

first of all create -file with name "Dockerfile" in root directory of project
for simple java file

```
Dockerfile > ...
1 FROM openjdk:8
2
3 WORKDIR /app
4
5 COPY . /app/
6
7 RUN javac Main.java
8
9 ENTRYPOINT [ "java", "Main" ]
```

Docker build -t jack/docker-helloworld .

Docker run jack/docker-helloworld .

for springboot

```
FROM mcr.microsoft.com/openjdk/jdk:17-ubuntu
COPY PortfolioUsingSpringBoot/target/PortfolioUsingSpringBoot-0.0.1-SNAPSHOT.jar PortfolioUsingSpringBoot-0.0.1-SNAPSHOT.jar
ENTRYPOINT ["java", "-jar", "/PortfolioUsingSpringBoot-0.0.1-SNAPSHOT.jar"]
```

docker build -t jagdish/docker-portfolio . -> to build

docker images -> to see images

docker ps -> to see running container

docker ps -a -> to see stopped container

docker rm id-name -> to remove container after stopping

docker images rm id-name -> to remove images

docker commands for spring boot app

docker build -t jagdish/portfolio-springboot . -> to build

docker images -> to see images

```
docker run -p 9090:8080 704b70baaa5a -> to run container  
ctr c or docker stop id      -> to stop  
docker start id - to run again    -> to start again after stopping
```

dockerizing springboot with MySQL using docker network (approach1)

```
docker pull MySQL:5.7  
docker images  
docker network create springboot-mysql-net  
docker network ls  
docker run --name mysqlDb --network springboot-mysql-net -e MYSQL_ROOT_PASSWORD=e  
MYSQL_DATABASE=Db-name -e MYSQL_USER=sa -e MYSQL_PASSWORD=1234 -d MySQL:5.7  
docker exec -it <container-id> bash  
MySQL -u<username> -p<password>  
show databases
```

-----Update the application.properties file

```
spring.datasource.url=jdbc:mysql://mysqlDb:3306/dbname  
spring.datasource.username=sa  
spring.datasource.password=1234
```

to build-----

```
docker build -t springbootmysql .  
docker images
```

```
docker run --network springboot-mysqtnet --name springboot--container -p  
8080:8080(container port:host port) -d "repo-name"
```

dockerizing springboot with MySQL using docker compose (approach2)

first create docker-compose.yml file in root

9

```
version: "3.8"  
services:  
  server:  
    build: .  
    restart: always  
    ports:  
      - 8080:8080  
    depends_on:  
      - mysqldb  
  mysqldb:  
    platform: linux/x86_64  
    image: "mysql:5.7"  
    restart: always  
    ports:  
      - 3307:3306  
    environment:  
      MYSQL_DATABASE: expensetracker  
      MYSQL_USER: scbushan05  
      MYSQL_PASSWORD: scbushan05  
      MYSQL_ROOT_PASSWORD: scbushan05
```

Configure application.properties file accordingly and build jar using install

```
#mysql properties  
spring.datasource.url=jdbc:mysql://mysqldb:3306/expensetracker  
spring.datasource.username=scbushan05  
spring.datasource.password=scbushan05  
  
#postgresql properties  
#spring.datasource.url=${JDBC_DATABASE_URL}  
#spring.datasource.username=${JDBC_DATABASE_USERNAME}  
#spring.datasource.password=${JDBC_DATABASE_PASSWORD}  
#spring.datasource.driver-class-name=org.postgresql.Driver  
  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true  
  
#server.port=5000  
  
server.servlet.context-path=/api/v1  
  
jwt.secret=b2tech
```

docker-compose up

