

UniFi Network Application in a docker-compose App

this turned out to be more complicated than I had anticipated. the reason for no longer running the unifi network application directly on my PC's or also server's installation and rather put it in a docker container was the somewhat cumbersome debian package Ubiquiti Networks provides for that. The problem there is mainly the dependency on MongoDB version 3, which is very old. For Ubuntu 22.04 there was a mongodb repo which could be installed as per UBN's instructions and which would then install MongoDB3 even on a modern Linux, but unfortunately those packages have then a dependency for an old libssl which is no longer part of ubuntu 24.04 so that's another obsolete dependency to track down to get the thing to work. And frankly I don't want to go down that rabbit hole and cause possible issues with other packages in the future.

So the decision was made to solve this with docker. I am going to use the linuxserver.io docker container, as those are usually well maintained. Unfortunately they have removed the MongoDB server from the docker image (as per docker's one service per container philosophy, but less convenient for us to use ;)), so we need to start a mongodb container as well and most importantly, we need to initialize it to create the db for unifi to use.

There is an example docker-compose.yml file available on the linuxserver.io page, but I found the instructions on initializing the mongoDB container a bit unclear.

I finally went with mongodb version 7 which seems to run fine, and i went with the javascript init script, but I modified it to use environment variables rather than hardcoded values for username and passwords etc. so one only has to adjust the docker-compose.yml or .env file between installations.

So here we go: this is my docker-compose file:

[docker-compose.yml](#)

```
---
# IMPORTANT: in the unifi controller web gui, go to
# settings -> system -> advanced and click the "Override" checkbox for
# "inform host"
# enter the external ip of your host or else adoption will fail
services:
  unifi-network-application:
    image: lscr.io/linuxserver/unifi-network-application:latest
    container_name: unifi-network-application
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Zurich
      - MONGO_USER=unifi
      - MONGO_PASS=ieXi6teenohxuob
      - MONGO_HOST=unifi-db
      - MONGO_PORT=27017
      - MONGO_DBNAME=unifi
      - MEM_LIMIT=1024 #optional
```

```
# - MEM_STARTUP=1024 #optional
# - MONGO_TLS= #optional
# - MONGO_AUTHSOURCE= #optional
volumes:
  - ./config:/config
ports:
  - 8443:8443
  - 3478:3478/udp
  - 10001:10001/udp
  - 8080:8080
# - 1900:1900/udp #optional
# - 8843:8843 #optional
# - 8880:8880 #optional
# - 6789:6789 #optional
# - 5514:5514/udp #optional
restart: unless-stopped
unifi-db:
  image: docker.io/mongo:7.0
  container_name: unifi-db
  volumes:
    - ./data:/data/db
    - ./init/init-mongo.js:/docker-entrypoint-initdb.d/init-
mongo.js:ro
  restart: unless-stopped
  environment:
    - PUID=1000
    - PGID=1000
    - TZ=Europe/Zurich
    - MONGO_USER=unifi
    - MONGO_PASS=ieXi6teenohxuob
    - MONGO_DBNAME=unifi
```

change the password to an individual one for your installation! and this is the javascript file, that goes in an init sub-folder:

[init/init-mongo.js](#)

```
db.getSiblingDB(process.env.MONGO_DBNAME).createUser({user:
process.env.MONGO_USER, pwd: process.env.MONGO_PASS, roles: [{role:
"dbOwner", db: process.env.MONGO_DBNAME}]});
db.getSiblingDB(process.env.MONGO_DBNAME+"_stat").createUser({user:
process.env.MONGO_USER, pwd: process.env.MONGO_PASS, roles: [{role:
"dbOwner", db: process.env.MONGO_DBNAME+"_stat"}]});
```

now simply run

```
docker compose up -d
```

and connect with a web-browser to port 8443 or 8080 to get to the web gui.

IMPORTANT you have to then go to settings → system → advanced and click the Override checkbox for inform host, now enter the IP address or hostname of your Host computer, as otherwise the unifi network application will always publish its docker internal IP address to the unifi equipment as ifnorm-host, which basically means that the equipment won't be successfully adopted.

From:

<http://wiki.psuter.ch/> - **pswiki**

Permanent link:

http://wiki.psuter.ch/doku.php?id=unifi_network_application_in_a_docker-compose_app

Last update: **24.06.2024 10:54**

