

# #Sonstiges | NGINX Proxy Manager Installation und Einrichtung

## Was wollen wir?

NGINX Proxymanager installieren, um interne Dienste nach außen per FQDN erreichbar zu machen.

## Warum wollen wir das?

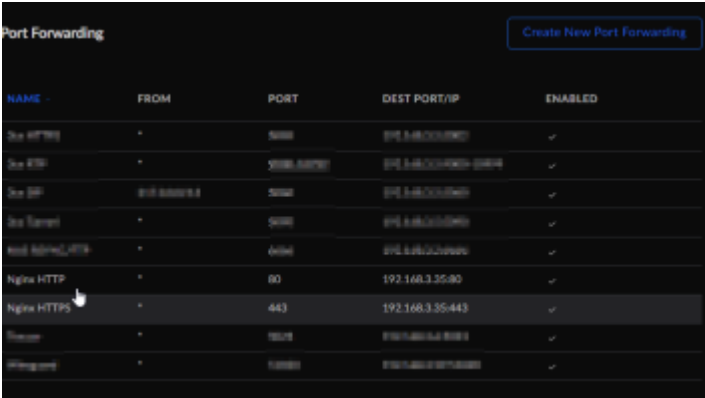
Um die gewünschten Dienste möglichst sicher mit SSL Zertifikat zu veröffentlichen und nur wenige Ports freigeben zu müssen.

## Und wie geht das genau?

Voraussetzung ist ein vorhandener Server/PC mit z.B. VMWare ESXi oder Proxmox, um eine VM zu installieren oder ein NAS mit Docker, sowie das Weiterleiten der Port 80/tcp und 443/tcp auf das entsprechende Zielsystem.

### Firewall konfigurieren:

Port 80 und 443 auf die IP des Zielsystems weiterleiten, die wir weiter unten installieren:



| NAME        | FROM          | PORT       | DEST PORT/IP        | ENABLED |
|-------------|---------------|------------|---------------------|---------|
| Dns HTTP    | *             | 8080       | 192.168.1.100:80    | ✓       |
| Dns HTTPS   | *             | 8080/HTTPS | 192.168.1.100:443   | ✓       |
| Dns SSH     | 192.168.1.100 | 8080       | 192.168.1.100:22    | ✓       |
| Dns Samba   | *             | 9000       | 192.168.1.100:139   | ✓       |
| Web Remote  | *             | 4000       | 192.168.1.100:8080  | ✓       |
| Nginx HTTP  | *             | 80         | 192.168.1.35:80     | ✓       |
| Nginx HTTPS | *             | 443        | 192.168.1.35:443    | ✓       |
| Proxmox     | *             | 8008       | 192.168.1.100:8008  | ✓       |
| Proxmox/SSH | *             | 22000      | 192.168.1.100:22000 | ✓       |

### Ubuntu installieren und Updates einspielen:

Die Grundinstallation von Ubuntu in einer VM überspringe ich an dieser Stelle mal.

Code

```
sudo apt-get update  
sudo apt-get upgrade
```

### Docker installieren:

Code

```
curl -fsSL https://get.docker.com | bash
```

### Docker Compose installieren:

Code

```
ps -L /git@ubuntu.com/docker/compose/releases/download/1.28.5/docker-compose -f $(cat /dev/null) > /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose
```

### Portainer installieren:

Code

```
sudo docker volume create portainer_data  
sudo docker run -d -p 8000:8000 -p 9000:9000 --name=portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock
```

### Nginx Proxy Manager installieren:

Portainer über <http://IP:9000> aufrufen und einen Account anlegen.

In Portainer ein Stack anlegen mit der config:

Code

```

version:
services:
  nginx-proxy-manager:

    ports:

    depends_on:

    environment:

    volumes:

db:

  environment:

  volumes:
    - ./data/mysql:/var/lib/mysql

```

Alles anzeigen

**Deploy Stack** anklicken und abwarten bis alles heruntergeladen und installiert wurde.

Danach starten wir die Docker Container *NGINX Proxy Manager* und die *Datenbank* für den Proxy Manager.

Nach dem Start der Container noch jeweils einmal drauf klicken und runter scrollen und beim Punkt "Restart Policies" noch auf das gewünschte Startverhalten nach Neustart des Servers einstellen.

