陳宗霆

Goal

- Why we need traefik?
- What difference between nginx & traefik?
- Where are properties & configs?
- How to use traefik?

Outline

- Traefik 定位
- 反向代理
- OSI
- Traefik Introduction & Practice
  - Configuration

- Difference between V1 & V2.0 & V2.1

Traefik 定位

一個**適合微服務**的**反向代理工具**

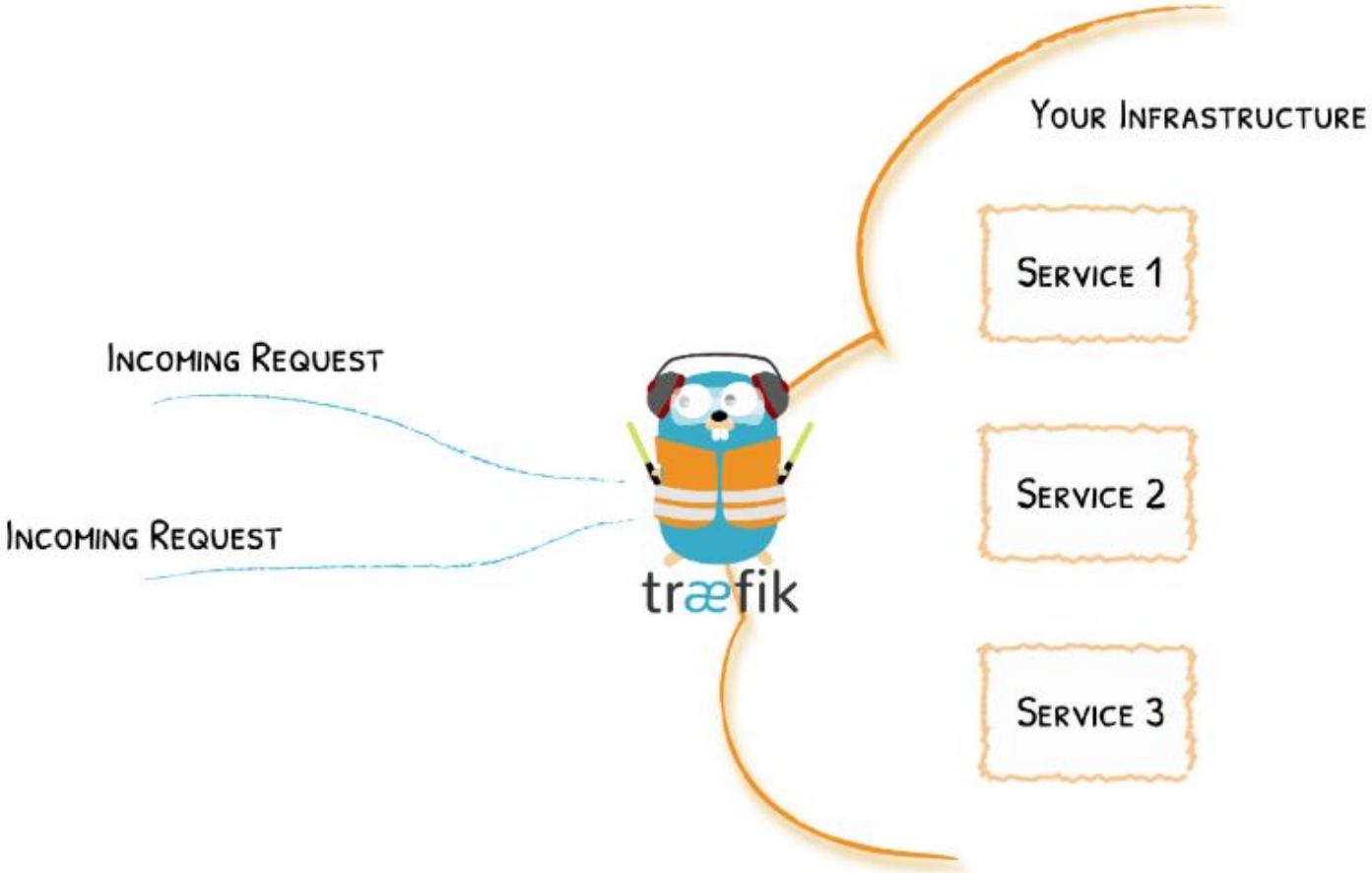
反向代理 (Hidden Server)

