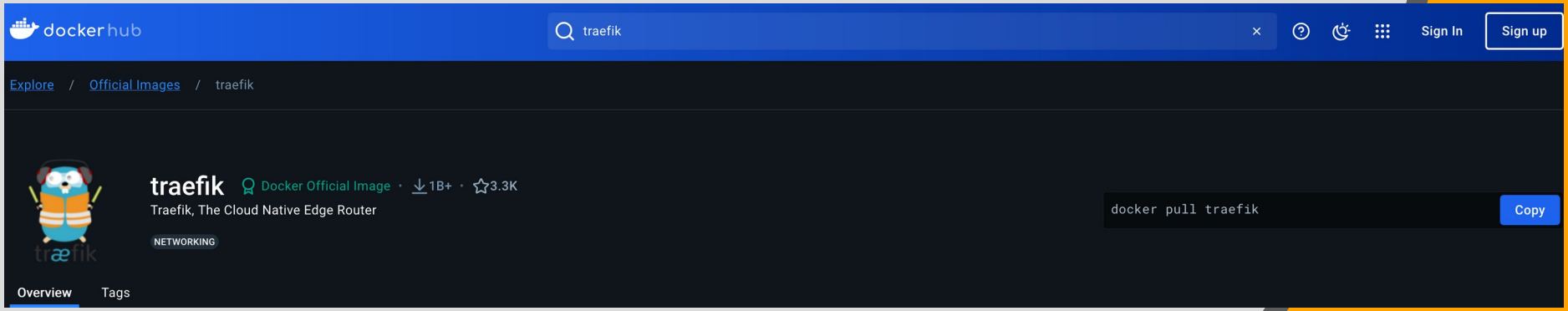
Linux Day Torino 2024

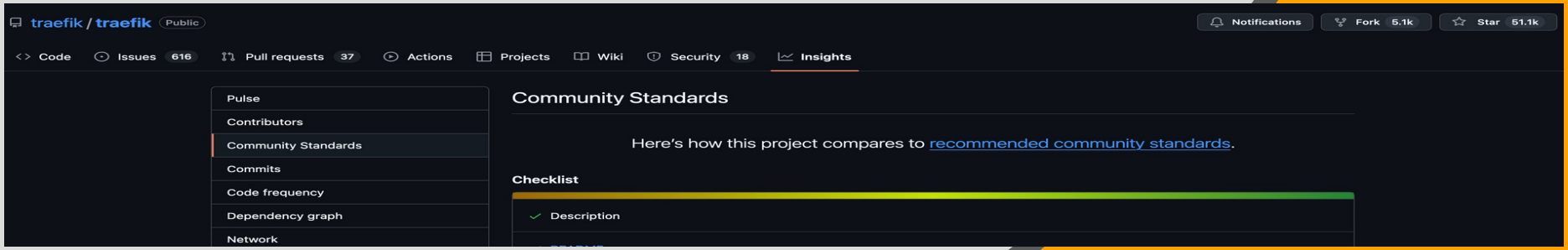
træfik

il reverse proxy
cloud native

@oirasor
linux.it
ils.org

Overview


docker hub
traefik
Sign In
Sign up
Explore / Official Images / traefik
traefik
Docker Official Image · 1B+ · 3.3K
Traefik, The Cloud Native Edge Router
NETWORKING
docker pull traefik
Copy
Overview
Tags


traefik / traefik Public
Notifications
Fork 5.1k
Star 51.1k
Code
Issues 616
Pull requests 37
Actions
Projects
Wiki
Security 18
Insights
Pulse
Contributors
Community Standards
Commits
Code frequency
Dependency graph
Network
Community Standards
Here's how this project compares to recommended community standards.
Checklist
Description

Stats

- 1 B+ Docker Hub downloads
- 50k+ Stars on GitHub
- 500+ Contributors

Use cases

- Reverse Proxy
- API Gateway
- Load Balancing
- Certificate Management
- Kubernetes Ingress (or IngressRoute?)

