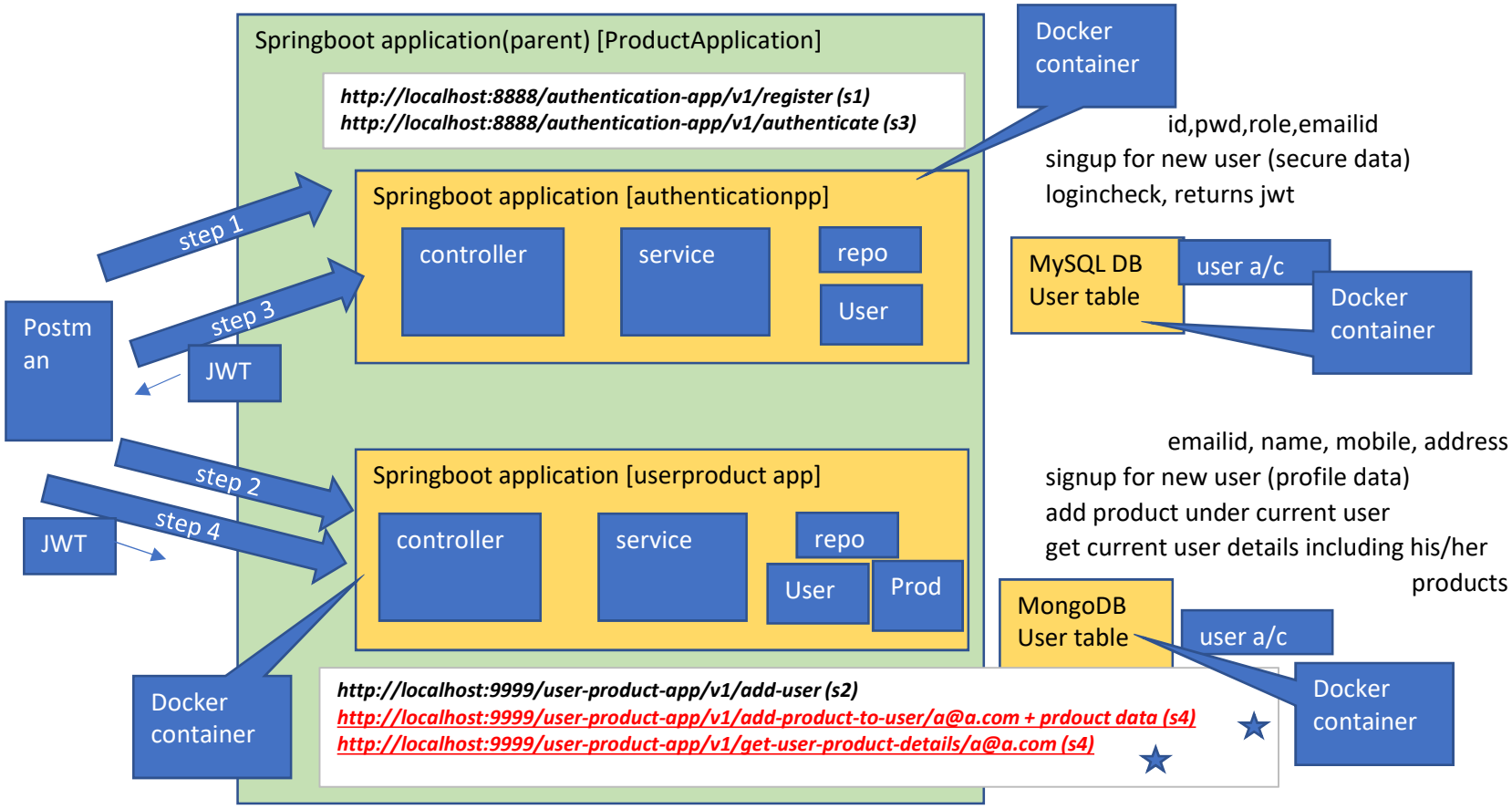
Stage 3

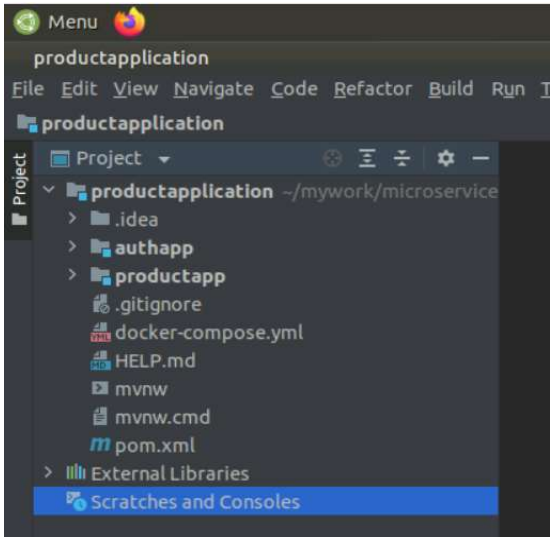

Springboot application(parent) [ProductApplication]
http://localhost:8888/authentication-app/v1/register (s1)
http://localhost:8888/authentication-app/v1/authenticate (s3)
Docker container
id,pwd,role,emailid
singup for new user (secure data)
logincheck, returns jwt
Springboot application [authenticationpp]
controller
service
repo
User
step 1
step 3
Postman
JWT
MySQL DB
User table
user a/c
Docker container
emailid, name, mobile, address
signup for new user (profile data)
add product under current user