安全

- 安全隔離
- 存取認證、訪問控制
- 抵抗突發流量

效率

- 負載平衡
- 靜態檔案
- 封包緩存



常見的反向代理工具

- Apache Http Server
- HAProxy
- Nginx

考量點

- Throughput
- Availability (Loss)
- Maximum concurrent size

微服務化造成的差異

- 服務數量非常多
  手動註冊(X)
- 服務可能會動態增減
  註冊後須重啟才生效 (not good)
- 網路設定更麻煩

- 需要自動監測 docker & kubernetes 變化的機制

Project: Nginx-proxy、Nginx Ingress

OSI

分層	名稱	簡介	相關網路協議
7	應用層(Application)	服務接口	HTTP、FTP、Telnet、SMTP
6	表現層(Presentation)	數據格式化、數據加密	IPX、XDP
5	交談層(Session)	建立、終止對話	SSL、LDAP、RPC
4	傳輸層(Transport)	端對端連結	TCP、UDP
3	網路層(Network)	尋址、路由	IP、ICMP、VRRP
2	資料連結層(Data-link)	封包傳輸	MPLS、XTP
1	實體層(Physical)	實體線路	IEEE 802.1、Ethernet

Let’s start traefik example0

Docker-compose.yml

```yaml
version: '3'

services:
  reverse-proxy:
    # The official v2.0 Traefik docker image
    image: traefik:v2.0
    # Enables the web UI and tells Traefik to listen to docker
    command: --api.insecure=true --providers.docker
    ports:
      # The HTTP port
      - "80:80"
      # The Web UI (enabled by --api.insecure=true)
      - "8080:8080"
    volumes:
      # So that Traefik can listen to the Docker events
      - /var/run/docker.sock:/var/run/docker.sock
  whoami:
    # A container that exposes an API to show its IP address
    image: containous/whoami
    labels:
      - "traefik.http.routers.whoami.rule=Host(`whoami.docker.localhost`)"
```