History

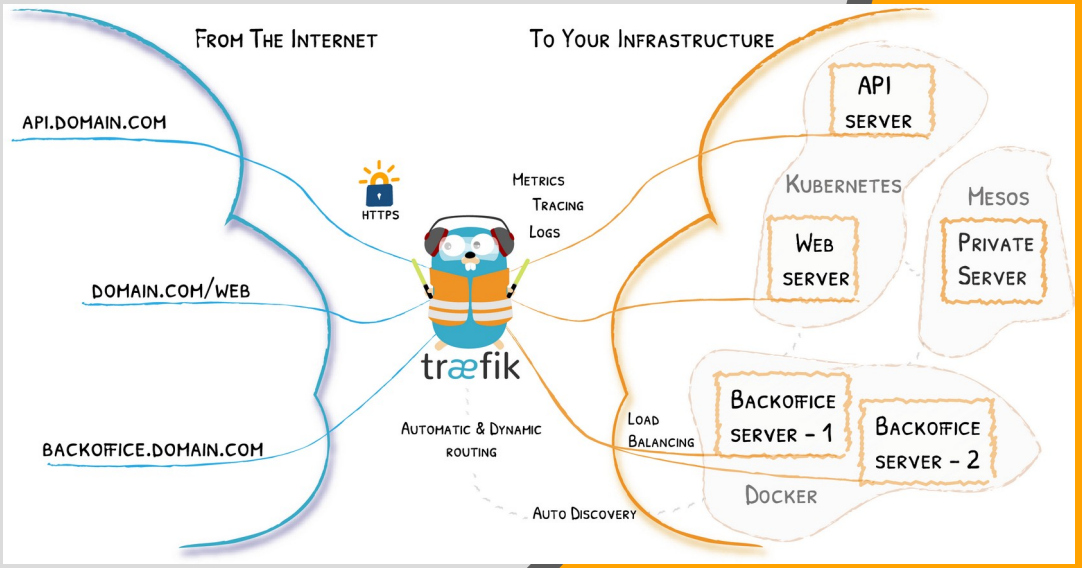
- Project born in 2015
- Founder: Emile Vauge
- French project (as a lot of open source EMEA projects)
  <=Tell me why!

Why?

- Tons of reverse Proxies but none were dynamic
- Microservices are dynamic which requires dynamic configurations
- Watches the Orchestrator for new events (provider integration is the way!)

