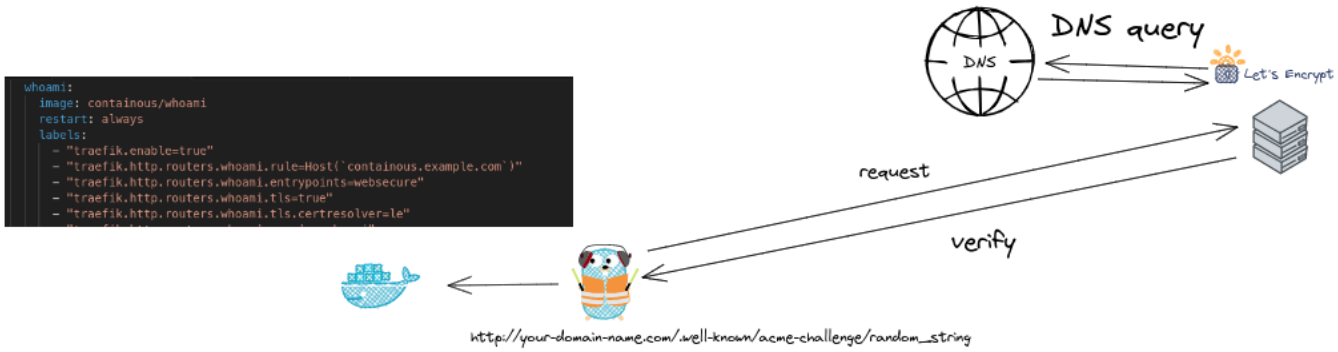
# Let's Encrypt: How does it Work? & More

## Overview

Traefik can automatically generate and renew TLS certificates using an ACME provider, such as Let’s Encrypt. This simplifies certificate management while ensuring secure HTTPS connections.

## How does Let's Encrypt Work?

Let's Encrypt is a free, automated, and open Certificate Authority that provides SSL/TLS certificates to enable encrypted connections to websites. Its mission is to create a more secure and privacy-respecting web by promoting the widespread adoption of HTTPS.




Let's Encrypt follows a protocol called ACME (Automated Certificate Management Environment) for the issuance and renewal of certificates. Here's a simplified version of how it works:

### Step 1: Domain Ownership Verification

To ensure that the certificate isn't provided to malicious entities, Let's Encrypt needs to confirm that the entity controlling the domain is the one requesting the certificate. This is achieved through

a challenge-response protocol.

In our example with Traefik, we're using the HTTP challenge method, where Traefik will be asked to put a specific file at a known location on your site, like `http://your-domain.com/.well-known/acme-challenge/random_string`. Let's Encrypt then verifies the file's presence, thus confirming domain ownership.

### Step 2: Certificate Issuance and Installation

Once domain ownership is confirmed, Let's Encrypt issues a certificate that can be installed on your server. Traefik automates this step by pulling the issued certificate and storing it locally, then using it whenever an HTTPS connection is initiated.

### Step 3: Automatic Renewal

Let's Encrypt certificates are valid for 90 days, but you don't have to worry about the expiration date because Traefik will automatically renew the certificates before they expire, as long as the ACME configuration is correct.

By automating the process of issuing, installing, and renewing certificates, Let's Encrypt has significantly lowered the complexity of setting up and maintaining HTTPS on a website.

> If you want to Setup your Own TLS-Challenge go check out this Guide: Docker-compose with Let's Encrypt: TLS Challenge

---

# More about Lets Encrypt & Traefik

## Let’s Encrypt and Rate Limiting

Let’s Encrypt imposes rate limits for API requests, which last up to one week and cannot be overridden. To avoid reaching these limits:

- Persist the `acme.json` file across container restarts to prevent Traefik from requesting new certificates unnecessarily.
- Use the Let’s Encrypt staging server with the `caServer` configuration option for testing.

## Certificate Resolvers

Certificate resolvers are defined in the static configuration and retrieve certificates from an ACME server. Each router that requires a certificate must reference a resolver explicitly using the `tls.certresolver` option.

## Configuration Reference

- **Domain Definition:** Certificates are requested for domain names specified in the router's dynamic configuration. Multiple domain names are supported, with one acting as the main domain and others as Subject Alternative Names (SANs).
- **ACME Challenges:** Each resolver must define an ACME challenge type: HTTP-01, DNS-01, or TLS-ALPN-01.

# Configuration Examples

## Enable ACME

```
# Static configuration
entryPoints:
  web:
    address: ":80"
  websecure:
    address: ":443"

certificatesResolvers:
  myresolver:
    acme:
      email: your-email@example.com
      storage: acme.json
      httpChallenge:
        entryPoint: web
```

## Single Domain from Router's Rule

```
# Dynamic configuration
labels:
  - traefik.http.routers.blog.rule=Host(`example.com`) && Path(`/blog`)
  - traefik.http.routers.blog.tls=true
  - traefik.http.routers.blog.tls.certresolver=myresolver
```

## Multiple Domains from Router's Rule

```
# Dynamic configuration
labels:
  - traefik.http.routers.blog.rule=(Host(`example.com`) && Path(`/blog`)) || Host(`blog.example.org`)
  - traefik.http.routers.blog.tls=true
  - traefik.http.routers.blog.tls.certresolver=myresolver
```

## Multiple Domains from Router's `tls.domain`

```
# Dynamic configuration
labels:
  - traefik.http.routers.blog.rule=Host(`example.com`) && Path(`/blog`)
  - traefik.http.routers.blog.tls=true
  - traefik.http.routers.blog.tls.certresolver=myresolver
  - traefik.http.routers.blog.tls.domains[0].main=example.com
  - traefik.http.routers.blog.tls.domains[0].sans=*.example.org
```

# ACME Challenges

## HTTP-01 Challenge

```
# Static configuration
entryPoints:
  web:
    address: ":80"
  websecure:
    address: ":443"

certificatesResolvers:
  myresolver:
    acme:
      httpChallenge:
        entryPoint: web
```

## DNS-01 Challenge

```
# Static configuration
certificatesResolvers:
  myresolver:
```

```
    acme:
      dnsChallenge:
        provider: digitalocean
        delayBeforeCheck: 0
        resolvers:
          - "1.1.1.1:53"
          - "8.8.8.8:53"
```

## TLS-ALPN-01 Challenge

```
# Static configuration
certificatesResolvers:
  myresolver:
    acme:
      tlsChallenge: {}
```

# Automatic Renewals

Traefik manages 90-day certificates and renews them automatically 30 days before expiry. For resolvers issuing custom-duration certificates, configure the renewal duration with the `certificatesDuration` option.

For more details, refer to the official Traefik documentation.

---

Revision #8
Created 12 September 2024 14:09:37 by aeoneros
Updated 11 February 2025 09:12:04 by aeoneros