# Traefik & ECS


- Configuration Examples
- Configuration Options
  - General
  - Routers
    - Configuration Options
  - Services
    - Configuration Options
  - Middleware
  - TCP
    - TCP Routers
      - Configuration Options
    - TCP Services
      - Configuration Options
  - UDP
    - TCP Middleware
    - UDP Routers
      - Configuration Options
    - UDP Services
      - Configuration Options
  - Specific Provider Options
    - Configuration Options


One of the best feature of Traefik is to delegate the routing configuration to the application level. With ECS, Traefik can leverage labels attached to a container to generate routing rules.

!!! warning "Labels & sensitive data"

```
We recommend to *not* use labels to store sensitive data (certificates, credentials, etc).
Instead, we recommend to store sensitive data in a safer storage (secrets, file, etc).
```

# Configuration Examples

??? example "Configuring ECS & Deploying / Exposing one Service"

````
Enabling the ECS provider

```yaml tab="Structured (YAML)"
providers:
  ecs: {}
```

```toml tab="Structured (TOML)"
[providers.ecs]
```

```bash tab="CLI"
--providers.ecs=true
```

Attaching labels to containers (in your ECS task definition)

```json
{
  "family": "my-service",
  "containerDefinitions": [
    {
      "name": "my-container",
      "image": "my-image:latest",
      "labels": {
        "traefik.http.routers.my-container.rule": "Host(`example.com`)"
      }
    }
  ]
}
```
````

??? example "Specify a Custom Port for the Container"

````
Forward requests for `http://example.com` to `http://<private IP of container>:12345`:

```json
{
  "family": "my-service",
  "containerDefinitions": [
    {
      "name": "my-container",
      "image": "my-image:latest",
      "labels": {
        "traefik.http.routers.my-container.rule": "Host(`example.com`)",
        "traefik.http.routers.my-container.service": "my-service",
        "traefik.http.services.my-service.loadbalancer.server.port": "12345"
      }
    }
  ]
}
```

!!! important "Traefik Connecting to the Wrong Port: `HTTP/502 Gateway Error`"
    By default, Traefik uses the first exposed port of a container.

    Setting the label `traefik.http.services.xxx.loadbalancer.server.port`
    overrides that behavior.
````

??? example "Specifying more than one router and service per container"

````
Forwarding requests to more than one port on a container requires referencing the service
loadbalancer port definition using the service parameter on the router.

In this example, requests are forwarded for `http://example-a.com` to `http://<private IP of
container>:8000` in addition to `http://example-b.com` forwarding to `http://<private IP of
container>:9000`:

```json
{
  "family": "my-service",
  "containerDefinitions": [
    {
````
````

```
      "name": "my-container",
      "image": "my-image:latest",
      "labels": {
        "traefik.http.routers.www-router.rule": "Host(`example-a.com`)",
        "traefik.http.routers.www-router.service": "www-service",
        "traefik.http.services.www-service.loadbalancer.server.port": "8000",
        "traefik.http.routers.admin-router.rule": "Host(`example-b.com`)",
        "traefik.http.routers.admin-router.service": "admin-service",
        "traefik.http.services.admin-service.loadbalancer.server.port": "9000"
      }
    }
  ]
}
```

# Configuration Options

!!! info "labels"

```
Labels are case-insensitive.
```

!!! tip "TLS Default Generated Certificates"

```
To learn how to configure Traefik default generated certificate, refer to the [TLS
Certificates](../http/tls/tls-certificates.md#acme-default-certificate) page.
```

# General

Traefik creates, for each elastic service, a corresponding service and router.

The Service automatically gets a server per elastic container, and the router gets a default rule attached to it, based on the service name.

# Routers

To update the configuration of the Router automatically attached to the service, add labels starting with `traefik.routers.{name-of-your-choice}.` and followed by the option you want to change.

For example, to change the rule, you could add the label `traefik.http.routers.my-service.rule=Host(`example.com`)`.

!!! warning "The character `@` is not authorized in the router name `<router_name>`."