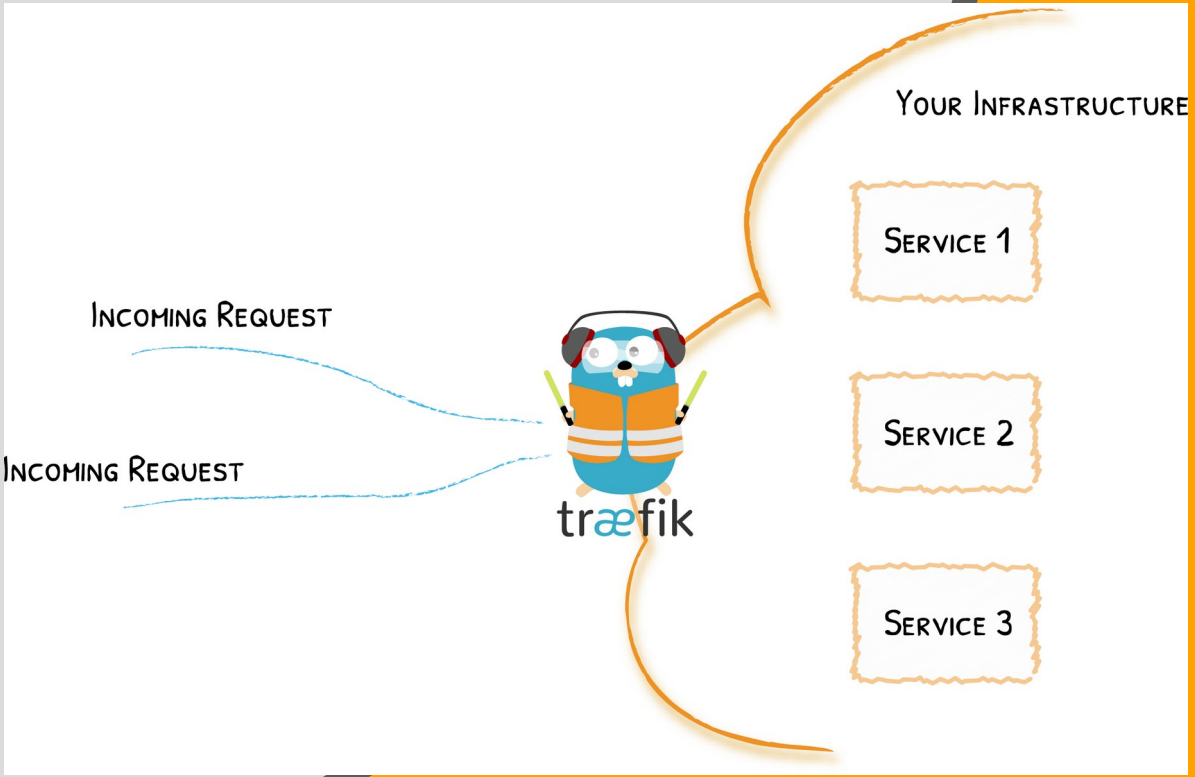
Basic overview

- Written in Go
- Edge Router
- Automatically discovers service configurations
- Native integration with AWS, Kubernetes, Docker, Mesos, Nomad and many more
- Traefik automatically synchronizes configuration changes in real-time, no restarts,
no down-time



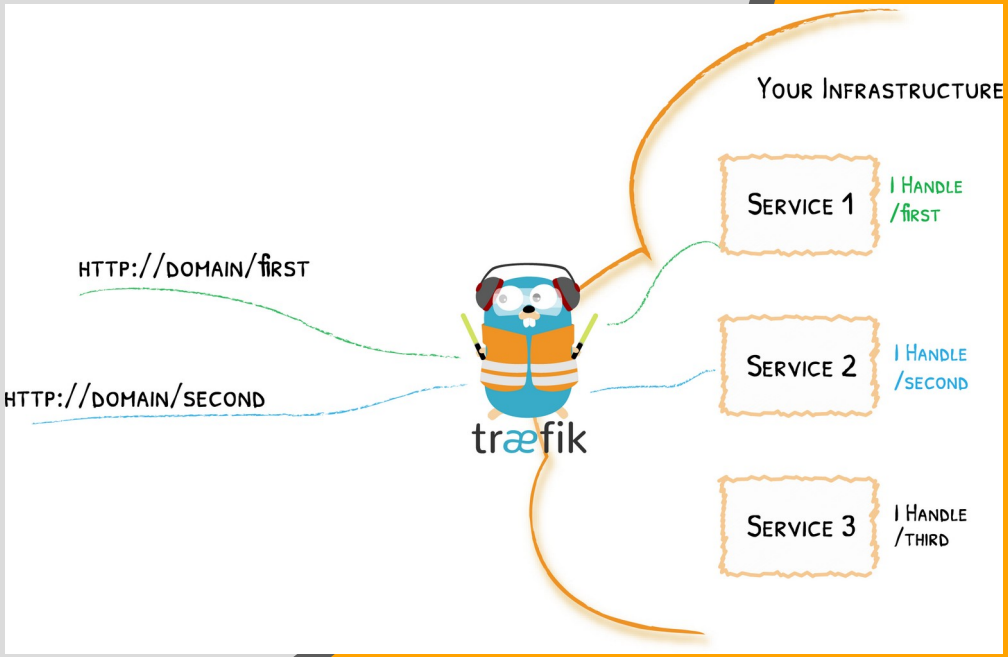
Edge router

- Front door to your platform
- Intercepts incoming requests and routes the request
- Defined logic to determine which service recieves the request based on Path, host, headers, and more