get current user details including his/her products
Springboot application [userproduct app]
controller
service
repo
User
Prod
step 2
step 4
JWT
MongoDB
User table
user a/c
Docker container
Docker container
http://localhost:9999/user-product-app/v1/add-user (s2)
http://localhost:9999/user-product-app/v1/add-product-to-user/a@a.com + prdouct data (s4)
http://localhost:9999/user-product-app/v1/get-user-product-details/a@a.com (s4)

Steps to execute microservice application in docker

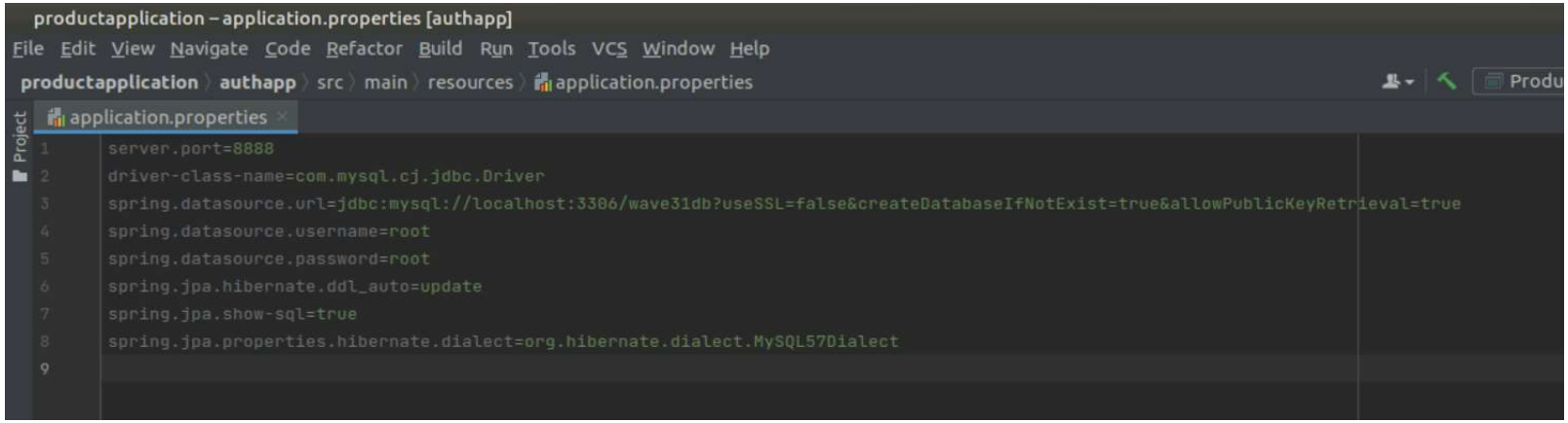
Note : Stop mysql and mongdb services in linux machine before starting below steps