# Configuration Options

| Label | Description | Value |
|---|---|---|
| `traefik.http.routers.<router_name>.rule` | See rule for more information. | `Host(`example.com`)` |
| `traefik.http.routers.<router_name>.ruleSyntax` | See ruleSyntax for more information. RuleSyntax option is deprecated and will be removed in the next major version. Please do not use this field and rewrite the router rules to use the v3 syntax. | `v3` |
| `traefik.http.routers.<router_name>.entrypoints` | See entry points for more information. | `web,websecure` |
| `traefik.http.routers.<router_name>.middlewares` | See middlewares overview for more information. | `auth,prefix,cb` |
| `traefik.http.routers.<router_name>.service` | See service for more information. | `myservice` |
| `traefik.http.routers.<router_name>.tls` | See tls for more information. | `true` |
| `traefik.http.routers.<router_name>.tls.certresolver` | See certResolver for more information. | `myresolver` |
| `traefik.http.routers.<router_name>.tls.domains[n].main` | See domains for more information. | `example.org` |
| `traefik.http.routers.<router_name>.tls.domains[n].sans` | See domains for more information. | `test.example.org,dev.example.org` |
| `traefik.http.routers.<router_name>.tls.options` | See TLS Options for more information, and Conflicting TLS Options when several routers serve the same host name. | `foobar` |
| `traefik.http.routers.<router_name>.observability.accesslogs` | The accessLogs option controls whether the router will produce access-logs. | `true` |
| `traefik.http.routers.<router_name>.observability.metrics` | The metrics option controls whether the router will produce metrics. | `true` |
| `traefik.http.routers.<router_name>.observability.tracing` | The tracing option controls whether the router will produce traces. | `true` |
| `traefik.http.routers.<router_name>.priority` | See priority for more information. | `42` |

# Services

To update the configuration of the Service automatically attached to the service, add labels starting with `traefik.http.services.{name-of-your-choice}.`, followed by the option you want to change.

For example, to change the `passHostHeader` behavior, you'd add the label `traefik.http.services.{name-of-your-choice}.loadbalancer.passhostheader=false`.

!!! warning "The character `@` is not authorized in the service name `<service_name>`."

## Configuration Options

| Label | Description | Value |
|---|---|---|
| `traefik.http.services.<service_name>.loadbalancer.server.port` | Registers a port.<br>Useful when the service exposes multiples ports. | `8080` |
| `traefik.http.services.<service_name>.loadbalancer.server.scheme` | Overrides the default scheme. | `http` |
| `traefik.http.services.<service_name>.loadbalancer.serverstransport` | Allows to reference a ServersTransport resource that is defined either with the File provider or the Kubernetes CRD one.<br>See serverstransport for more information. | `foobar@file` |
| `traefik.http.services.<service_name>.loadbalancer.passhostheader` | | `true` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.headers.<header_name>` | See health check for more information. | `foobar` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.hostname` | See health check for more information. | `example.org` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.interval` | See health check for more information. | `10` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.unhealthyinterval` | See health check for more information. | `10` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.path` | See health check for more information. | `/foo` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.method` | See health check for more information. | `foobar` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.status` | See health check for more information. | `42` |

| Label | Description | Value |
|---|---|---|
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.port` | See health check for more information. | `42` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.scheme` | See health check for more information. | `http` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.timeout` | See health check for more information. | `10` |
| `traefik.http.services.<service_name>.loadbalancer.healthcheck.followredirects` | See health check for more information. | `true` |
| `traefik.http.services.<service_name>.loadbalancer.sticky.cookie` | | `true` |
| `traefik.http.services.<service_name>.loadbalancer.sticky.cookie.httponly` | | `true` |
| `traefik.http.services.<service_name>.loadbalancer.sticky.cookie.name` | | `foobar` |
| `traefik.http.services.<service_name>.loadbalancer.sticky.cookie.path` | | `/foobar` |
| `traefik.http.services.<service_name>.loadbalancer.sticky.cookie.secure` | | `true` |
| `traefik.http.services.<service_name>.loadbalancer.sticky.cookie.samesite` | | `none` |
| `traefik.http.services.<service_name>.loadbalancer.sticky.cookie.maxage` | | `42` |
| `traefik.http.services.<service_name>.loadbalancer.responseforwarding.flushinterval` | `FlushInterval` specifies the flush interval to flush to the client while copying the response body. | `10` |

# Middleware

You can declare pieces of middleware using labels starting with `traefik.http.middlewares.{name-of-your-choice}.`, followed by the middleware type/options.

For example, to declare a middleware `redirectscheme` named `my-redirect`, you'd write `traefik.http.middlewares.my-redirect.redirectscheme.scheme: https`.

More information about available middlewares in the dedicated middlewares section.

!!! warning "The character `@` is not authorized in the middleware name."

??? example "Declaring and Referencing a Middleware"

```yaml
# ...
# Declaring a middleware
traefik.http.middlewares.my-redirect.redirectscheme.scheme=https
# Referencing a middleware
traefik.http.routers.my-service.middlewares=my-redirect
```

!!! warning "Conflicts in Declaration"

```
If you declare multiple middleware with the same name but with different parameters, the
middleware fails to be declared.
```

# TCP

You can declare TCP Routers and/or Services using labels.