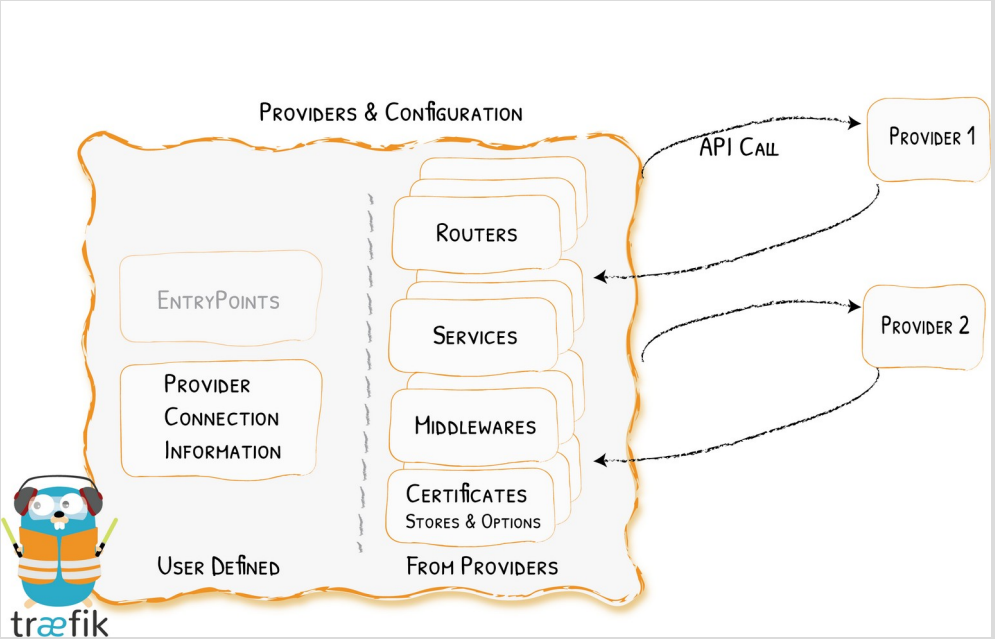


Auto service discovery

- Traditional edge /routers reverse proxies need a configuration file to get every possible route to the service(s)
- Traefik get all the information directly to the service
- This means less configurations after Traefik initialization and always up to date rules