8080 port 上有 Dashboard

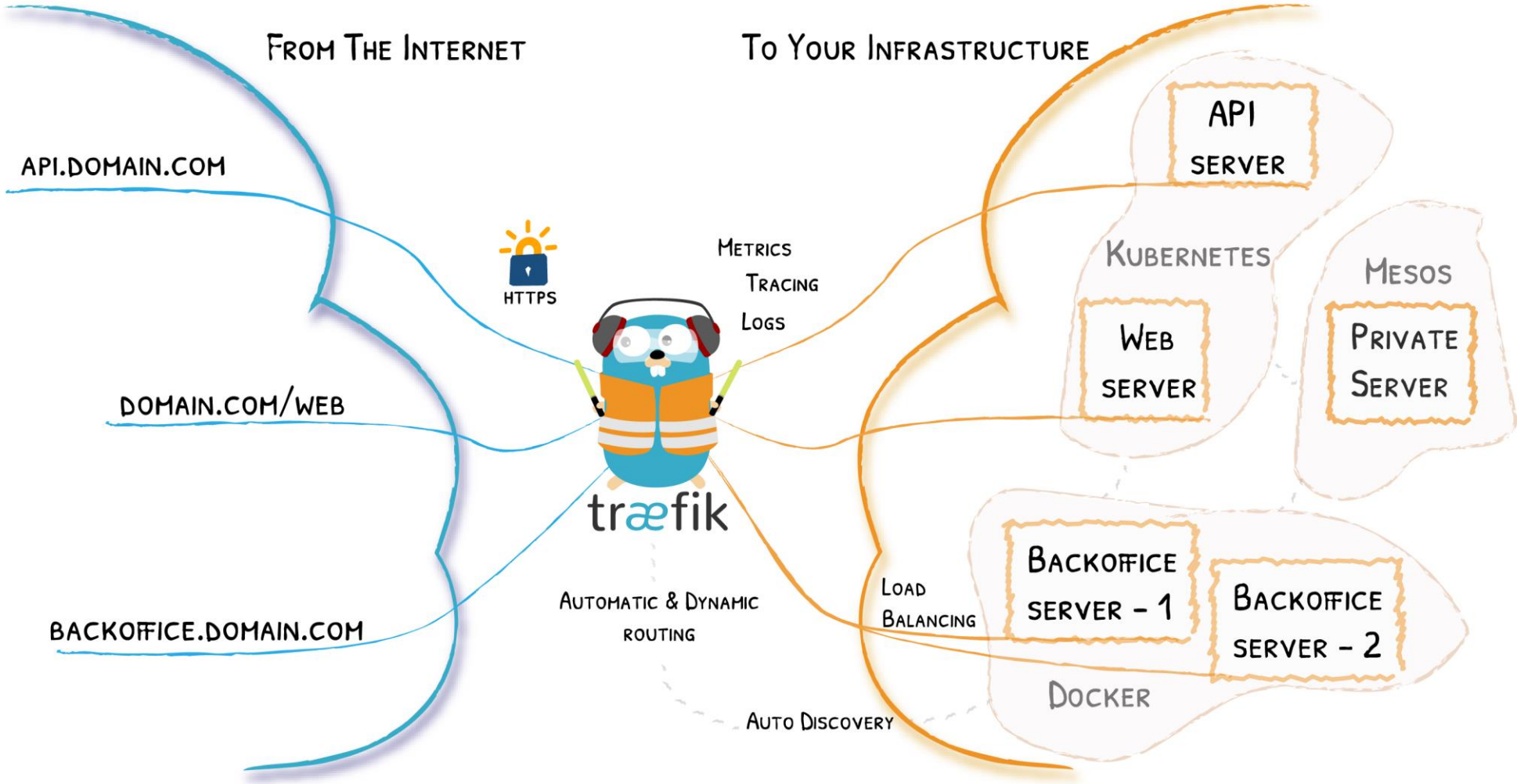
Try to scale out service~

Docker-compose.yml

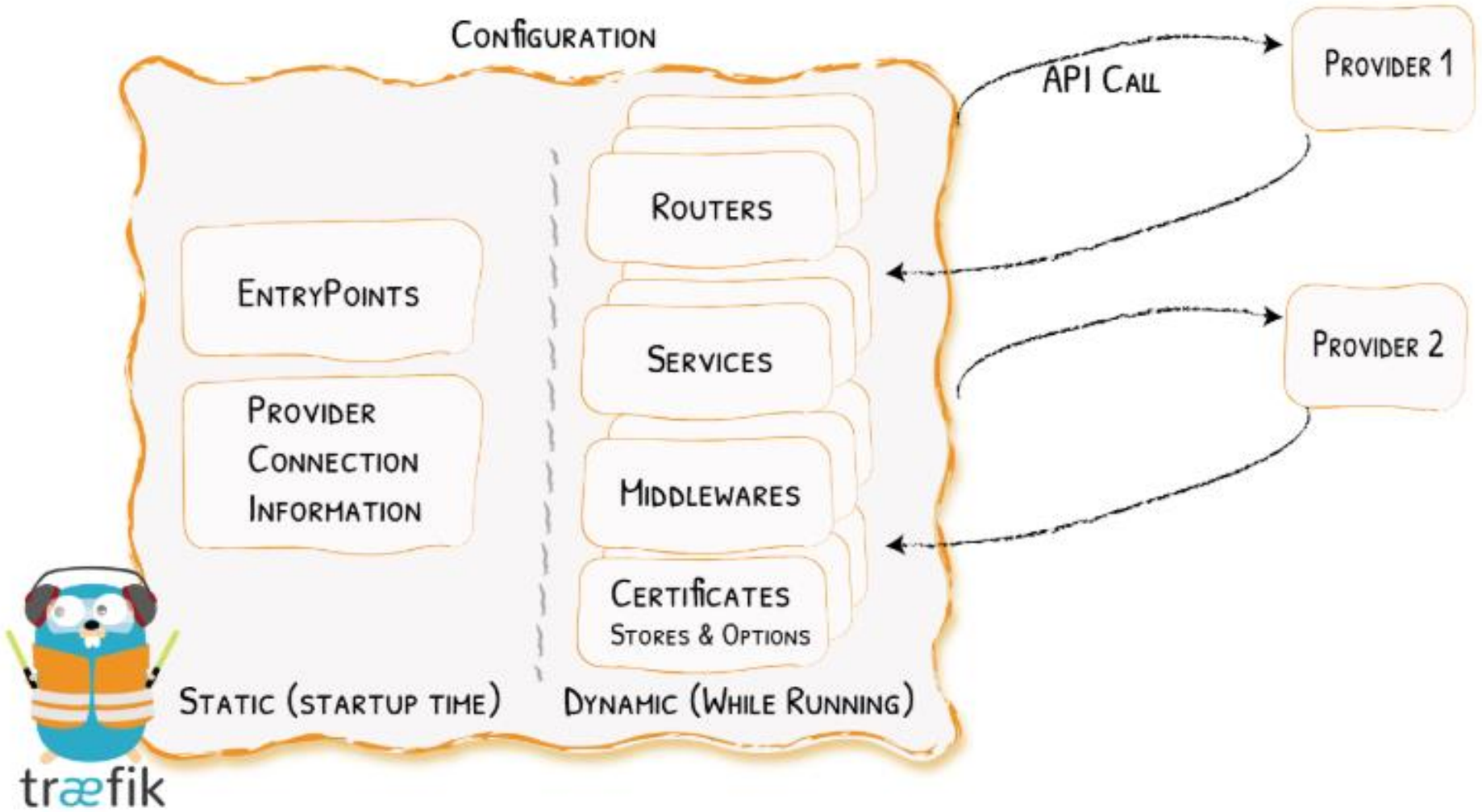
```yaml
version: '3'

services:
  reverse-proxy:
    # The official v2.0 Traefik docker image
    image: traefik:v2.0
    # Enables the web UI and tells Traefik to listen to docker
    command: --api.insecure=true --providers.docker
    ports:
      # The HTTP port
      - "80:80"
      # The Web UI (enabled by --api.insecure=true)
      - "8080:8080"
    volumes:
      # So that Traefik can listen to the Docker events
      - /var/run/docker.sock:/var/run/docker.sock
  whoami:
    # A container that exposes an API to show its IP address
    image: containous/whoami
    labels:
      - "traefik.http.routers.whoami.rule=Host(`whoami.docker.localhost`)"
```

???

Traefik


From The Internet
To Your Infrastructure
API.DOMAIN.COM
DOMAIN.COM/WEB
BACKOFFICE.DOMAIN.COM
HTTPS
Metrics
Tracing
Logs
træfik
Automatic & Dynamic routing
Load Balancing
Auto Discovery
API server
Kubernetes
Web server
Mesos
Private Server
Backoffice server - 1
Backoffice server - 2
Docker