Dann kann man über <http://IP:81> den Proxy Manager aufrufen und sich mit "admin@example.com" und dem Kennwort "changeme" einloggen.

**Nachdem man seine Login Daten angepasst hat kann man einen proxy Host einrichten:**

Domain Name eingeben (eine Domain die auf eure DynDNS Adresse oder feste IP zeigt)\*.

Scheme: http.

Forward Hostname/IP: IP des Servers/VM/Docker Containers/NAS.

Forward Port: Port des Dienstes den man veröffentlichen will.

Haken bei "Block Common Exploits" und "Websockets Support" setzen.

Unter SSL klickt man "Request new SSL certificate" an und setzt die Haken bei "Force SSL" und "HTTP/2 Support".

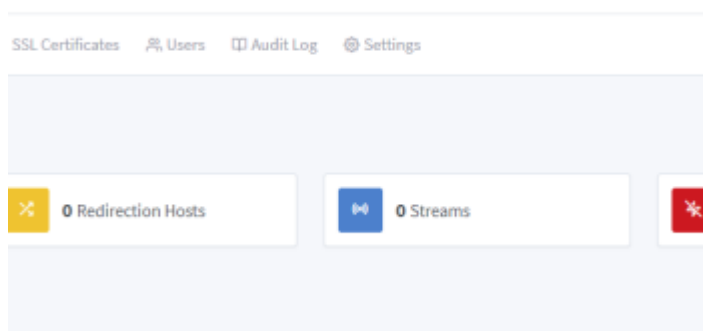
Unter *Advanced* kann man je nach Dienst noch Code eingeben.

Dann "Safe" anklicken und kurz warten bis das SSL Zertifikat erfolgreich angefordert wurde.

Noch einmal den Eintrag öffnen und prüfen, ob die Einstellungen bei SSL beibehalten wurden.

Nun kann man über <https://DEINEDOMAIN.de> den Dienst aufrufen

Hier im Beispiel mein Synology-NAS, welches intern unter <https://192.168.3.2:5001> erreichbar ist:



Forward Hostname / IP \*

Scheme \*

Forward Port \*

Cache Assets

Block Common Exploits

Websockets Support

Access List

Details Custom locations SSL Advanced

SSL Certificate

Force SSL

HTTP/2 Support

HSTS enabled

HSTS Subdomains

192.168.3.2

https

5001

Let's Encrypt

\* Ich habe das mit meiner Domain folgendermaßen gelöst:

Dyndns läuft auf meiner UDR

Domain <http://www.meinedomain.de> ist eingerichtet bei meinem Anbieter.

In den DNS Einstellungen lege ich mir dann einen CNAME an, der auf meine Dyndns Adresse zeigt z.B. meinnas.meinedomain.de

Im Proxymanager lege ich nun den Host Eintrag mit meinnas.meinedomain.de an und lasse das Zertifikat darauf ausstellen

Nun kann ich, wenn ich die Adresse meinnas.meinedomain.de aufrufe, die Weboberfläche meiner Synology erreichen, ohne extra einen Port eingeben zu müssen oder ähnliches und das Ganze ist auch noch mit einem Zertifikat versehen, welches sich selbst (über den Proxymanager) aktualisiert.

Genau so verfähre ich dann mit jedem weiteren Dienst, den ich von extern erreichen möchte.

Der Vorteil: ich habe nur Port 80 und 443 auf meinen Proxymanager weitergeleitet und keiner meiner Dienste ist von extern direkt erreichbar.

Viel Spass mit NGINX Proxy Manager 😎

Disclaimer: Alle Anleitungen/Tutorials sind nach bestem Wissen und Gewissen verfasst, gehen immer von den definierten Software/Firmware-Versionen aus und sind auf das englische GUI ausgelegt.

Es gibt keine Garantien auf Erfolg. Im Falle eines Misserfolges hilft aber sicherlich die Community hier immer weiter.

Keiner der Autoren oder der Betreiber des Forums ist für die aus der Nutzung resultierenden Probleme/Herausforderungen verantwortlich.

Jegliche hier beschriebenen Schritte erfolgen ausnahmslos in eigener Verantwortung des Durchführenden. Eltern haften für ihre Kinder.



Auswählen: \_\_\_\_\_

Gültige Software-Version Keine Firmware-Relevanz!