Supported providers

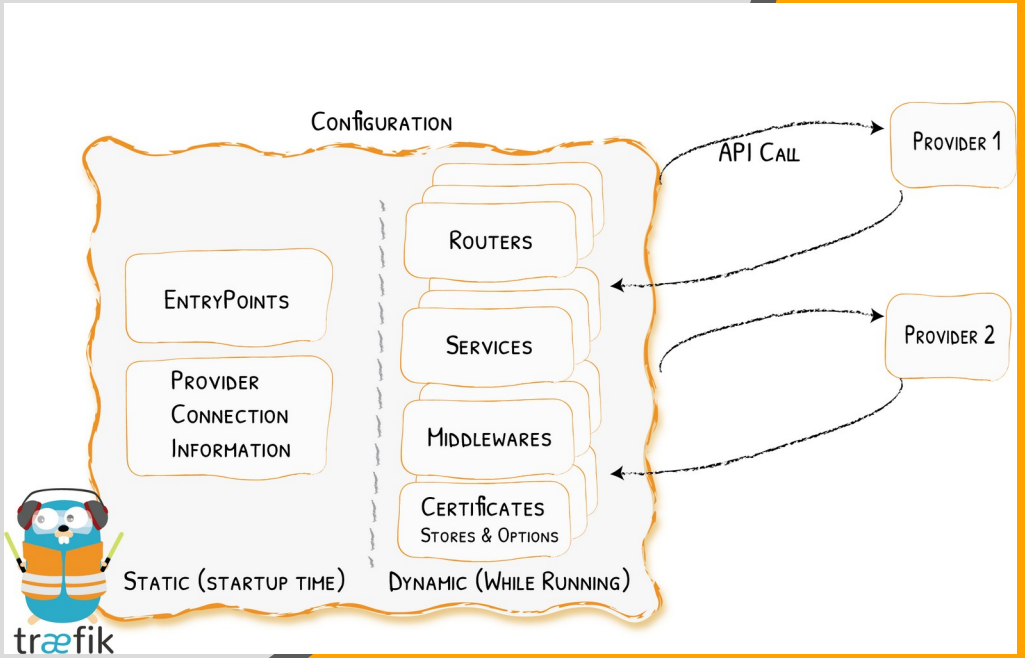


Provider	Type	Configuration Type	Provider Name
Docker	Orchestrator	Label	`docker`
Kubernetes IngressRoute	Orchestrator	Custom Resource	`kubernetescrd`
Kubernetes Ingress	Orchestrator	Ingress	`kubernetes`
Kubernetes Gateway API	Orchestrator	Gateway API Resource	`kubernetesgateway`
Consul Catalog	Orchestrator	Label	`consulcatalog`
Nomad	Orchestrator	Label	`nomad`
ECS	Orchestrator	Label	`ecs`
File	Manual	YAML/TOML format	`file`
Consul	KV	KV	`consul`
Etcd	KV	KV	`etcd`
ZooKeeper	KV	KV	`zookeeper`
Redis	KV	KV	`redis`
HTTP	Manual	JSON format	`http`

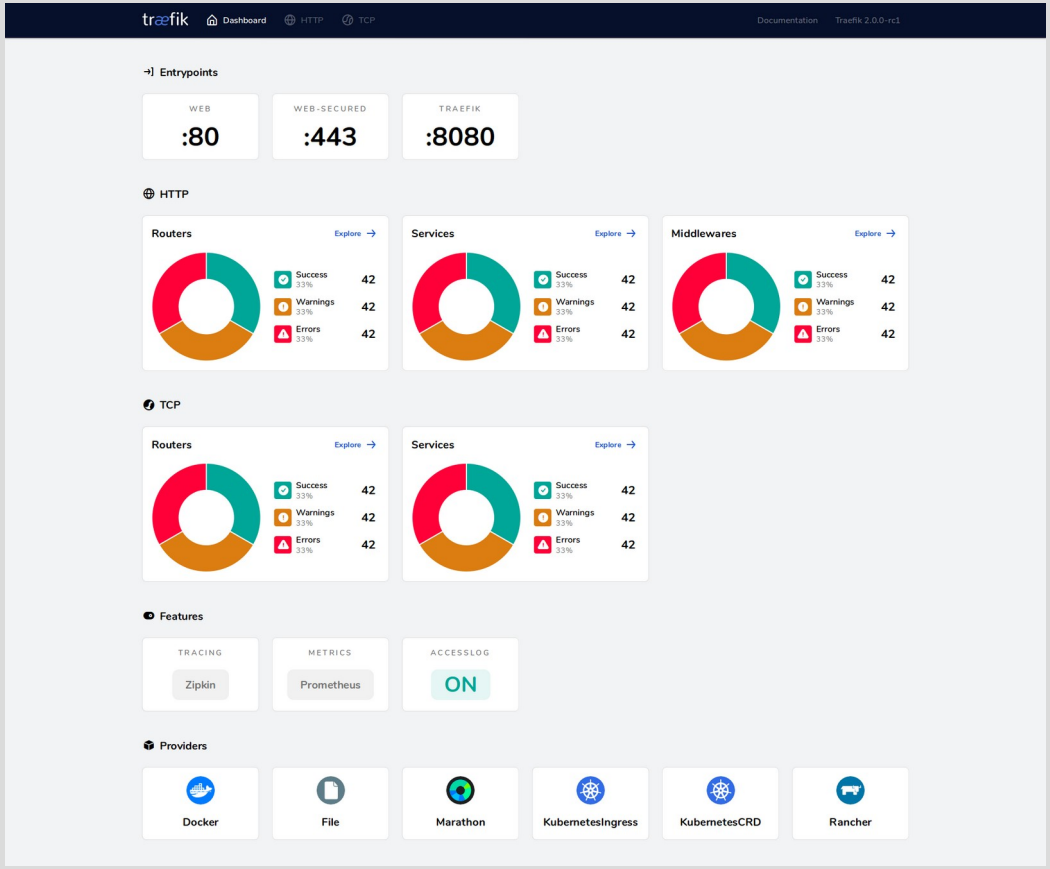
Configuration introduction

Configuration in Traefik can refer to two different things:

