# UniFi Network Application

docker-compose.yaml

```
services:
  # Define the services to run
  db:
    image: mongo
    volumes:
      - ./db:/data/db # MongoDB data persistence
      - ./db/init-mongo.js:/docker-entrypoint-initdb.d/init-mongo.js:ro
    ports:
      - 27017:27017 # MongoDB default port
    restart: always # Restart policy
    networks:
      backend: null
  app:
    image: lscr.io/linuxserver/unifi-network-application # Docker image to use
    environment:
      # Environmental variables for the container
      - PUID=1000 # User ID
      - PGID=1000 # Group ID
      - TZ=Europe/Zurich # Timezone
      - MONGO_HOST=db # MongoDB host
      - MONGO_USER= ${DB_USER} # MongoDB username
      - MONGO_PASS= ${DB_PW} # MongoDB password
      - MONGO_PORT=27017 # MongoDB port
      - MONGO_DBNAME= ${DB_NAME} # MongoDB database name
      - MEM_LIMIT=1024 #optional  # Memory limit for the container
      - MEM_STARTUP=1024 #optional  # Memory to allocate on container startup
    volumes:
      # Volumes to mount in the container
      - ./web:/config # Map host directory to container directory
    ports:
      # Ports to expose and forward
      - 8443:8443 # HTTPS portal
      - 3478:3478/udp # STUN service
```

```
      - 10001:10001/udp # UniFi AP discovery
      - 8080:8080 # HTTP portal
      - 6789:6789 #optional  # Mobile speed test port
      - 161:161/udp #SNMP
    restart: always # Restart policy for the container
    depends_on:
      - db
    networks:
      frontend: null
      backend: null
networks:
  frontend: null
  backend: null
```

.env

```
DB_USER=unifi
DB_PW=password
DB_NAME=unifi_db
```

init-mongo.js

```
db.getSiblingDB("unifi_db").createUser({user: "unifi", pwd: "password", roles: [{role:
"dbOwner", db: "unifi_db"}]});
db.getSiblingDB("unifi_db_stat").createUser({user: "unifi", pwd: "password", roles: [{role:
"dbOwner", db: "unifi_db_stat"}]});
```

---

Revision #3
Created 2024-07-31 09:39:01 UTC by CoreStream Group
Updated 2024-07-31 09:52:27 UTC by CoreStream Group