Uge 15 - Dockerfile, volumes, bind mount og network

Referencer

- Dockerfile keywords - https://docs.docker.com/reference/dockerfile/
- Dockerfile best practices - https://docs.docker.com/build/building/best-practices/
- Mosquitto Dockerfile eksempel: https://dev.to/johnscode/a-basic-mqtt-docker-deployment-1aah
- Eksempel SQLite3 Dockerfile: https://github.com/KEINOS/Dockerfile_of_SQLite3/blob/master/Dockerfile

Hvad er Dockerfile?

- Til at konfigurere et image så det har et andet startpunkt når en container laves ud fra det
- F.eks. installér et program på Debian image, prop nogle konfigurationer og andre filer fra host PC ind i det, definér volumes, expose porte osv. uden at det skal gøres som flags

Dockerfile nærmere

- I en Dockerfile kan vi f.eks.:
  - Køre alm. terminalkommandoer ligesom med shell scripts
  - Kopiere en mappe fra host til container
  - Expose en port
  - Tilføje environment variables til container

Hvorfor Dockerfile?

- Når en container laves fra en Dockerfile har den samme startpunkt uanset hvilken enhed det bygges på
- Tillader jer at lave permanente (men samtidigt justerbare) konfigurationer til Docker images
- Lave let deployment af en skræddersyet løsning til flere maskiner
- Meget let syntax

Docker network

- Bruges til at forbinde containere (og host hvis relevant)
- Kan kommunikere mellem containere ved brug af container name eller container IP adresse
- Avanceret brug er `overlay` network f.eks., hvor containere på forskellige hosts kan forbindes

Docker network type 1 - `bridge`

- Hyppigt anvendt; forbinder containere der er på samme bridge netværket
- Det er muligt at have flere bridge netværk på én host; i så fald kan kun containere på samme bridge netværk forbinde
- Medmindre andet er angivet forbinder nye containere automatisk til default bridge
- Sikkerhedsanbefalingen er dog at hvis 2 containere skal forbindes, så lav ét bridge netværk til dem

Brugerdefineret Docker bridge eksempel

- Opret eget bridge netværk der hedder `bruger-bridge-net`
- Forbind eclipse-mosquitto container til dette netværk

```
docker network create -d bridge bruger-bridge-net
docker run --network=bruger-bridge-net -itd eclipse-mosquitto
```

Lille midtvejs-øvelse

- Lav et netværk til FMS projekt
- Start en container der indeholder MQTT broker og slut til dit brugerdefinerede netværk
- Lav et Docker image til mosquitto-clients (f.eks. ved at bruge Debian image fra Docker Hub)
- Start 2 containere ud fra det image - én publisher, én subscriber og forbind begge til det samme netværk som broker
- Få alle 3 til at snakke sammen
- Test også at de ikke kan kommunikere udenfor bridge netværk

Liste af Docker netværk på maskinen

- `docker network ls` eller `docker network list
- Med navn på netværk fra ovenstående kommando kan netværket inspiceres:
  - `docker network inspect NETWORK`
- Lav et nyt med `docker network create --driver=bridge NETWORK`
- Check efter med `docker network ls`

Forbind eksisterende container til et netværk

- Tjek eksempler på: https://docs.docker.com/reference/cli/docker/network/connect/
    - Det er muligt at "hotplug" en container til et netværk, altså gøre det mens den allerede kører.
    - Muligt at indstille så container altid forbinder til et bestemt netværk når den starter

Docker network avanceret

- Subnetting: `docker network create --driver=bridge --subnet=192.168.0.0/16 br0`
- IP range og gateway:

```
docker network create \
 --driver=bridge \
 --subnet=172.28.0.0/16 \
 --ip-range=172.28.5.0/24 \
 --gateway=172.28.5.254 \
 br0
```

Opgave 1 - Dockerfile Mosquitto

```
* Lav en Dockerfile der kan bruges til at bygge et image der overholder følgende:
* Expose port
* Bruger config fra host (som ligger i en mappe i samme niveau som Dockerfile)
* Starter automatisk broker når container køres
* Tilføj autenticering (ligesom før med broker)
```

Test der viser I har udført opgave 1

- I kan bygge image med `docker build` kommando
- I kan køre en container baseret på det med `docker run`
- I kan publish og subscribe med autenticering

Opgave 2 - Dockerfile SQLite3

- Lav en Dockerfile der kan bruges til at bygge et image der overholder følgende:
  - Indeholder en tabel der passer til jeres FMS projekt (skal indeholde tidsstempel også)
  - Det er muligt at indsætte rækker (`INSERT`) fra CLI på host computer
  - Det er muligt at trække rækker/tabel ud (`SELECT`) fra CLI på host computer

Tests der viser i har udført opgave 2

- I kan bygge image og køre det uden fejl med `docker run`
- I kan bruge `docker exec` til at `SELECT` fra den database I allerede har lagt i den
- I kan bruge `docker exec` til at `INSERT` i den samme tabel i lige har kørt `SELECT` på

Opgave 3 - Docker network - få systemerne til at snakke sammen

- Vælg og løs én af følgende:
  - Brug Docker network til at forbinde de 2 containere (MQTT og SQLite) på en måde sådan at data kan modtages via subscriber i én container og derefter indsættes i database i SQLite container
  - Brug almindelig shell script og/eller C++ program på host computer til at modtage via subscriber (forbundet til container broker) og indsætte data i SQLite tabel i container.

Tests der viser i har udført opgave 3, del 1

- Subscriber skal automatisk indsætte data i SQLite, så i skal kunne publish fra host, og `SELECT` fra SQLite containeren og bevise at publiceret data er lagt ind i databasen.

Tests der viser i har udført opgave 3, del 2

- Kommer an på hvordan i har sat koden op. Jeg regner med at publish fra host eller anden computer så kan bevises enten ligesom del 1, eller at i faktisk i jeres kode/script har shell output der siger når data er blevet indsat.