- The fully dynamic routing configuration (referred to as the dynamic configuration)
- The startup configuration (referred to as the static configuration)



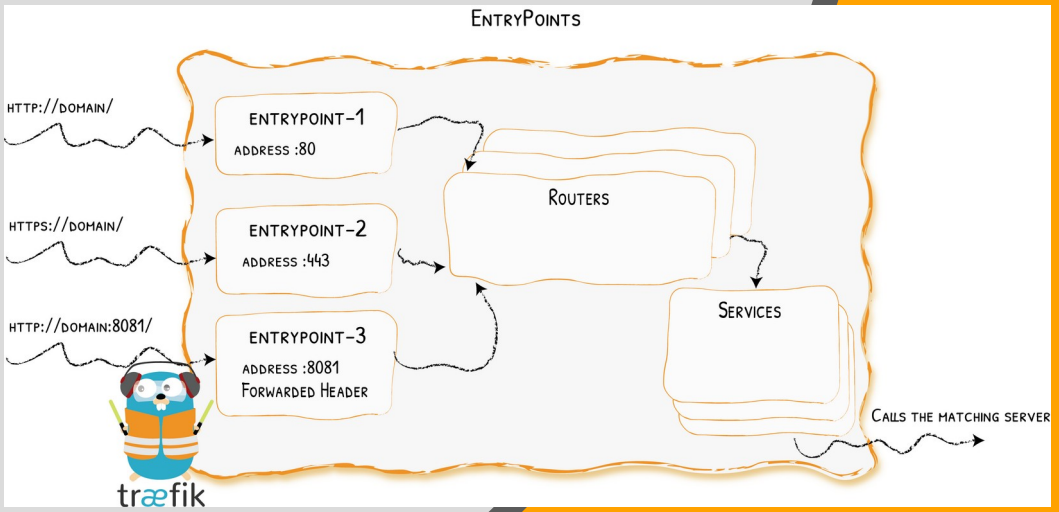
Dashboard


traefik
Dashboard
HTTP
TCP
Documentation
Traefik 2.0.0-rc1
Entrypoints
WEB
:80
WEB-SECURED
:443
TRAEFIK
:8080
HTTP
Routers
Explore
Success 33% 42
Warnings 33% 42
Errors 33% 42
Services
Explore
Success 33% 42
Warnings 33% 42
Errors 33% 42
Middlewares
Explore
Success 33% 42
Warnings 33% 42
Errors 33% 42
TCP
Routers
Explore
Success 33% 42
Warnings 33% 42
Errors 33% 42
Services
Explore
Success 33% 42
Warnings 33% 42
Errors 33% 42
Features
TRACING
Zipkin
METRICS
Prometheus
ACCESSLOG
ON
Providers
Docker
File
Marathon
KubernetesIngress
KubernetesCRD
Rancher

