

Traefik v3

Overview

Traefik is the single ingress point for all named HTTP/HTTPS traffic on Centerpoint. It runs on the `traefik-net` Docker bridge network and discovers routes automatically from container labels — no manual reload is needed when stacks are added or removed. All TLS termination happens at Traefik; individual application containers serve plain HTTP internally.

Access

Type	URL / Endpoint	Notes
Dashboard	<code>https://traefik.home.local</code>	Internal only — LAN access required

The dashboard is exposed only on the internal Step-CA route. There is no external (internet-facing) route for the Traefik dashboard.

Configuration

Image: `traefik:v3.6.13` **Compose project:** `traefik` (same project as `step-ca`)

Ports

Port	Protocol	Purpose
<code>80</code>	TCP	HTTP — auto-redirects all traffic to HTTPS
<code>443</code>	TCP	HTTPS — primary endpoint
<code>8080</code>	TCP	Traefik API / Dashboard (internal)

Static Configuration — `traefik.yml`

Location: `/home/jeeves/docker/traefik/traefik.yml`

Key settings:

```
api:
  dashboard: true
  insecure: false

entryPoints:
  web:
    address: ":80"
  http:
    redirections:
      entryPoint:
        to: websecure
        scheme: https
  websecure:
    address: ":443"

providers:
  docker:
    network: traefik-net
    exposedByDefault: false
  file:
    directory: /config
    watch: true

certificatesResolvers:
  letsencrypt:
    acme:
      email: <redacted>
      storage: /letsencrypt/acme.json
      httpChallenge:
        entryPoint: web
  step-ca:
    acme:
      email: <redacted>
      storage: /letsencrypt/step-ca.json
      caServer: https://ca.home.local:9000/acme/acme/directory
      certificatesDuration: 720
```

```
experimental:
  plugins:
    geoblock:
      moduleName: github.com/PascalMinder/geoblock
      version: v0.3.6
    crowdsec-bouncer-traefik-plugin:
      moduleName: github.com/maxlerebourg/crowdsec-bouncer-traefik-plugin
      version: v1.3.0
```

Certificate Resolvers

Resolver	Scope	Method	CA	Duration
letsencrypt	External (<code>*.jeeves5454.ddns.net</code>)	HTTP challenge	Let's Encrypt	90 days
step-ca	Internal (<code>*.home.local</code>)	ACME	Internal Step-CA at <code>ca.home.local:9000</code>	30 days

Plugins

Plugin	Version	Purpose
PascalMinder/geoblock	v0.3.6	Blocks requests from countries not in allowlist
maxlerebourg/crowdsec-bouncer-traefik-plugin	v1.3.0	Forwards requests to CrowdSec bouncer for IP checks

Geoblock is configured to allow **CA**, **US**, and **IN** only. The middleware is named `plex-geoblock@file` and is defined in `/home/jeeves/docker/traefik/config/crowdsec.yml`.

Dynamic Configuration

Location: `/home/jeeves/docker/traefik/config/` (file-watched)

File	Contents
<code>crowdsec.yml</code>	<code>plex-geoblock</code> and <code>crowdsec-bouncer</code> middleware defs
<code>dynamic.yml</code>	Static file-provider routes for Plex (no Docker labels)
<code>middlewares.yml</code>	Additional shared middleware definitions

Standard Dual-Route Label Pattern

Each service that requires both internal and external access uses this label pattern in its `docker-compose.yml`:

```
labels:
  - "traefik.enable=true"

  # External route – Let's Encrypt TLS + security middleware
  - "traefik.http.routers.<name>-ext.rule=Host(<svc>.jeeves5454.ddns.net)"
  - "traefik.http.routers.<name>-ext.entrypoints=websecure"
  - "traefik.http.routers.<name>-ext.tls.certresolver=letsencrypt"
  - "traefik.http.routers.<name>-ext.middlewares=authentik-auth@docker,plex-geoblock@file,crowdsec-bouncer@file"

  # Internal route – Step-CA TLS, no SSO middleware
  - "traefik.http.routers.<name>-int.rule=Host(<svc>.home.local)"
  - "traefik.http.routers.<name>-int.entrypoints=websecure"
  - "traefik.http.routers.<name>-int.tls.certresolver=step-ca"

  # Backend
  - "traefik.http.services.<name>-svc.loadbalancer.server.port=<port>"
```

Volumes / Bind Mounts

Host Path	Container Path	Purpose
/home/jeeves/docker/traefik/traefik.yml	/traefik.yml	Static config
/home/jeeves/docker/traefik/config	/config	Dynamic config (file-watched)
/home/jeeves/docker/traefik/letsencrypt	/letsencrypt	ACME certificate storage
/var/log/traefik	/var/log/traefik	Access logs (JSON)
/var/run/docker.sock	/var/run/docker.sock	Docker label discovery

Dependencies

- `traefik-net` Docker network (must be created manually before first start)
- `step-ca` container (must be healthy for internal certs to renew)
- CrowdSec LAPI at `crowdsec:8080` (bouncer requires it; Traefik starts without it but will error on blocked requests)

Notes / Gotchas

- `traefik-net` is externally managed — always create it first:

```
docker network create traefik-net
```

- Traefik discovers routes from container labels only on containers attached to `traefik-net`. If a container is not on this network it will be invisible to Traefik even with `traefik.enable=true`.
- `exposedByDefault: false` means every service must explicitly opt in with `traefik.enable=true`.
- The `step-ca` certificate resolver stores its ACME account and certs in `/letsencrypt/step-ca.json` (separate from the Let's Encrypt storage). Do not delete this file or certificates will be re-issued from scratch.
- Access logs are written to `/var/log/traefik/access.log` in JSON format and are also bind-mounted into the CrowdSec container for log ingestion.

Last Updated: 2026-06-16

Revision #1

Created 2026-06-17 00:21:49 UTC by Admin

Updated 2026-06-17 00:22:45 UTC by Admin