

# Deployment einer Webanwendung

---

Von Armin Spöllmann und Jannik Möbius

[https://github.com/ItZzMJ/ToDoList\\_OpenAPI](https://github.com/ItZzMJ/ToDoList_OpenAPI)

## Inhaltsverzeichnis

- [Netzwerkconfiguration](#)
  - [Statische IP Festlegen](#)
- [Benutzer anlegen und konfigurieren](#)
- [Firewall konfiguration](#)
- [ToDo-Listen-Verwaltung und Nextcloud deployen](#)
  - [Autostart](#)
  - [Mit Docker](#)

## Vorgehen

### Netzwerkconfiguration

#### Statische IP Festlegen

Zuerst die Datei `/etc/dhcpd.conf` wie folgt editieren um eine statische IP zu erhalten

```
#static IP configuration LAN
interface eth0
static ip_address=192.168.24.134/24
static routers=192.168.24.254
static domain_name_servers=192.168.24.254

#static IP configuration WLAN
interface wlan0
static ip_address=192.168.24.164/24
static routers=192.168.24.254
static domain_name_servers=192.168.24.254
```

Anschließend den Daemon mit neustarten

```
sudo systemctl restart dhcpd
```

Für WLAN die Datei `/etc/wpa_supplicant/wpa_supplicant.conf` editieren

```
ctrl_interface=DIR=/var/run/wpa_supplicant
GROUP=netdev
country=de
```

```
update_config=1
network={
    ssid="<SSID>"
    psk="<PASSWORD>"
}
```

Anschließend den Service neustarten

```
sudo systemctl restart dhcpcd
```

Verfügbare Netzwerke anzeigen

```
wpa_cli -i wlan0 list_networks
```

Falls kein Netzwerk angezeigt wird, Netzwerk einrichten In das CLI gehen mit wpa\_cli

```
scan
scan_results
add_network
add_network 0 ssid "SSID"
add_network 0 psk "passphrase"
enable_network 0
save_config
quit
```

Nun kan man sich im WLAN anmelden

```
wpa_cli -i wlan0 select_network 0
```

Benutzer anlegen und konfigurieren

```
sudo adduser benutzer72
sudo adduser fernzugriff
```

Fernzugriff soll sudo Rechte bekommen also wird er zur sudo group hinzugefügt

```
sudo adduser fernzugriff sudo
```

Firewall konfiguration

Zuerst Firewall installieren

```
sudo apt-get install ufw
```

Firewall Regeln festlegen

```
sudo ufw allow 22
sudo ufw allow from 192.168.24.0/16 to any port 22
sudo ufw allow 80
sudo ufw allow 443
```

Firewall aktivieren

```
sudo ufw enable
```

ToDo-Listen-Verwaltung und Nextcloud deployen

Git, Docker und Docker Compose installieren

```
sudo apt install git docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Set up docker as non-root user

```
sudo groupadd docker
sudo usermod -aG docker pi
```

Code von git pullen

```
git clone https://github.com/ItZzMJ/ToDoList_OpenAPI.git
cd ToDoList_OpenAPI
```

App starten

```
python main.py
```

Nun ist die API über die IP 192.168.24.164 erreichbar

## Autostart

Für App autostart supervisor installieren:

```
sudo apt install supervisor
```

Eine Konfigurationsdatei erstellen

```
sudo nano /etc/supervisor/conf.d/todolist.conf
```

Content der Configdatei:

```
[program:flask_app]
command = python main.py
directory = /home/pi/code/ToDoList_OpenAPI/
autostart = true
autorestart = true
```

Supervisor aktualisieren und starten

```
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start flask_app
```

## Mit Docker

Die Datei Dockerfile erstellen und folgendes einfügen

```
# syntax=docker/dockerfile:1
FROM python:3.7-alpine
WORKDIR /code
RUN apk add --no-cache gcc musl-dev linux-headers nano bash
COPY ./requirements.txt requirements.txt
RUN pip install -r requirements.txt
EXPOSE 80
COPY . /data
CMD ["python", "/data/main.py"]
```

Die Datei docker-compose.yml erstellen und folgendes einfügen

```
version: '3'
services:
  flask:
    build: .
    ports:
      - "80:80"
      - "8000:5000"
    restart: unless-stopped

  nextcloud:
    image: nextcloud:latest
    ports:
      - "8080:80"
    volumes:
      - ./nextcloud:/var/www/html
    restart: unless-stopped
```

Container builden

```
docker-compose build
```

Container starten

```
docker-compose up -d
```