zurück

Dokumentation: 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.mash4077.dedyn.io`	Docker-Port 9000
Traefik-Dashboard	`traefik.mash4077.dedyn.io`	Interner Dienst (`api@internal`)
Pi-hole	`pihole.mash4077.dedyn.io`	`http://192.168.178.10/admin`
DokuWiki	`wiki.mash4077.dedyn.io`	`http://192.168.178.89/dokuwiki/`
phpMyAdmin	`phpmyadmin.mash4077.dedyn.io`	`http://192.168.178.89/phpmyadmin`
Webroot	`web.mash4077.dedyn.io`	`http://192.168.178.89/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.yaml

```yaml
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: lars.weiss@gmail.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

```yaml
pihole:
  rule: "Host(`pihole.mash4077.dedyn.io`)"
  entryPoints:
    - websecure
  tls:
    certResolver: le
  service: pihole-svc
  middlewares:
    - pihole-redirect
```

ACME-HTTP-Router:

snippet.yaml

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

Dummy-Service:

snippet.yaml

```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