Traefik Edge router and proxy

CONCEPTS

EntryPoint port :80 / :443
↓ rule matches
Router rule + priority
↓ runs
Middlewares auth, redirect, headers
↓ forwards
Service load balancer
↓ to
Backend container / server

> Static config (entrypoints, providers, resolvers) is read at start; dynamic config (routers, services, middlewares) reloads live.

RUN AND STATIC CONFIG

`traefik --configFile=<file>`
Start with a config file

`traefik --entryPoints.<name>.address=<addr>`
Define an entry point, e.g. :80

`traefik --providers.docker=true`
Enable the Docker provider

`traefik --providers.file.directory=<dir>`
Load dynamic config from a folder

`traefik --api.dashboard=true`
Serve the dashboard

`traefik --providers.docker.exposedByDefault=false`
Expose only labelled containers

DOCKER LABELS

`traefik.enable=true`
Include this container

`traefik.http.routers.<r>.rule=Host(`<host>`)`
Route a hostname to it

`traefik.http.routers.<r>.entrypoints=<ep>`
Listen on this entry point

DOCKER LABELS (cont.)

`traefik.http.services.<s>.loadbalancer.server.port=<port>`
Backend port inside the container

`traefik.http.routers.<r>.tls=true`
Serve this router over TLS

`traefik.http.routers.<r>.tls.certresolver=<res>`
Get a certificate from a resolver

`traefik.http.routers.<r>.middlewares=<m>`
Attach middlewares, comma separated

ROUTER RULES

`Host(`<domain>`)`
Match a hostname

`PathPrefix(`/<prefix>`)`
Match a path prefix

`Host(`a`) && PathPrefix(`/api`)`
Combine with AND

MIDDLEWARES

`redirectscheme.scheme=https`
Redirect HTTP to HTTPS

TLS AND ACME

`--certificatesresolvers.<res>.acme.email=<addr>`
Account email

`--certificatesresolvers.<res>.acme.storage=<file>`
Where certs are stored

`--certificatesresolvers.<res>.acme.httpchallenge.entrypoint=web`
Use the HTTP challenge

FILE PROVIDER

`http.routers.<r>.rule`
Router in dynamic YAML

`http.services.<s>.loadBalancer.servers[].url`
Backend address in a service