
```

Revisión #10
Creado 2025-07-31 00:55:30 UTC por IM
Actualizado 2026-03-07 17:24:19 UTC por IM