

Title: Docker compose wait for database service initialisation

Post Body:

I have a spring boot project which I'd like to containerize using docker. I have a couple of spring boot applications which connect to same MySql server.

My spring applications requires the database to be completely setup (i.e. all the tables to be created and some data to be inserted in some of the tables) in order to start.

I am using **Docker** version **18.09.0** and **docker-compose** version **1.23.1** and ubuntu **16.04** LTS

I have two files `create.sql` and `insert.sql`, which I use to initialise the database to be used by the application.

I create the images using the command `docker-compose.yml` and it runs successfully and creates the images.

I have the following questions.

I assume when using docker-compose, a container starts as soon as all its dependent containers have started. Is there a way to wait for the mysql server to be up and ready to accept connections, before my API container gets started?

If I chose to create containers separately for the applications and mysql, and not use docker-compose, how do I make sure that my applications connect to the mysql container?

Is there any other tool which might help me achieve this?

Note: I have tried to use `docker inspect <container_id>` to find the the IpAddress for the mysql container and use it to connect, but it doesn't work as well.

The following are the files I am using to create images.

`docker-compose.yml` file.

```
version: '3'
services:
  demo-mysql:
    image: demo-mysql
    build: ./demo-mysql
    volumes:
      - /mnt/data/mysql-data:
```

The following is the `Dockerfile` for the spring boot project

```
FROM java:8
VOLUME /tmp
ARG DATA_PATH=/src/main/resources
ARG APP_PORT=8080
EXPOSE ${APP_PORT}
ADD /build/libs/demo-api.jar
```

The following is the `Dockerfile` I used to create my mysql image

```
FROM mysql:5.7
ENV MYSQL_DATABASE=demo \
    MYSQL_USER=root \
    MYSQL_ROOT_PASSWORD=root
ADD ./1.0/create.sql /docker-ent
```

Accepted Answer: None

Highest Rated Answer:

Use the `healthcheck` feature of docker-compose (<https://docs.docker.com/compose/compose-file/#healthcheck>). Something like this:

```
services:
  demo-mysql:
    image: demo-mysql
    build: ./demo-mysql
    volumes:
      - /mnt/data/mysql-data:/var/lib/mysql
```

The depending containers will not start until the demo-mysql container is healthy