Docker-compose with Let's Encrypt : HTTP Challenge

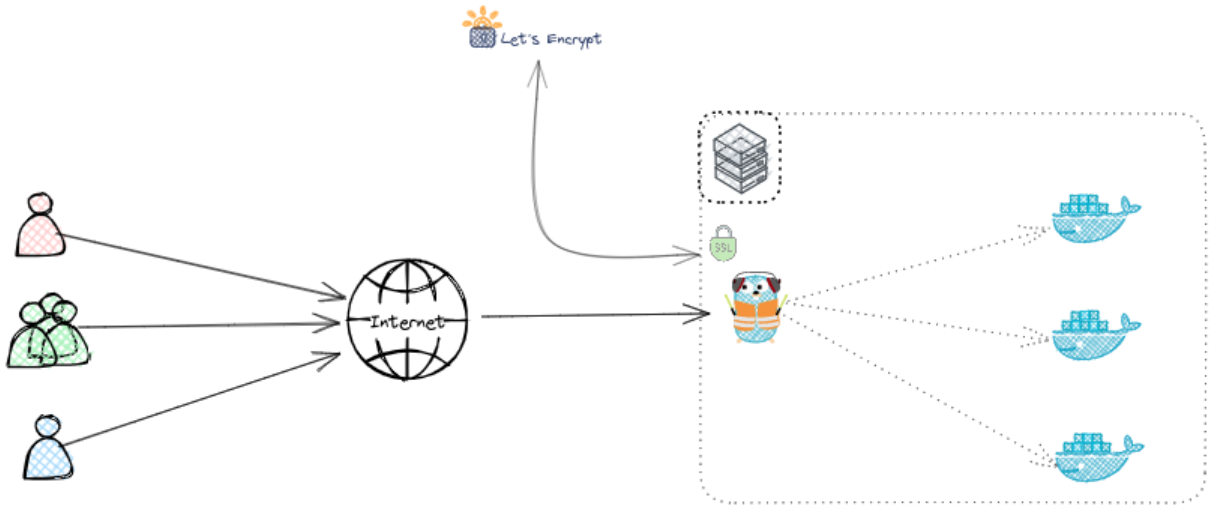
Introduction

This guide provides information on how to set up a simple **HTTP-Challenge** for Traefik to use Let's Encrypt and certify your domains/websites. We will configure Traefik to act as a reverse proxy for a simple "Whoami" application and secure the app using Let's Encrypt.

Understanding Let's Encrypt: Check this guide.

Overview of HTTP-Challenge

The **HTTP-01 challenge** is a method used by Let's Encrypt to verify domain ownership. Let's Encrypt requests a specific file to be available at `http://your-domain/.well-known/acme-challenge/`. Traefik serves this file, and Let's Encrypt verifies its presence to confirm domain ownership.



Difference Between HTTP-Challenge & TLS-Challenge

The **HTTP Challenge** uses HTTP requests on port 80 to verify domain ownership by serving a specific file at `http://your-domain/.well-known/acme-challenge/`. The **TLS Challenge** verifies ownership during the TLS handshake on port 443 by presenting a special certificate, making it more suitable for HTTPS-only environments or when port 80 is blocked.

Prerequisite

For the HTTP challenge you will need:

- A publicly accessible host allowing connections on port `80` with Docker and Docker Compose installed.
- A DNS record with the domain you want to expose pointing to this host.

Step 0: Configuring DNS Records

Before proceeding, make sure your domain name is correctly configured. Create a **DNS A Record** that points your domain to the public IP address of your server.

If you don't know what a DNS A Record is, check out this post from Cloudflare.

Step 1: Create ACME File

In this guide, we will store the ACME data in a `letsencrypt` directory within the folder containing your Docker Compose file. This folder will store your certificates.

```
mkdir ./letsencrypt
touch ./letsencrypt/acme.json
chmod 600 ./letsencrypt/acme.json
```

Step 2: Installing and Configuring Traefik

There are multiple ways to set up your Traefik configuration—either directly in the `docker-compose.yaml` file or by outsourcing it to external configuration files. Find more information here.

In this setup, we will configure the HTTP challenge for Let's Encrypt directly in the `docker-compose.yaml` file.

docker-compose.yaml

```
version: "3.3"
services:
  traefik:
    image: "traefik:v3.3"
    container_name: "traefik"
    command:
      - "--api.insecure=true"
      - "--providers.docker=true"
      - "--providers.docker.exposedbydefault=false"
      - "--entryPoints.web.address=:80"
      - "--entryPoints.websecure.address=:443"
      - "--certificatesresolvers.myresolver.acme.httpchallenge=true"
      - "--certificatesresolvers.myresolver.acme.httpchallenge.entrypoint=web"
      - "--certificatesresolvers.myresolver.acme.email=yourmail@example.com"
      - "--certificatesresolvers.myresolver.acme.storage=/letsencrypt/acme.json"
    ports:
      - "80:80"
      - "443:443"
      - "8080:8080"
    volumes:
      - "./letsencrypt:/letsencrypt"
      - "/var/run/docker.sock:/var/run/docker.sock:ro"
```

> Replace `yourmail@example.com` with your actual email address and `whoami.example.com` with your domain name.

Step 3: Integrating Let's Encrypt

You can now integrate automatic certification for your apps by ensuring they are configured with Traefik labels to use the `myresolver` certificate resolver.

```
  whoami:
    image: "traefik/whoami"
    container_name: "simple-service"
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.whoami.rule=Host(`whoami.example.com`)"
      - "traefik.http.routers.whoami.entrypoints=websecure"
```

```
    - "traefik.http.routers.whoami.tls.certresolver=myresolver"
```

Step 4: Starting the Services

Start the services with the following command (only works if your working directory is where your `docker-compose.yaml` file is saved):

```
docker-compose up -d
```

You should now be able to access your Whoami application over HTTPS, secured by a Let's Encrypt certificate.

Conclusion

In this guide, we demonstrated how to set up Traefik as a reverse proxy with Let's Encrypt using the HTTP-Challenge to secure a simple Whoami application. By following these steps, you can apply the same configuration to your own services and ensure secure communication with HTTPS.

Revision #2
Created 12 September 2024 14:14:00 by aeoneros
Updated 11 February 2025 09:58:07 by aeoneros