Step 1 copy microsevice application into linux machine

open in intellij



Step 2 Do neccesary changes in application properties if needed



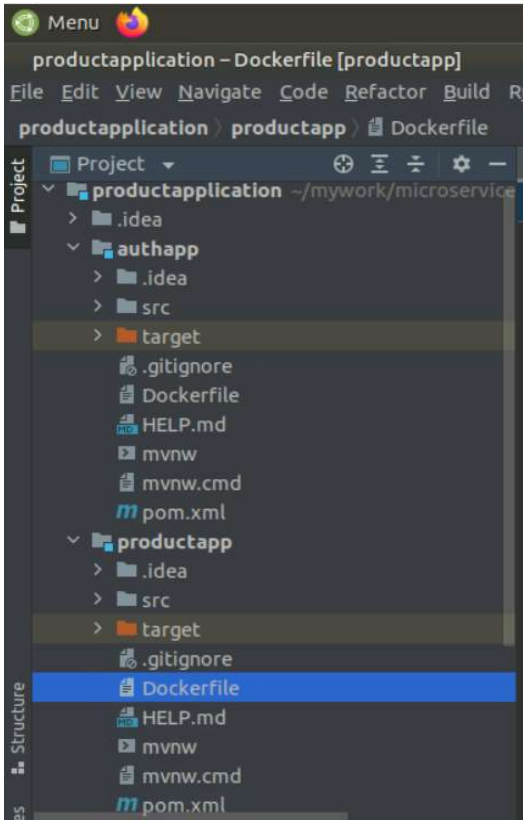
Step 3 Build jars for both child application

Step 4 Create and define Dockerfile for authapp

Dockerfile

Plugins supporting Dockerfile files found.

```
# required openjdk
FROM openjdk

# work directory
WORKDIR usr/lib

# jars
ADD ./target/authapp-0.0.1-SNAPSHOT.jar /usr/lib/authapp-0.0.1-SNAPSHOT.jar

# entry point
ENTRYPOINT ["java","-jar","authapp-0.0.1-SNAPSHOT.jar"]
```



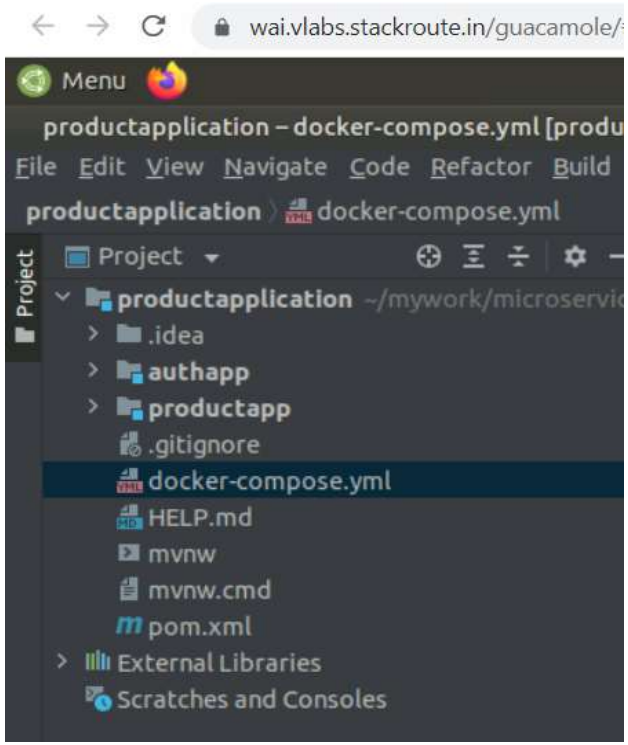
Step 5 Create and define Dockerfile for productapp

Dockerfile

Plugins supporting Dockerfile files found.

```
# required openjdk
FROM openjdk

# work directory
WORKDIR usr/lib

ENV MONGO_DATABASE="sprint5"
ENV MONGO_URL="mongodb://localhost:27017/sprint5"

# jars
ADD ./target/productapp-0.0.1-SNAPSHOT.jar /usr/lib/productapp-0.0.1-SNAPSHOT.jar

# entry point
ENTRYPOINT ["java","-jar","productapp-0.0.1-SNAPSHOT.jar"]
```

Step 6 Create and define compose file at parent level




```
customer_service:
  image: user_product_app_image
  build: productapp/
  container_name: user_produt_app_container
  network_mode: host
  restart: always
  depends_on:
    - mongo_service

mongo_service:
  image: mongo:3.4-jessie
  container_name: mongo_container
  network_mode: host
```

docker-compose.yml

```
version : '3.3'

services:
  auth_service:
    image: auth_app_image
    build: authapp/
    container_name: auth_app_container
    network_mode: host
    restart: always
    depends_on:
      - mysql_service

  mysql_service:
    image: mysql:5.5
    container_name: mysql_container
    network_mode: host

    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_USERNAME: user
      MYSQL_PASSWORD: root
      MYSQL_ALLOW_EMPTY_PASSWORD: "yes"
```

sudo docker-compose up --build

sudo docker-compose down