Tailscale

- Configuration Example
- Domain Definition
- Tailscale Certificates Renewal

Provision TLS certificates for your internal Tailscale services. {: .subtitle }

To protect a service with TLS, a certificate from a public Certificate Authority is needed. In addition to its vpn role, Tailscale can also provide certificates for the machines in your Tailscale network.

Configuration Example

To obtain a TLS certificate from the Tailscale daemon, a Tailscale certificate resolver needs to be configured as below.

!!! example "Enabling Tailscale certificate resolution"

````
```yaml tab="File (YAML)"
entryPoints:
  web:
    address: ":80"

  websecure:
    address: ":443"

certificatesResolvers:
  myresolver:
    tailscale: {}
```

```toml tab="File (TOML)"
[entryPoints]
````
````

```
  [entryPoints.web]
    address = ":80"

  [entryPoints.websecure]
    address = ":443"

[certificatesResolvers.myresolver.tailscale]
```

```bash tab="CLI"
--entrypoints.web.address=:80
--entrypoints.websecure.address=:443
# ...
--certificatesresolvers.myresolver.tailscale=true
```

??? example "Domain from Router's Rule Example"

```yaml tab="Docker & Swarm"
labels:
  - traefik.http.routers.blog.rule=Host(`monitoring.yak-bebop.ts.net`) && Path(`/metrics`)
  - traefik.http.routers.blog.tls.certresolver=myresolver
```

```yaml tab="Docker (Swarm)"
deploy:
  labels:
    - traefik.http.routers.blog.rule=Host(`monitoring.yak-bebop.ts.net`) && Path(`/metrics`)
    - traefik.http.routers.blog.tls.certresolver=myresolver
```

```yaml tab="Kubernetes"
apiVersion: traefik.io/v1alpha1
kind: IngressRoute
metadata:
  name: blogtls
spec:
  entryPoints:
    - websecure
  routes:
```

```
    - match: Host(`monitoring.yak-bebop.ts.net`) && Path(`/metrics`)
      kind: Rule
      services:
        - name: blog
          port: 8080
  tls:
    certResolver: myresolver
```

```yaml tab="File (YAML)"
## Dynamic configuration
http:
  routers:
    blog:
      rule: "Host(`monitoring.yak-bebop.ts.net`) && Path(`/metrics`)"
      tls:
        certResolver: myresolver
```

```toml tab="File (TOML)"
## Dynamic configuration
[http.routers]
  [http.routers.blog]
  rule = "Host(`monitoring.yak-bebop.ts.net`) && Path(`/metrics`)"
  [http.routers.blog.tls]
    certResolver = "myresolver"
```

??? example "Domain from Router's tls.domain Example"

```yaml tab="Docker & Swarm"
labels:
  - traefik.http.routers.blog.rule=Path(`/metrics`)
  - traefik.http.routers.blog.tls.certresolver=myresolver
  - traefik.http.routers.blog.tls.domains[0].main=monitoring.yak-bebop.ts.net
```

```yaml tab="Docker (Swarm)"
deploy:
  labels:
```

```
    - traefik.http.routers.blog.rule=Path(`/metrics`)
    - traefik.http.routers.blog.tls.certresolver=myresolver
    - traefik.http.routers.blog.tls.domains[0].main=monitoring.yak-bebop.ts.net
```

```yaml tab="Kubernetes"
apiVersion: traefik.io/v1alpha1
kind: IngressRoute
metadata:
  name: blogtls
spec:
  entryPoints:
    - websecure
  routes:
    - match: Path(`/metrics`)
      kind: Rule
      services:
        - name: blog
          port: 8080
  tls:
    certResolver: myresolver
    domains:
      - main: monitoring.yak-bebop.ts.net
```

```yaml tab="File (YAML)"
http:
  routers:
    blog:
      rule: "Path(`/metrics`)"
      tls:
        certResolver: myresolver
        domains:
          - main: "monitoring.yak-bebop.ts.net"
```

```toml tab="File (TOML)"
## Dynamic configuration
[http.routers]
  [http.routers.blog]
```

```
    rule = "Path(`/metrics`)"
    [http.routers.blog.tls]
      certResolver = "myresolver"
      [[http.routers.blog.tls.domains]]
        main = "monitoring.yak-bebop.ts.net"
```

!!! info "Referencing a certificate resolver"

```
Defining a certificate resolver does not imply that routers are going to use it automatically.
Each router or entrypoint that is meant to use the resolver must explicitly
[reference](../../../routing-configuration/http/routing/router.md#opt-tls-certResolver) it.
```

Domain Definition

A certificate resolver requests certificates for a set of domain names inferred from routers, according to the following:

- If the router has a `tls.domains` option set, then the certificate resolver derives this router domain name from the main option of `tls.domains`.
- Otherwise, the certificate resolver derives the domain name from any `Host()` or `HostSNI()` matchers in the router's rule.

!!! info "Tailscale Domain Format"

```
A domain is only considered if it is a Tailscale-specific one—that is, in the form `machine-
name.domains-alias.ts.net`.
```

Tailscale Certificates Renewal

Traefik automatically tracks the expiry date of each Tailscale certificate it fetches and starts to renew a certificate 14 days before its expiry to match the Tailscale daemon renewal policy.

□□□ #2
□ □□□ □□□ 2026-10-31 14:01:48 UTC
□ □□□ □□□ 2026-10-31 14:01:49 UTC