Traefik Configuration

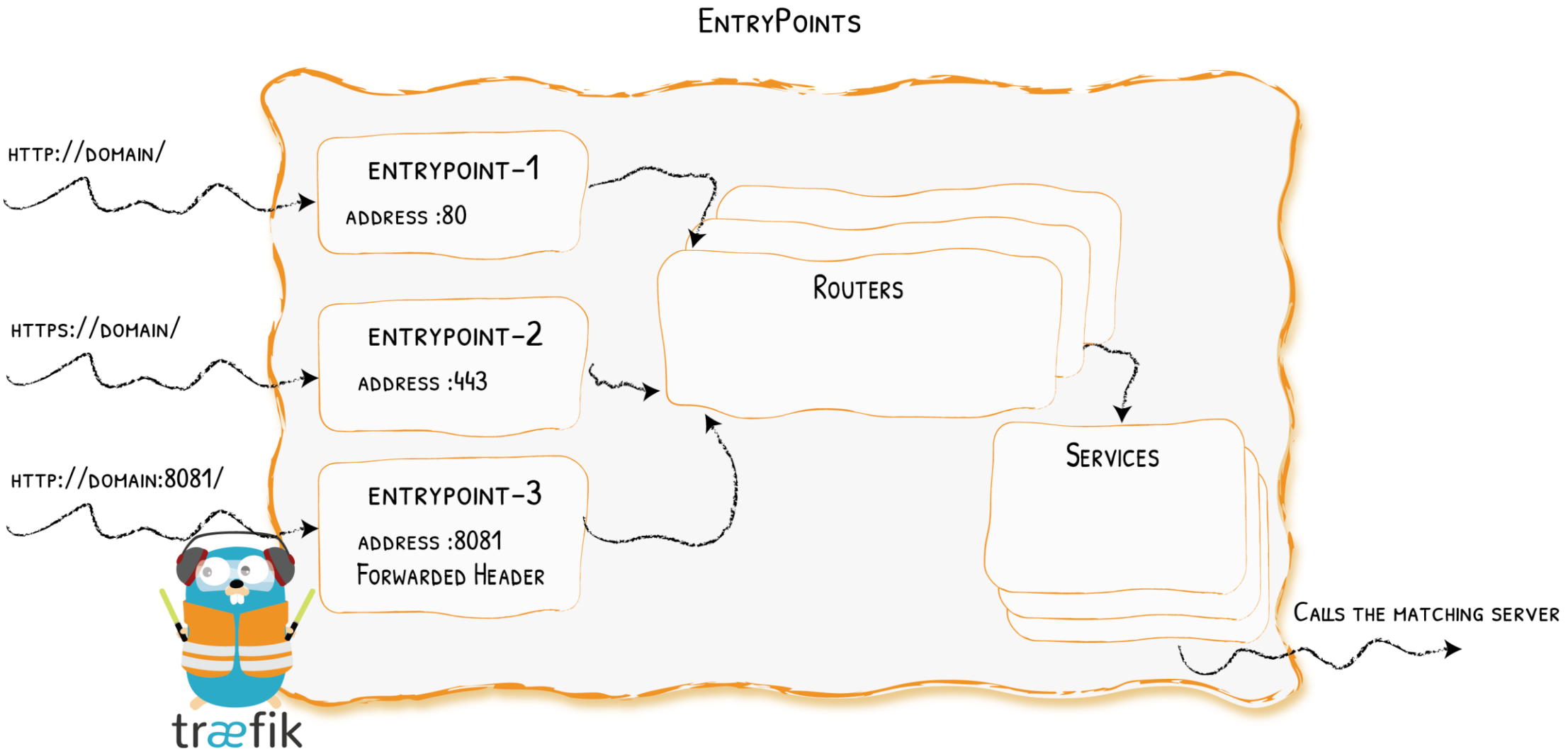


traefik.toml

- TOML (Tom's Obvious, Minimal Language)
- All configurations are here

```
volumes:
- /var/run/docker.sock:/var/run/docker.sock
- $PWD/traefik.toml:/etc/traefik/traefik.toml
```

EntryPoints



In traefik.toml

```
[entryPoints]

  [entryPoints.http]
    address = ":80"

  [entryPoints.traefik]
    address = ":8080"
```