EntryPoints

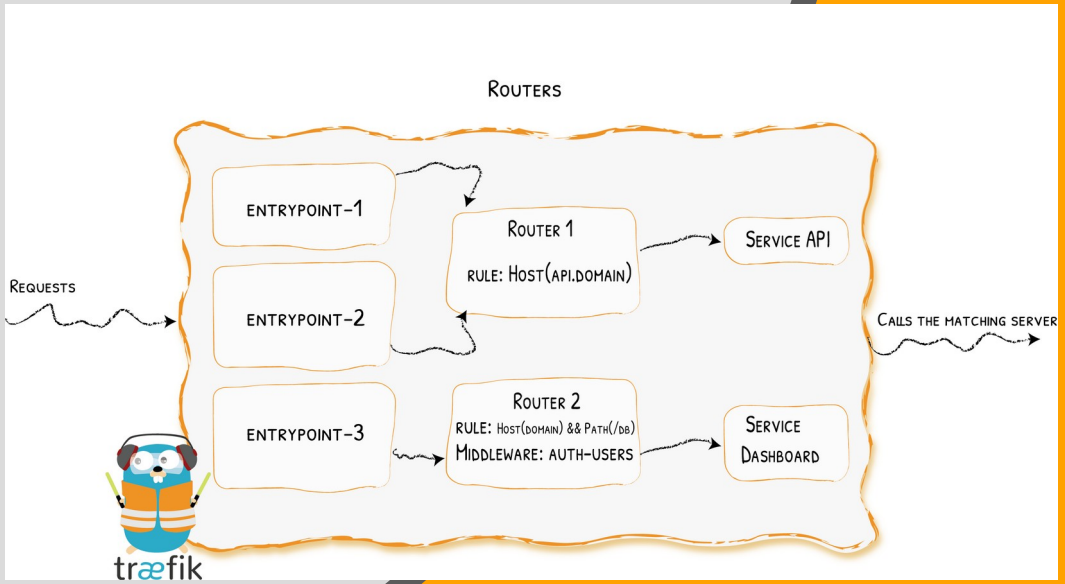
EntryPoints are the network entry points into Traefik.

They define the port which will receive the packets, and whether to listen for TCP or UDP.



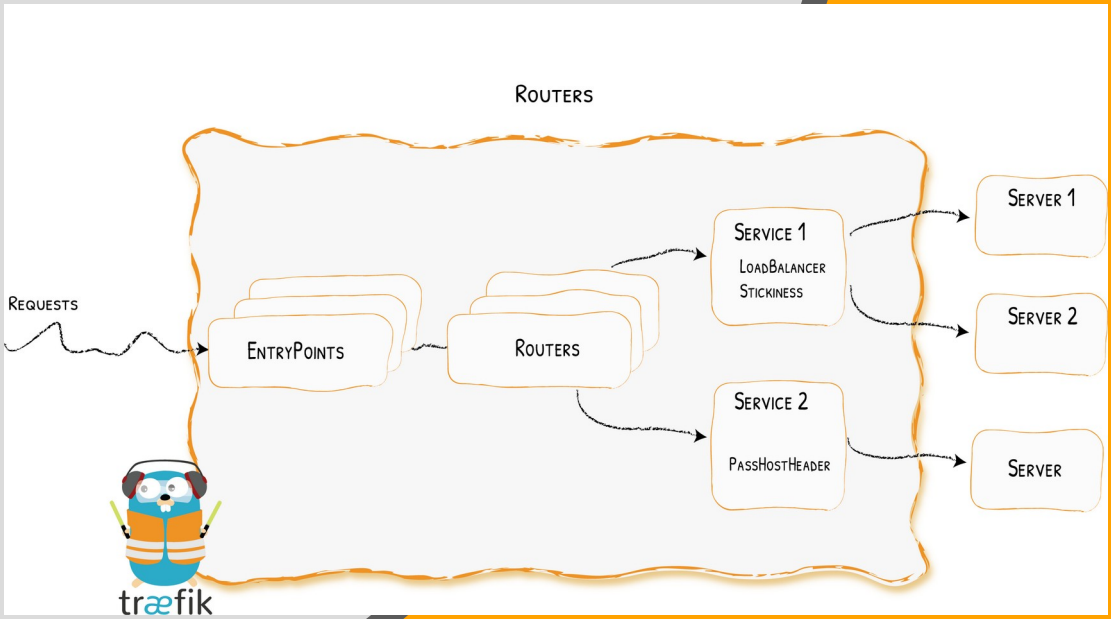
Routers

A router is in charge of connecting incoming requests to the services that can handle them. In the process, routers may use pieces of middleware to update the request, or act before forwarding the request to the service.



