

TCP Middlewares Overview

TCP Middlewares in Traefik let you manage connections on the fly. You can configure them in multiple ways (Docker labels, TOML, YAML, etc.).

Configuration Example in a `docker-compose.yaml`

Configuration Example in a `docker-compose.yaml`

```
# As a Docker Label
whoami:
  # A container that exposes an API to show its IP address
  image: traefik/whoami
  labels:
    # Create a middleware named `foo-ip-allowlist`
    - "traefik.tcp.middlewares.foo-ip-allowlist.ipallowlist.sourcerange=127.0.0.1/32, 192.168.1.7"
    # Apply the middleware named `foo-ip-allowlist` to the router named `router1`
    - "traefik.tcp.routers.router1.middlewares=foo-ip-allowlist@docker"
```

Configuration Example in a `dynamic.yml`

Configuration Example in a `dynamic.yml`

```
tcp:
  routers:
    router1:
      service: myService
      middlewares:
        - "foo-ip-allowlist"
```

rule: "Host(`example.com`)"

middlewares:

foo-ip-allowlist:

ipAllowList:

sourceRange:

- "127.0.0.1/32"

- "192.168.1.7"

services:

service1:

loadBalancer:

servers:

- address: "10.0.0.10:4000"

- address: "10.0.0.11:4000"

Available TCP Middlewares

Middleware	Purpose	Area
<u>InFlightConn</u>	Limits the number of simultaneous connections.	Security, Request lifecycle
<u>IPAllowList</u>	Limit the allowed client IPs.	Security, Request lifecycle

Revision #2

Created 12 September 2024 14:10:18 by aeoneros

Updated 4 February 2025 14:32:51 by aeoneros