Now we can download the code from [here](https://github.com/sanjoy-sust/book-manager/tree/feature/docker). Or we can clone from and checkout the **feature/docker**branch.  Here we can see the Dockerfile is similar to the code described on my previous [article](https://dzone.com/articles/all-about-hibernate-manytomany-association).

**Dockerfile**

FROM java:8

2

COPY /build/libs/book-manager-1.0-SNAPSHOT.jar book-manager-1.0-SNAPSHOT.jar

3

ENTRYPOINT ["java","-jar","book-manager-1.0-SNAPSHOT.jar"]

And the docker-compose.yml file looks like this:

version: '3'

services:

docker-mysql:

restart: always

container\_name: docker-mysql

image: mysql

environment:

MYSQL\_DATABASE: book\_manager

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_ROOT\_HOST: '%'

volumes:

- ./sql:/docker-entrypoint-initdb.d

ports:

- "6033:3306"

healthcheck:

test: "/usr/bin/mysql --user=root --password=root--execute \"SHOW DATABASES;\""

interval: 2s

timeout: 20s

retries: 10

book-manager-app:

restart: on-failure

build: ./

expose:

- "10222"

ports:

- 10222:10222

environment:

WAIT\_HOSTS: mysql:3306

depends\_on:

- docker-mysql

I defined two services named by docker-mysql and book-manager-app. Service book-manager-app is dependent on docker-mysql. We are using docker-compose version 3. MySQL will run on the 3306 port at a docker container but we can access it publicly from 6033. book-manager-app will run on port 10222.  We have an initial DML and DDL file at the SQL directory which will run during startup time of Docker and MySQL setup.

**Run Application With docker-compose**

Now we will apply some commands to run our application. I think we have already downloaded project from the above [link](https://github.com/sanjoy-sust/book-manager/tree/feature/docker). We will go to the project root directory. To run the application we will use following commands:

* docker-compose up — This will execute Dockerfile commands and will run services defined in the docker-compose file.
* docker-compose down — This will stop and remove all containers that were running by docker-compose file.
* docker-compose up --build — If we do an update on the Dockerfile, the war/jar file, or the docker-compose file, then we have to execute this command to get updated data on the Docker machine.

So, I think after running docker-compose up, It runs the application with all services in the Docker machine. Ignore some initial exception logs for database communication. To check whether it runs or not, we can check <http://localhost:10222/book>.