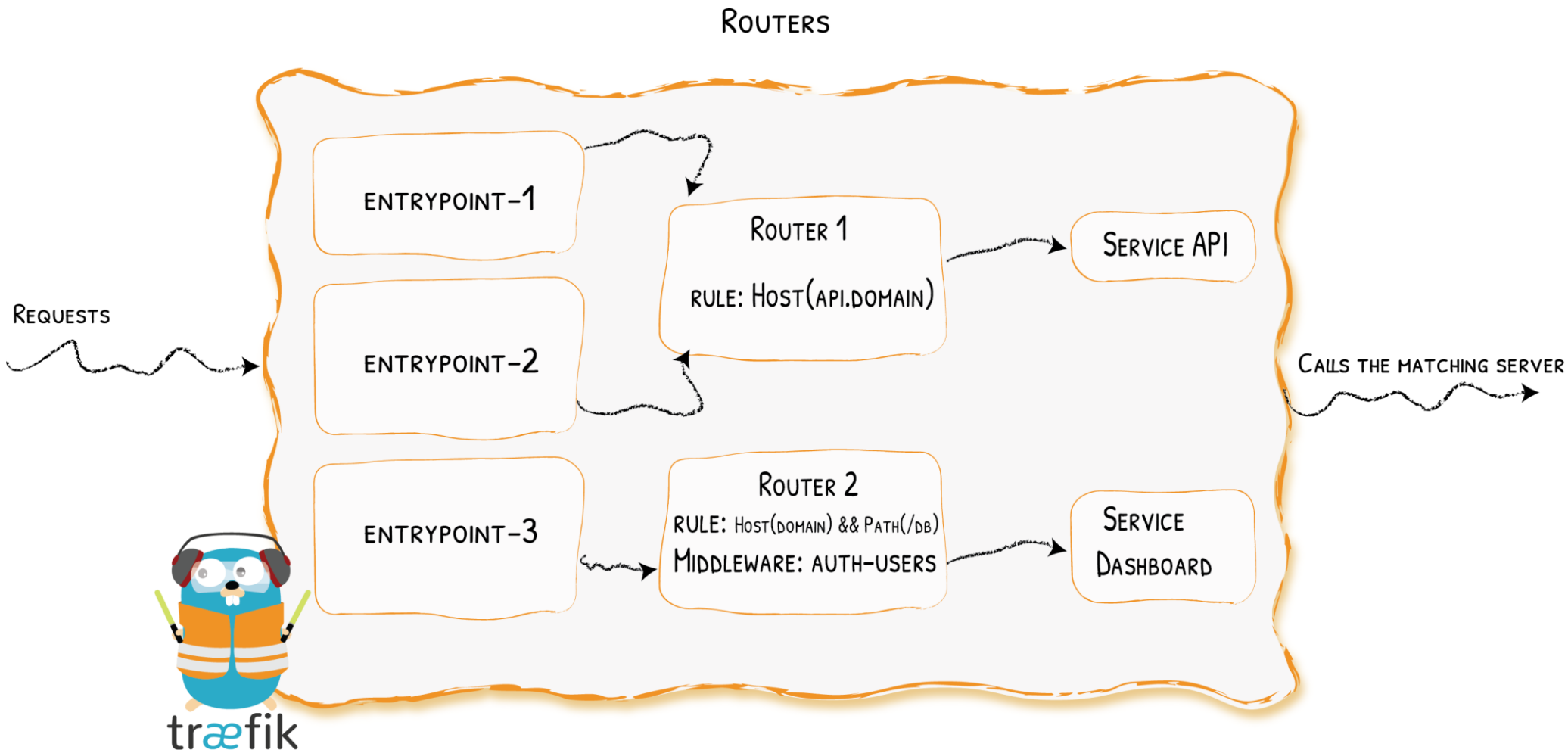
Try to start example1 & modify it

Provider

- File (need filename)
- Docker (listen docker.sock)
- Kubernetes (Ingress provider)
- Consul
- Marathon
- Rancher
- Docker Swarm

How about services not register?
docker run -it --name testcase containous/whoami

Routers


Routers
Requests
entrypoint-1
entrypoint-2
entrypoint-3
Router 1
rule: Host(api.domain)
Router 2
rule: Host(domain) && Path(/db)
Middleware: auth-users
Service API
Service Dashboard
Calls the matching server
traefik

Mapping rule

- Hearder
  rule = "Header(‘key’,’value’)"
- Host
  rule = "Host(‘traefik.io’)"
- Method
  rule = "Method(‘GET’)"
- Path
  rule = "Path(‘/traefik’)"
- PathPrefix
  rule = "PathPrefix(‘/products/’)"

