

Dienst konfigurieren

Um nun den Dienst eines anderen Containers anzubieten, wird mit den Docker Labeln gearbeitet. Diese liest Traefik automatisch laufend aus und stellt die Dienste bereit, so wie sie in den Labeln beschrieben werden. Es folgt ein Beispiel-Dienst von Traefik selbst: whoami. Dieser zeigt lediglich einige Serverinformationen an, jedoch eignet er sich hervorragend dazu, um die automatische Konfiguration zu demonstrieren.

Zunächst erstellen wir eine neue Docker Compose Konfiguration mit folgendem Inhalt.

```
services:
  whoami:
    image: "traefik/whoami"
    container_name: "simple-service"
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.whoami.rule=Host(`whoami.localhost`)"
      - "traefik.http.services.whoami.loadbalancer.server.port=80"
    networks:
      - traefik

networks:
  traefik:
    name: traefik
    external: true
```

Sofern der Dienst für HTTPS anstatt HTTP konfiguriert wird, sind folgende Label zusätzlich hinzuzufügen:

traefik.http.routers.whoami.tls=true

traefik.http.routers.whoami.tls.certresolver=einzelzertifikat

Der Certresolver ist auf den Namen des Resolvers zu ändern, wie dieser in der Traefik Konfiguration angelegt wurde.

Hier sehen wir nun 3 Label für die Traefik Konfiguration.

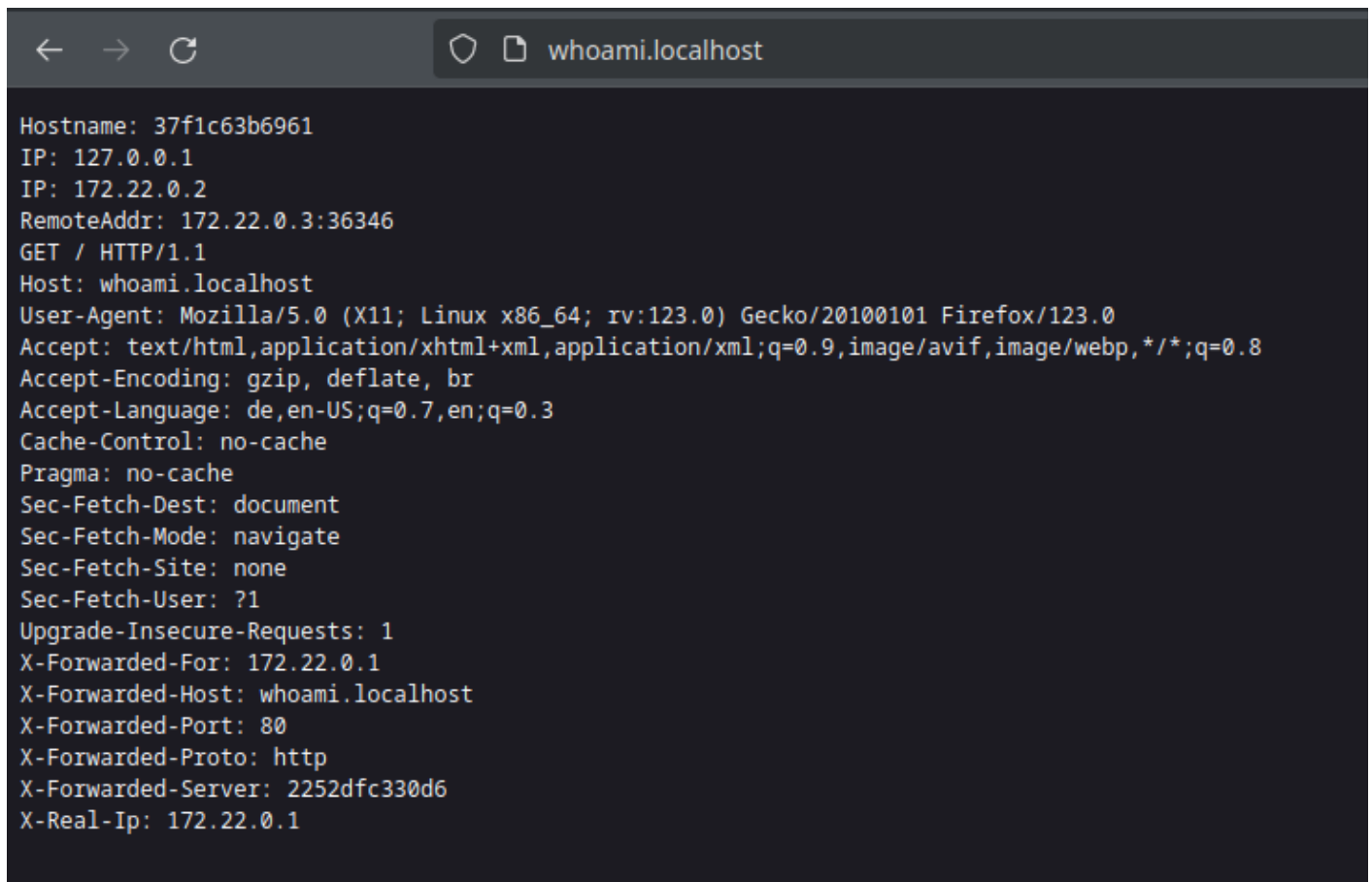
1. Enable: damit wird dieser Container für Traefik markiert und von diesem verwaltet
2. Rule: hier wird festgelegt, welche Regel zum Aufruf des Dienstes führt, in diesem Fall durch den Aufruf über die Domäne whoami.localhost
3. Port: Gibt den Port des Dienstes an, an welchen Traefik die Anfragen weiterleitet, sofern der Dienst nur einen Port bereitstellt, kann diese Option weggelassen werden.

Bitte bei 2. auf die Anführungszeichen achten, es sind nicht die geraden einfachen, sondern die schrägen einzelnen Striche, die wie ein Backslash verlaufen.

Damit Traefik und Whoami miteinander kommunizieren können, wird Whoami in das Netzwerk aufgenommen, zu dem bereits Traefik hinzugefügt wurde. In diesem Beispiel heißt das Netzwerk *traefik*, wie bereits in der Installationsanleitung.

Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach gewählter Installation `sudo docker-compose up -d` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegebenen Optionen. Sollten die Container bereits mit dieser Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

Anschließend kann mit <http://whoami.localhost> der neue Dienst über den Reverse Proxy getestet werden.



```
← → ↻ whoami.localhost

Hostname: 37f1c63b6961
IP: 127.0.0.1
IP: 172.22.0.2
RemoteAddr: 172.22.0.3:36346
GET / HTTP/1.1
Host: whoami.localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:123.0) Gecko/20100101 Firefox/123.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: de,en-US;q=0.7,en;q=0.3
Cache-Control: no-cache
Pragma: no-cache
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 172.22.0.1
X-Forwarded-Host: whoami.localhost
X-Forwarded-Port: 80
X-Forwarded-Proto: http
X-Forwarded-Server: 2252dfc330d6
X-Real-IP: 172.22.0.1
```

Besonders angenehm: Traefik muss nicht neu gestartet werden, sondern sobald der neue Container gestartet ist, wird dieser automatisch von Traefik erkannt und dieser konfiguriert sich selbst. Die Seite ist kurz danach bereits über Traefik erreichbar.

