

Traefik

Traefik used for ingress and cert management

```
traefik:
  image: traefik:v3
  restart: always
  container_name: traefik
  ports:
    - "80:80" # <== http
    #- "8080:8080" # <== :8080 is where the dashboard runs on
    - "443:443" # <== https
  command:
    ##### These are the CLI commands that will configure Traefik and tell it how to work!
    #####
    ## API Settings - https://docs.traefik.io/operations/api/, endpoints -
    https://docs.traefik.io/operations/api/#endpoints ##
    - --api=true # <== Enabling insecure api, NOT RECOMMENDED FOR PRODUCTION
    - --api.dashboard=true # <== Enabling the dashboard to view services, middlewares,
    routers, etc...
    - --api.debug=true # <== Enabling additional endpoints for debugging and profiling
    - --servertransport.insecureskipverify=true
    - --log.level=INFO
    #- --log.filepath=/var/logs/traefik.log
    - --accesslog=true
    #- --accesslog.filepath=/var/logs/traefik-access.log

    ## Log Settings (options: ERROR, DEBUG, PANIC, FATAL, WARN, INFO) -
    https://docs.traefik.io/observability/logs/ ##
    #- --log.level=DEBUG # <== Setting the level of the logs from traefik

    ## Provider Settings - https://docs.traefik.io/providers/docker/#provider-configuration
    ##
    - --providers.docker=true # <== Enabling docker as the provider for traefik
    - --providers.docker.exposedbydefault=false # <== Don't expose every container to
    traefik, only expose enabled ones
```

```

- --providers.file.filename=/dynamic.yml # <== Referring to a dynamic configuration file
- --providers.file.directory=/rules
- --providers.docker.network=web # <== Operate on the docker network named web

## Entrypoints Settings - https://docs.traefik.io/routing/entrypoints/#configuration ##
- --entrypoints.web.address=:80 # <== Defining an entrypoint for port :80 named web
- --entrypoints.web-secured.address=:443 # <== Defining an entrypoint for https on port
:443 named web-secured
- --entrypoints.web.http.redirects.entrypoint.to=web-secured
- --entrypoints.web.http.redirects.entrypoint.scheme=https
- --entrypoints.web-secured.asDefault=true
- --entrypoints.web-secured.http.tls.certResolver=mytlschallenge

## Certificate Settings (Let's Encrypt) -
https://docs.traefik.io/https/acme/#configuration-examples ##
- --certificatesResolvers.mytlschallenge.acme.httpChallenge.entryPoint=web
- --certificatesResolvers.mytlschallenge.acme.tlschallenge=true # <== Enable TLS-ALPN-01
to generate and renew ACME certs
- --certificatesResolvers.mytlschallenge.acme.email=kevin@kevinsloan.net # <== Setting
email for certs
- --certificatesResolvers.mytlschallenge.acme.storage=/letsencrypt/acme.json # <==
Defining acme file to store cert information

volumes:
- ./letsencrypt:/letsencrypt # <== Volume for certs (TLS)
- /var/run/docker.sock:/var/run/docker.sock # <== Volume for docker admin
- ./dynamic.yml:/dynamic.yml # <== Volume for dynamic conf file, **ref: line 27
- ./rules:/rules
- /mnt/user/Share/Docker/Traefik/logs:/var/logs

networks:
## Placing traefik on the network named web, to access containers on this network
web:
  ipv4_address: 172.18.0.2

labels:
#### Labels define the behavior and rules of the traefik proxy for this container ####
- traefik.enable=true # <== Enable traefik on itself to view dashboard and assign
subdomain to view it
- traefik.http.routers.traefik-web.rule=Host(`traefik.kevinsloan.net`) # <== Setting the

```

domain for the dashboard

- traefik.http.routers.traefik-web.entrypoints=web
- traefik.http.routers.traefik-secured.rule=Host(`traefik.kevinsloan.net`)
- traefik.http.routers.traefik-secured.entrypoints=web-secured
- traefik.http.routers.traefik-secured.service=api@internal # <== Enabling the api to be a service to access

rules directory to setup custom entrypoints

aoo-hassio.toml

```
[http.routers]
[http.routers.hassio-rtr]
  entryPoints = ["web-secured"]
  rule = "Host(`homeassist.kevinsloan.net`)"
  service = "hassio-svc"
[http.routers.hassio-rtr.tls]
  certresolver = "mytlschallenge"

[http.services]
[http.services.hassio-svc]
[http.services.hassio-svc.loadBalancer]
  passHostHeader = true
[[http.services.hassio-svc.loadBalancer.servers]]
  url = "http://192.168.123.108:8123" # or whatever your external host's IP:port is
```

app-kuma.toml

```
[http.routers]
[http.routers.kuma-rtr]
  entryPoints = ["web-secured"]
  rule = "Host(`kuma-uptime.kevinsloan.net`)"
  service = "kuma-svc"
[http.routers.kuma-rtr.tls]
  certresolver = "mytlschallenge"

[http.services]
[http.services.kuma-svc]
[http.services.kuma-svc.loadBalancer]
```

```
[[http.services.kuma-svc.loadBalancer.servers]]
```

```
url = "http://192.168.123.101:80" # or whatever your external host's IP:port is
```

app-pihole.toml

```
[http.routers]
```

```
  [http.routers.pihole-rtr]
```

```
    entryPoints = ["web-secured"]
```

```
    rule = "Host(`pihole.kevinsloan.net`)"
```

```
    service = "pihole-svc"
```

```
  [http.routers.pihole-rtr.tls]
```

```
    certresolver = "mytlschallenge"
```

```
[http.services]
```

```
  [http.services.pihole-svc]
```

```
    [http.services.pihole-svc.loadBalancer]
```

```
      [[http.services.pihole-svc.loadBalancer.servers]]
```

```
        url = "http://192.168.123.107:80" # or whatever your external host's IP:port is
```

Revision #3

Created 29 May 2020 01:46:07 by sloan

Updated 22 February 2025 23:49:51 by sloan