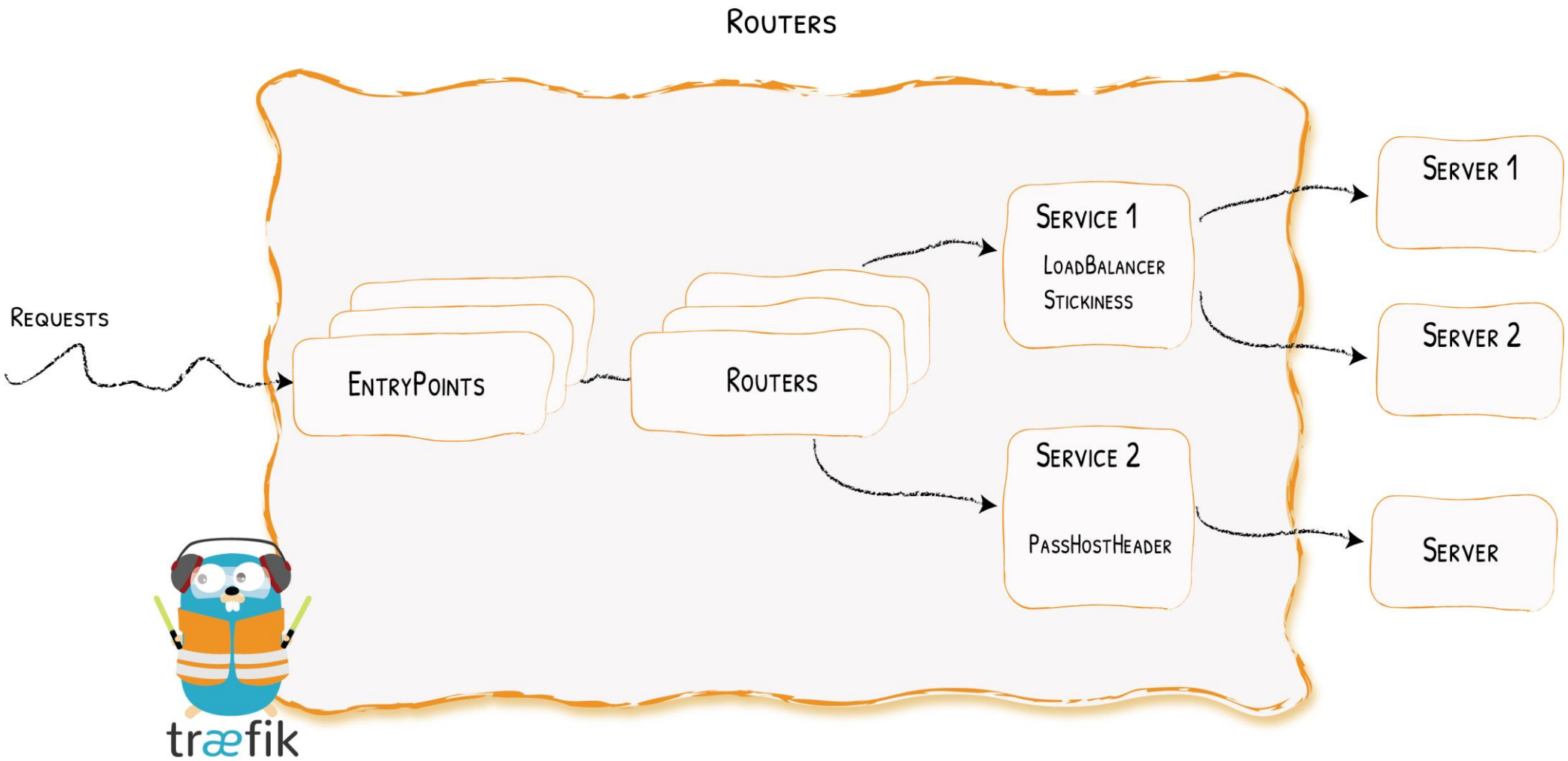
Router setting in Docker provider

```yaml
whoami:
  # A container that exposes an API to show its IP address
  image: containous/whoami
  labels:
    - "traefik.http.routers.whoami.rule=Host(`whoami.docker.localhost`)"
```

註冊　traefik
類型　http (http/tcp)
種類　routers(routers/services/...)
服務名 E.g., whoami
屬性　rule

Try to modify Host to PathPrefix

Services


Routers
Requests
EntryPoints
Routers
Service 1
LoadBalancer
Stickiness
Service 2
PassHostHeader
Server 1
Server 2
Server
træfik

Service setting

- Servers properties (E.g., path, port)
- Load Balance
- Sticky sessions
- Health Check
- Mirroring

Example 2_1

Why it can’t work!!!

