

# Nginxproxymanager Setup

Vor kurzem musste ich mein geliebtes und bis dato immer funktionierendes Proxy Setup zwischen meinem apache2 Host und meiner Docker Instanz auflösen. Grund hierfür war das ich keine einfache Möglichkeit hatte mir nach einem Neustart die neuen IP Adressen der jeweiligen Docker Container zu holen um dann diese Daten in meine Vhost Configen eintragen zu lassen. Es war also nach jedem Neustart der Container nötig die IP Adressen der jeweiligen Subdomains für die Container einzeln anzupassen. --> Geht gar nicht .. -.-

Ergo musste etwas neues, effizienteres her und was gibt es nicht schöneres als Networking ... \*würg\* ... ok also ran an die Bullethen .. ich wills ins Bett :D

.....

Vor kurzem musste ich mein geliebtes und bis dato immer funktionierendes Proxy Setup zwischen meinem apache2 Host und meiner Docker Instanz auflösen. Grund hierfür war das ich keine einfache Möglichkeit hatte mir nach einem Neustart die neuen IP Adressen der jeweiligen Docker Container zu holen um dann diese Daten in meine Vhost Configen eintragen zu lassen. Es war also nach jedem Neustart der Container nötig die IP Adressen der jeweiligen Subdomains für die Container einzeln anzupassen. --> Geht gar nicht .. -.-

Ergo musste etwas neues, effizienteres her und was gibt es nicht schöneres als Networking ... \*würg\* ... ok also ran an die Bullethen .. ich wills ins Bett :D

Wir loggen uns per SSH ein und erstellen mit dem Benutzer root (mache ich immer so) im Home Verzeichnis ein neues Verzeichnis für die Daten unseres neuen Containers. Anschließend wechseln wir in unser neues Verzeichnis und erstellen eine neue **docker compose YML**.

Code

```
cd /home && mkdir nginxproxy && cd nginxproxy && touch docker-compose.yml
```

Folgender Inhalt müssen wir in die neue Datei übertragen:

Code

```
nano docker-compose.yml
```

Code

```
version:                                                                                                     '3.8'
services:
  app:

  ports:
    - - '80:80' #
    - - '443:443' #
    - - '81:81'

  volumes:
    - ./data:/data # <-- findet ihr nach dem Setup unter /home/nginxproxy
    - ./letsencrypt:/etc/letsencrypt # <-- findet ihr nach dem Setup unter /home/nginxproxy
```

Alles anzeigen

Also .. wir erstellen einen neuen Container mit dem Image `jc21/nginx-proxy-manager:latest`. Dieser Container wird später die Verwaltung des Proxys übernehmen. Da ich an meinem Server mehr wie nur eine

IP Adresse habe, war es mir möglich für diesen speziellen Container eine andere IP Adresse zu nutzen und ich haben den jeweiligen Port an die neue IP Adresse gebunden. Wichtig ist hierbei das ihr FALLS ihr schon einen Webserver auf dem Server laufen habt ihr diesen ebenfalls an eine IP Adresse bindet.

**Merke: Das hier geht nur wenn du A) noch keinen Webserver am laufen hast oder B) du eine 2te IP Adresse hast.**

IP Adressen binden könnt ihr den Apache2 zb in der Datei `/etc/apache2/ports.conf`. Bitte erkundigt euch diesbezüglich nochmal selber.

Mit einem `docker-compose up -d` startet ihr nun den neuen Container im Hintergrund. Magieeeeeee!  
whistling

Wenn alle ready ist loggen wir uns als erstes im Nginx Proxymanager ein mit folgenden Daten: (Diese sind von dem Developer so vorgeben und haben nichts mit einer Grausamkeit meinerseits zu tun <sup>8)</sup> bitte unbedingt ändern)

Code

Email:  
Password: changeme

Bitte bedenkt bevor ihr die jeweiligen Subdomains hier im Manager eintragen wollt das ihr noch unbedingt die DNS Einträge kontrollieren müsst. Eventuelle IP Änderungen solltet ihr hier schon berücksichtigt haben! Bevor wir jetzt einen Host anlegen könnt müssen wir noch für docker ein spezielles, eigenes Netzwerk anlegen und **ALLE** Container, die über den Proxymanager kommunizieren wollen hinzufügen.

Also wieder auf die Shell (*Mal nebenbei: **Hast du gewusst:** Es gibt Millionen Codezeilen in der [Gemeinschaftsprojekte der Linux Foundation](#). Zum Vergleich: Windows XP umfasst 45 Millionen Codezeilen.*):

Code

```
docker network create <networkname>
```

STOP: Wir erinnern uns an unsere `docker-compose.yml` welche wir anfangs erstellt haben .. ? Wir öffnen diese und fügen ändern die Datei auf folgendes ab:

Code

```

version: '3.8'
services:
  app:

  ports:
    - '80:80'
    - '443:443'
    - '81:81'

  volumes:

  networks:
    healthcheck: # Kann man .. muss man nicht ..

```

```

networks:
  <networkname>:
    external: true

```

### Alles anzeigen

Was haben wir jetzt geändert? Wir haben unseren Proxymanager in unser neues Netzwerk verbunden und damit quasi den Grundstein dafür geschaffen das der Proxymanager Container mit den anderen noch kommenden Containern kommunizieren kann. Ich gehe einfach mal davon aus das es da bei euch noch mehr Container gibt die ihr verbinden wollt ..

Dafür würde dann der Befehl

### Code

```
docker network connect <networkname> --alias nginx nginx
```

ausreichen. Hierbei fügt ihr den bereits vorhandenen Container `nginx` mit dem neuen Alias (Hostname des Containers) `nginx` in euer neues Netzwerk ein.

Legt jetzt einfach eure einzelnen ProxyHosts im Proxymanager an und stellt eure Einstellungen bezüglich SSL ein.

Ja .. ich denke das waren genau die Schritte die ich gemacht habe .. Also ich wünsche euch viel Erfolg und würde mich freuen wenn ihr bei Fragen oder Problemen einen Beitrag bzw ein Thema hier eröffnet ;)

↳ [page not found or type unknown](#)