**Linking Microservice with MySQL**   
  
To link authentication microservice with MySQL database follow steps below:

* Create a file named 'Dockerfile' in the authentication-service folder with the below specified content

FROM openjdk:8-jdk-alpine

COPY target/authentication-service-0.0.1-SNAPSHOT.jar app.jar

ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom -Djava.net.preferIPv4Stack=true","-jar","/app.jar"]

* Explanation for the Dockerfile configuration
  + FROM command pulls JDK 8 from docker hub
  + COPY command copies the jar created in target folder to the project root folder with name app.jar
  + ENTRYPOINT execute the java command and starts the REST API Service
* Modify docker-build\docker-compose.yml so that the entire file content looks like the one below. Ideally authentication-service had been added in the file:

version: '3'

services:

  payroll-mysql:

    image: mysql:8.0.18

    ports:

      - "33061:3306"

    environment:

      - MYSQL\_ROOT\_PASSWORD=root

    volumes:

      - d:/payrolldb:/var/lib/mysql

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

  authentication-service:

    image: authentication-app

    build: authentication-service/.

    ports:

      - 8091:8091

    depends\_on:

      - payroll-mysql

    links:

      - payroll-mysql

* Explanation for authentication-service configuration
  + image - defines the name for authentication-service
  + build - denotes that Docker file is present in authentication-service folder
  + ports
    - left hand side port number denotes the port that will be exposed
    - right hand side port number denotes the port number defined in the application.properties file of authentication-service
  + depends\_on - denotes that authentication-service requires start of mysql server
  + links - denotes that authentication-service is linked to payroll-mysql. This ensure database connectivity from authentication-service to mysql database
* Find below the database connection changes that needs to be done in application.properties file of authentication-service. Look out for the following changes
  + MySQL connection URL changes
    - localhost changed to payroll-mysql
    - port number changed to 33061
    - Configures public key retrieval and SSL
  + Change port of the microservice (server.port) to avoid port number conflicts
* The primary change is that, we are changing the localhost as payroll-mysql, port number. Change the server port aligned to the docker compose configuration.

server.port=8091

spring.datasource.url=jdbc:mysql://payroll-mysql:33061/ormlearn?allowPublicKeyRetrieval=true&useSSL=false

* Execute maven build in command line on authentication-service folder to create jar file with updated configuration
* Execute 'docker-compose up' command in docker-build folder. This will start mysql server and authentication-service
* Test the *http://localhost:8090/authenticate* REST API using Postman or curl command and verify if the service works end to end and returns back the token.