??? example "Declaring TCP Routers and Services"

```yaml
traefik.tcp.routers.my-router.rule=HostSNI(`example.com`)
traefik.tcp.routers.my-router.tls=true
traefik.tcp.services.my-service.loadbalancer.server.port=4123
```

!!! warning "TCP and HTTP"

```
If you declare a TCP Router/Service, it will prevent Traefik from automatically creating an
HTTP Router/Service (like it does by default if no TCP Router/Service is defined).
You can declare both a TCP Router/Service and an HTTP Router/Service for the same elastic
service (but you have to do so manually).
```

## TCP Routers

### Configuration Options

| Label | Description | Value |
|---|---|---|
| `traefik.tcp.routers.<router_name>.entrypoints` | See entry points for more information. | `ep1,ep2` |
| `traefik.tcp.routers.<router_name>.rule` | See rule for more information. | `HostSNI(`example.com`)` |

| Label | Description | Value |
|---|---|---|
| `traefik.tcp.routers.<router_name>.ruleSyntax` | configure the rule syntax to be used for parsing the rule on a per-router basis.<br>RuleSyntax option is deprecated and will be removed in the next major version.<br>Please do not use this field and rewrite the router rules to use the v3 syntax. | `v3` |
| `traefik.tcp.routers.<router_name>.service` | See service for more information. | `myservice` |
| `traefik.tcp.routers.<router_name>.tls` | See TLS for more information. | `true` |
| `traefik.tcp.routers.<router_name>.tls.certresolver` | See certResolver for more information. | `myresolver` |
| `traefik.tcp.routers.<router_name>.tls.domains[n].main` | See TLS for more information. | `example.org` |
| `traefik.tcp.routers.<router_name>.tls.domains[n].sans` | See TLS for more information. | `test.example.org,dev.example.org` |
| `traefik.tcp.routers.<router_name>.tls.options` | See TLS for more information. | `mysoptions` |
| `traefik.tcp.routers.<router_name>.tls.passthrough` | See Passthrough for more information. | `true` |
| `traefik.tcp.routers.<router_name>.priority` | See priority for more information. | `42` |

# TCP Services

## Configuration Options

| Label | Description | Value |
|---|---|---|
| `traefik.tcp.services.<service_name>.loadbalancer.server.port` | Registers a port of the application. | `423` |
| `traefik.tcp.services.<service_name>.loadbalancer.server.tls` | Determines whether to use TLS when dialing with the backend. | `true` |
| `traefik.tcp.services.<service_name>.loadbalancer.serverstransport` | Allows to reference a ServersTransport resource that is defined either with the File provider or the Kubernetes CRD one.<br>See serverstransport for more information. | `foobar@file` |

# UDP

You can declare UDP Routers and/or Services using tags.

??? example "Declaring UDP Routers and Services"

```
```yaml
traefik.udp.routers.my-router.entrypoints=udp
traefik.udp.services.my-service.loadbalancer.server.port=4123
```
```

!!! warning "UDP and HTTP"

```
If you declare a UDP Router/Service, it will prevent Traefik from automatically creating an
HTTP Router/Service (like it does by default if no UDP Router/Service is defined).
You can declare both a UDP Router/Service and an HTTP Router/Service for the same elastic
service (but you have to do so manually).
```

## TCP Middleware

You can declare pieces of middleware using tags starting with `traefik.tcp.middlewares.{name-of-your-choice}.`, followed by the middleware type/options.

For example, to declare a middleware `InFlightConn` named `test-inflightconn`, you'd write `traefik.tcp.middlewares.test-inflightconn.inflightconn.amount=10`.

More information about available middlewares in the dedicated middlewares section.

??? example "Declaring and Referencing a Middleware"

```
```yaml
# ...
# Declaring a middleware
traefik.tcp.middlewares.test-inflightconn.inflightconn.amount=10
# Referencing a middleware
traefik.tcp.routers.my-service.middlewares=test-inflightconn
```
```

!!! warning "Conflicts in Declaration"

```
If you declare multiple middleware with the same name but with different parameters, the
middleware fails to be declared.
```

## UDP Routers

### Configuration Options

| Label | Description | Value |
|---|---|---|
| `traefik.udp.routers.<router_name>.entrypoints` | See entry points for more information. | `ep1,ep2` |
| `traefik.udp.routers.<router_name>.service` | See service for more information. | `myservice` |

## UDP Services

### Configuration Options

| Label | Description | Value |
|---|---|---|
| `traefik.udp.services.<service_name>.loadbalancer.server.port` | Registers a port of the application. | `423` |

# Specific Provider Options

## Configuration Options

| Label | Description | Value |
|---|---|---|
| `traefik.enable` | You can tell Traefik to consider (or not) the ECS service by setting `traefik.enable` to true or false. This option overrides the value of `exposedByDefault`. | `true` |

---

□□□ #2
□ □□□ □□□ 2026-10-31 14:05:24 UTC
□ □□□ □□□ 2026-10-31 14:05:25 UTC