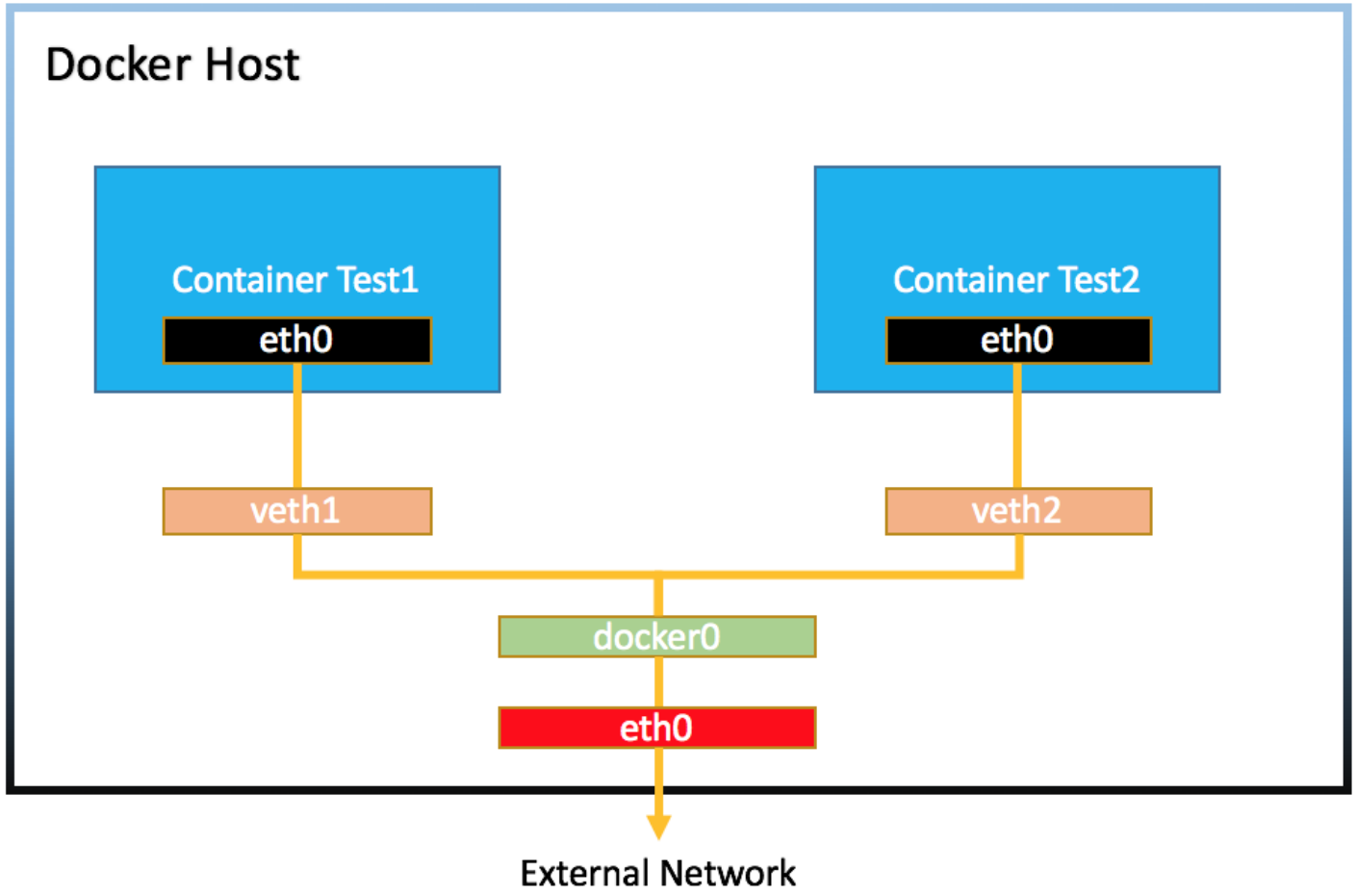
Gate Timeout? Network Error?

Traefik is a Layer 7 application, have no permission to control network

When network error

- You give a wrong routing path, correct it !
- The routing path never exists

Docker network



docker network ls

docker-compose network
=> {folder}_default

ip addr show | grep docker0

Example 2_2

Example3

File Provider

Other interesting function

- Middleware
- Logs
- tracing

Difference version

- V1 ⇔ V2