

# Complete setup

The following shows a complete setup for traefik with automatic ssl certificates, automatic http redirects to https, ipwhitelisting and custom error messages

## File structure

```
├─ config
|   ├── acme.json
|   ├── dynamic_config.yml
|   ├── error_handling
|   |   ├── default.conf
|   |   └─ error_pages
|   |       ├── 403.html
|   |       ├── 404.html
|   |       └─ ...
|   └─ traefik.yml
├─ docker-compose.yml
└─ logs
    ├── access.log
    └─ error.log
```

### docker-compose.yml

```
version: '3'

services:
  traefik:
    image: traefik
    container_name: traefik
    depends_on:
      - error_pages
    ports:
      - "80:80"
      - "443:443"
      - "8080:8080"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
```

- ./config/traefik.yml:/etc/traefik/traefik.yml
- ./config/dynamic\_config.yml:/etc/traefik/dynamic\_config.yml
- ./config/acme.json:/etc/traefik/acme.json

environment:

- NETCUP\_CUSTOMER\_NUMBER=\${NETCUP\_CUSTOMER\_NUMBER}
- NETCUP\_API\_KEY=\${NETCUP\_API\_KEY}
- NETCUP\_API\_PASSWORD=\${NETCUP\_API\_PASSWORD}

labels:

```
traefik.enable: "true"
traefik.http.routers.traefik.service: 'api@internal'
traefik.http.routers.traefik.middlewares: "secured@file"
traefik.http.routers.traefik.entrypoints: "websecure"
traefik.http.routers.traefik.rule: "Host(`traefik.${SITE}`)"
```

error\_pages:

image: nginx:latest

container\_name: error\_pages

volumes:

- ./config/error\_handling/error\_pages:/usr/share/nginx/error\_pages
- ./config/error\_handling/default.conf:/etc/nginx/conf.d/default.conf
- ./logs:/var/log/nginx

labels:

traefik.enable: true

```
traefik.http.routers.error-router.rule: HostRegexp(`{host:.+}`)
traefik.http.routers.error-router.priority: 1
traefik.http.routers.error-router.entrypoints: websecure
traefik.http.routers.error-router.middlewares: error-pages-middleware
traefik.http.middlewares.error-pages-middleware.errors.status: 400-599
traefik.http.middlewares.error-pages-middleware.errors.service: error-pages-service
traefik.http.middlewares.error-pages-middleware.errors.query: /{status}.html
traefik.http.services.error-pages-service.loadbalancer.server.port: 80
```

## config/traefic.yml

global:

checkNewVersion: true

sendAnonymousUsage: false

entryPoints:

```
web:
  address: :80
  http:
    redirections:
      entrypoint:
        to: websecure
      scheme: https
websecure:
  address: :443
  http:
    middlewares:
      - error-pages-middleware@docker
  tls:
    certResolver: netcup

certificatesResolvers:
  netcup:
    acme:
      email: <email@example.com>
      storage: /etc/traefik/acme.json
      # caServer: https://acme-staging-v02.api.letsencrypt.org/directory
    dnsChallenge:
      provider: netcup
      delayBeforeCheck: 900
      resolvers:
        - "root-dns.netcup.net:53"
        - "second-dns.netcup.net:53"

api:
  dashboard: true

providers:
  docker:
    endpoint: "unix:///var/run/docker.sock"
    exposedByDefault: false
  file:
    filename: /etc/traefik/dynamic_config.yml
    watch: true
```

config/dynamic\_config.yml

```
http:
  middlewares:
    localonly:
      ipWhiteList:
        sourceRange:
          - "192.168.0.0/24"
          - "127.0.0.1/32"
      secured:
        chain:
          middlewares:
            - error-pages-middleware@docker
            - localonly
  tls:
    stores:
      default:
        defaultGeneratedCert:
          resolver: netcup
          domain:
            # main: "example.com"
            sans: "/*.example.com"
```

### config/error\_handling/default.conf

```
server {
    listen    80;
    server_name localhost;

    error_page 403 /403.html;
    error_page 404 /404.html;
    # error_page <code> /<code>.html;
    location /{
        internal;
        root /usr/share/nginx/error_pages;
    }
}
```

### config/error\_handling/error\_pages/xxx.html

You can create your own custom error pages or insert predefined ones.

Some examples: <https://github.com/tarampampam/error-pages>

## Things to keep in mind

- Set your email and domain name(s) accordingly.
- Change the provider settings for the DNS challenge according to your provider [here](#).
- Make sure you use the staging caServer of letsencrypt first and then switch back to the normal one, when everything works as intended.

---

Revision #2

Created 14 September 2023 17:31:45 by Levin

Updated 14 September 2023 17:51:13 by Levin