y-nana-docker-compose
https://youtu.be/SXwC9fSwct8

Table of Contents

1. Preconditions

1.1. Docker installation

```
curl -L https://get.docker.com | sh && sudo usermod -aG docker $USER && sudo reboot
```

2. How to run MONGO stack using Docker only

You need to perform next steps:

1. Create a new docker network

```
docker network create mongo-network
```

2. Run MongoDB container

```
docker run \
-p 27017:27017 \
-e MONGO_INITDB_ROOT_USERNAME=admin \
-e MONGO_INITDB_ROOT_PASSWORD=superSecret \
--network mongo-network \
--name mongodb \
mongo
```

3. Start Mongo-express container

WORKS (access to WEB UI using admin:pass)

```
docker run \
-p 8081:8081 \
-e ME_CONFIG_MONGODB_ADMINUSERNAME=admin \
-e ME_CONFIG_MONGODB_ADMINPASSWORD=superSecret \
-e ME_CONFIG_MONGODB_SERVER=mongodb \
-e ME_CONFIG_MONGODB_URL=mongodb://admin:superSecret@mongodb:27017/ \
--network mongo-network \
--name mongo-express \
mongo-express
```

WORKS (access to WEB UI without any credentials)

```
docker run \
-p 8081:8081 \
-e ME_CONFIG_MONGODB_ADMINUSERNAME=admin \
-e ME_CONFIG_MONGODB_ADMINPASSWORD=superSecret \
-e ME_CONFIG_MONGODB_SERVER=mongodb \
-e ME_CONFIG_MONGODB_URL=mongodb://admin:superSecret@mongodb:27017/ \
-e ME_CONFIG_BASICAUTH=false \
--network mongo-network \
--name mongo-express \
mongo-express
```

WORKS (access to WEB UI with custom credentials – test:test for this example)

```
docker run \
-p 8081:8081 \
-e ME_CONFIG_MONGODB_ADMINUSERNAME=admin \
-e ME_CONFIG_MONGODB_ADMINPASSWORD=superSecret \
-e ME_CONFIG_MONGODB_SERVER=mongodb \
-e ME_CONFIG_MONGODB_URL=mongodb://admin:superSecret@mongodb:27017/ \
-e ME_CONFIG_BASICAUTH=true \
-e ME_CONFIG_BASICAUTH_USERNAME=test \
-e ME_CONFIG_BASICAUTH_PASSWORD=test \
--network mongo-network \
--name mongo-express \
mongo-express
```

3. MONGO stack using docker compose

Create a new docker-compose.yaml file

```
version: '3.1'
services:
  mongo-db:
    image: mongo
    ports:
      - 27017:27017
    environment:
      MONGO_INITDB_ROOT_USERNAME: admin
      MONGO_INITDB_ROOT_PASSWORD: superSecret

  mongo-ui:
    image: mongo-express
    ports:
      - 8081:8081
    environment:
      ME_CONFIG_MONGODB_ADMINUSERNAME: admin
      ME_CONFIG_MONGODB_ADMINPASSWORD: superSecret
      ME_CONFIG_BASICAUTH: true
      ME_CONFIG_BASICAUTH_USERNAME: test
      ME_CONFIG_BASICAUTH_PASSWORD: test
      ME_CONFIG_MONGODB_SERVER: mongo-db
      ME_CONFIG_MONGODB_URL: mongodb://admin:superSecret@mongo-db:27017/
    depends_on:
        - 'mongo-db'
```

How to start the stack and delete all the containers?
docker compose up – use this command it there is only 1 docker-compose.yaml file in the folder where you are going to use this command
docker compose -f <yaml File> up – use this command if you have more than 1 yaml file or it has a custom name

How to stop the stack and delete all the containers?
docker compose down
docker compose -f <yaml File> down

You also can usestart/stop – in this case containers will not be deleted
docker compose start OR docker compose stop
docker compose -f <yaml File> start OR docker compose -f <yaml File> stop

3.1. The Final version of docker-compose.yaml file

```yaml
version: '3.1'
services:
  mongo-db:
    image: mongo
    ports:
      - 27017:27017
    environment:
      MONGO_INITDB_ROOT_USERNAME: admin
      MONGO_INITDB_ROOT_PASSWORD: superSecret
    volumes:
      - mongo-db_data:/data/db
    networks:
      - mongo-compose-network

  mongo-ui:
    image: mongo-express
    ports:
      - 8081:8081
    environment:
      ME_CONFIG_MONGODB_ADMINUSERNAME: admin
      ME_CONFIG_MONGODB_ADMINPASSWORD: superSecret
      ME_CONFIG_BASICAUTH: true
      ME_CONFIG_BASICAUTH_USERNAME: test
      ME_CONFIG_BASICAUTH_PASSWORD: test
      ME_CONFIG_MONGODB_SERVER: mongo-db
      ME_CONFIG_MONGODB_URL: mongodb://admin:superSecret@mongo-db:27017/
    depends_on:
      - 'mongo-db'
    networks:
      - mongo-compose-network

volumes:
  mongo-db_data:
    driver: local

networks:
  mongo-compose-network:
    driver: bridge
```