Services

The Services are responsible for configuring how to reach the actual services that will eventually handle the incoming requests.



Providers

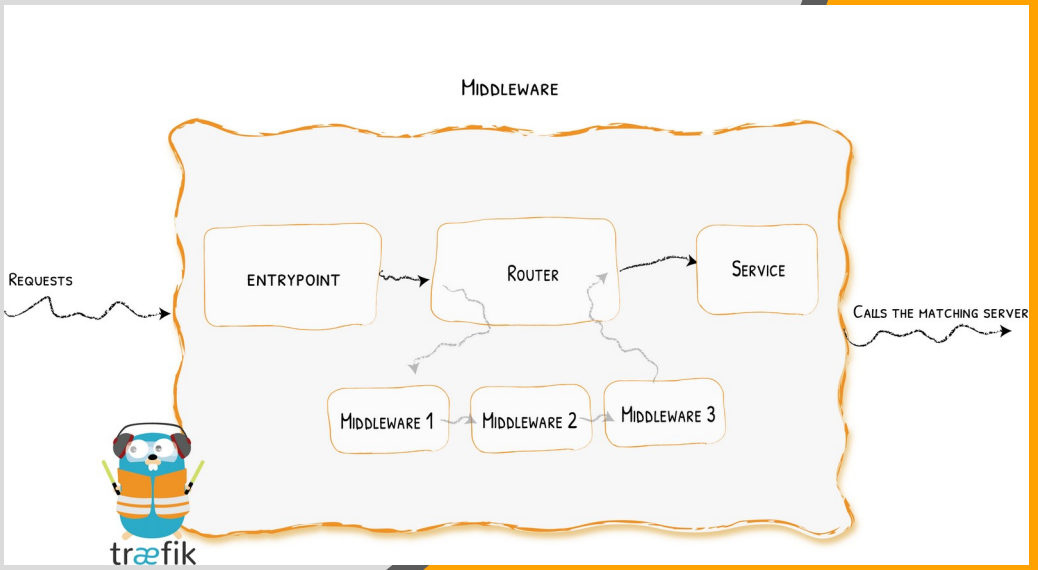
- A Provider links an infrastructure component (Orchestrator, Container Engine, Key-Value store, or File to Traefik)
- Configure Traefik to connect to the Provider
- Traefik detects configuration changes and events from the Provider

Middlewares

Attached to the routers, pieces of middleware are a means of tweaking the requests before they are sent to your service (or before the answer from the services are sent to the clients).

There are several available middleware in Traefik, some can modify the request, the headers, some are in charge of redirections, some add authentication, and so on.

Middlewares that use the same protocol can be combined into chains to fit every scenario.



Metrics

Traefik provides metrics in the OpenTelemetry format as well as the following vendor specific backends:

- Datadog
- InfluxDB2
- Prometheus
- StatsD

Traefik Proxy hosts an official Grafana dashboard for both on-premises and Kubernetes deployments.

Metric	Type	Labels	Description
Config reload total	Count		The total count of configuration reloads.
Config reload last success	Gauge		The timestamp of the last configuration reload success.
Open connections	Gauge	`entrypoint`, `protocol`	The current count of open connections, by entrypoint and protocol.
TLS certificates not after	Gauge		The expiration date of certificates.

Tracing

OpenTelemetry is a collection of APIs, SDKs, and tools.

Use it to instrument, generate, collect, and export telemetry data (metrics, logs, and traces) to help you analyze your software’s performance and behavior.

Advanced techniques

- Not covered with deep dive demo
- Maybe I will start a dedicated course on video.linux.it (oirasor_channel@video.linux.it)
- If interested feel free to contact: oirasor@linux.it

Advanced techniques

- Load Balancing (WRR)
- Progressive Delivery (WRR)
- Mirroring with Traefik
- Sticky Sessions
- Nested health checks

Copyleft