

[zurück](#)

Traefik Reverse Proxy Setup mit Let's Encrypt

□ **Stand: 2025-05-27**

□ Zielsetzung

Ein Reverse-Proxy mit **Traefik** in Docker soll mehrere interne Dienste über Subdomains erreichbar machen.

Dabei übernimmt Traefik:

- TLS-Terminierung (Let's Encrypt-Zertifikate)
- Subdomain-basiertes Routing
- Middleware-Redirects (z. B. /admin, /dokuwiki)
- Automatische Zertifikatsverwaltung über ACME (HTTP-Challenge)

⚙️ Setup-Details

□ Subdomains (über `dedyn.io`)

Dienst	Subdomain	Zielintern (HTTP)
-----	-----	-----
Portainer	portainer.xxxxxx.dedyn.io	Docker-Port 9000
Traefik-Dashboard	traefik.xxxxxx.dedyn.io	Interner Dienst (api@internal)
Pi-hole	pihole.xxxxxx.dedyn.io	http://192.168.xxx.xxx/admin
DokuWiki	wiki.xxxxxx.dedyn.io	http://192.168.xxx.xxx/dokuwiki/
phpMyAdmin	phpmyadmin.xxxxxx.dedyn.io	http://192.168.xxx.xxx/phpmyadmin
Webroot	web.xxxxxx.dedyn.io	http://192.168.xxx.xxx/index.php

□ Relevante Dateien

docker-compose.yml

- Enthält:
 - traefik, lam, portainer, ldap, samba
- Traefik lauscht auf Port 80 & 443

- ACME-Zertifikatsspeicher: `/letsencrypt/acme.json`
- Mounts:
 - `dynamic.yml` als externe Konfiguration (`/etc/traefik/dynamic.yml`)
 - `traefik.yml` als statische Konfiguration (`/etc/traefik/traefik.yml`)

traefik.yml

snippet.yml

```
log:
  level: INFO

api:
  dashboard: true

entryPoints:
  web:
    address: ":80"
  websecure:
    address: ":443"

providers:
  docker:
    exposedByDefault: false
  file:
    filename: /etc/traefik/dynamic.yml
    watch: true

certificatesResolvers:
  le:
    acme:
      email: user@Email.de
      storage: /letsencrypt/acme.json
      httpChallenge:
        entryPoint: web
```

dynamic.yml

- Enthält:
 - Alle HTTP-Router (z. B. `pihole`, `dokuwiki`, `webroot`)
 - Middleware (`redirectRegex`) zur Umleitung auf `/admin`, `/dokuwiki/` usw.
 - ACME-HTTP-Router für `/.well-known/acme-challenge/`

Beispiel für Pi-hole:

[snippet.yaml](#)

```
pihole:
  rule: "Host(`pihole.xxxxxx.dedyn.io`)"
  entryPoints:
    - websecure
  tls:
    certResolver: le
  service: pihole-svc
  middlewares:
    - pihole-redirect
```

ACME-HTTP-Router:

[snippet.yaml](#)

```
acme-http:
  rule: "PathPrefix(`/well-known/acme-challenge/`)"
  entryPoints:
    - web
  service: noop
  priority: 100
```

Dummy-Service:

[snippet.yaml](#)

```
noop:
  loadBalancer:
    servers:
      - url: "http://127.0.0.1"
```

□ Zertifikate (Let's Encrypt)

- Wird über ACME + HTTP-Challenge angefordert
- Voraussetzung:
 - Port 80 öffentlich erreichbar
 - DNS zeigt auf WAN-IP
- Zertifikate landen in: `/letsencrypt/acme.json`

Traefik vergibt „**TRAEFIK DEFAULT CERT**“ nur, wenn: - Port 80 blockiert ist - keine Challenge erfolgreich - kein Zertifikat gecached

□ Apache-Backend-Server (192.168.178.89)

- Apache dient als Backend für:
 - phpMyAdmin
 - DokuWiki
 - Webroot (index.php)

Wichtig: - HTTPS auf Apache deaktiviert (a2dissite default-ssl.conf) - Nur HTTP (Port 80) aktiviert - Keine Redirects von HTTP → HTTPS im Apache

□ Probleme und Lösungen

Problem	Ursache	Lösung
-----	-----	-----
TRAEFIK DEFAULT CERT	Zertifikat nicht erfolgreich angefordert	ACME-Router eingebaut, HTTP erreichbar gemacht
404 bei /dokuwiki	Root-Pfad falsch	Middleware redirectRegex eingebaut
Browser zeigt „nicht sicher“	Browser-Cache / HSTS	chrome://net-internals/#hsts + löschen

□ Fazit

- Setup läuft stabil mit gültigen TLS-Zertifikaten
- Dienste sind über dedyn.io Subdomains erreichbar
- Docker + Traefik lösen dynamische Konfiguration mit `dynamic.yml`
- Apache wurde entlastet und liefert nur noch statische Inhalte über HTTP



Lars Weiß 27.05.2025 12:55

From:
<http://wiki.nctl.de/dokuwiki/> - □ Veni. Vidi. sudo rm -rf / vici.

Permanent link:
http://wiki.nctl.de/dokuwiki/doku.php?id=it-themen:allgemein:traefik_reverse_proxy_setup

Last update: **01.10.2025 17:22**



