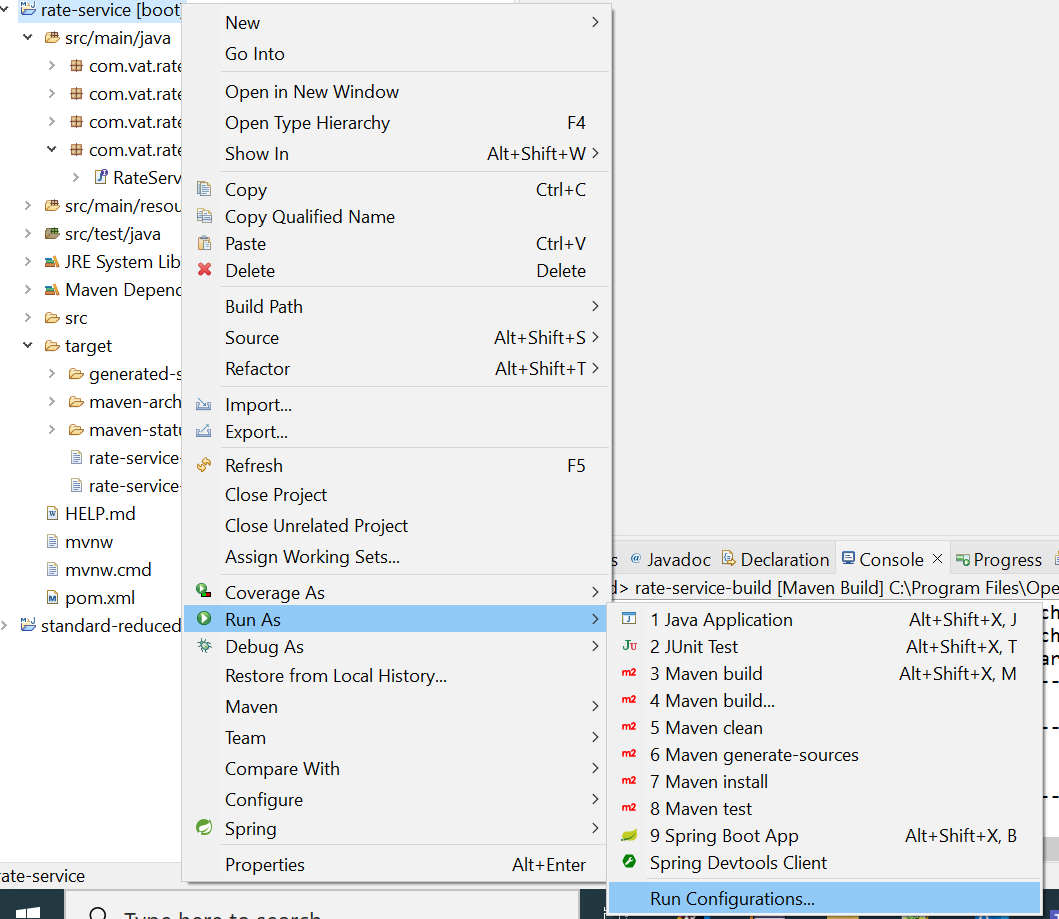
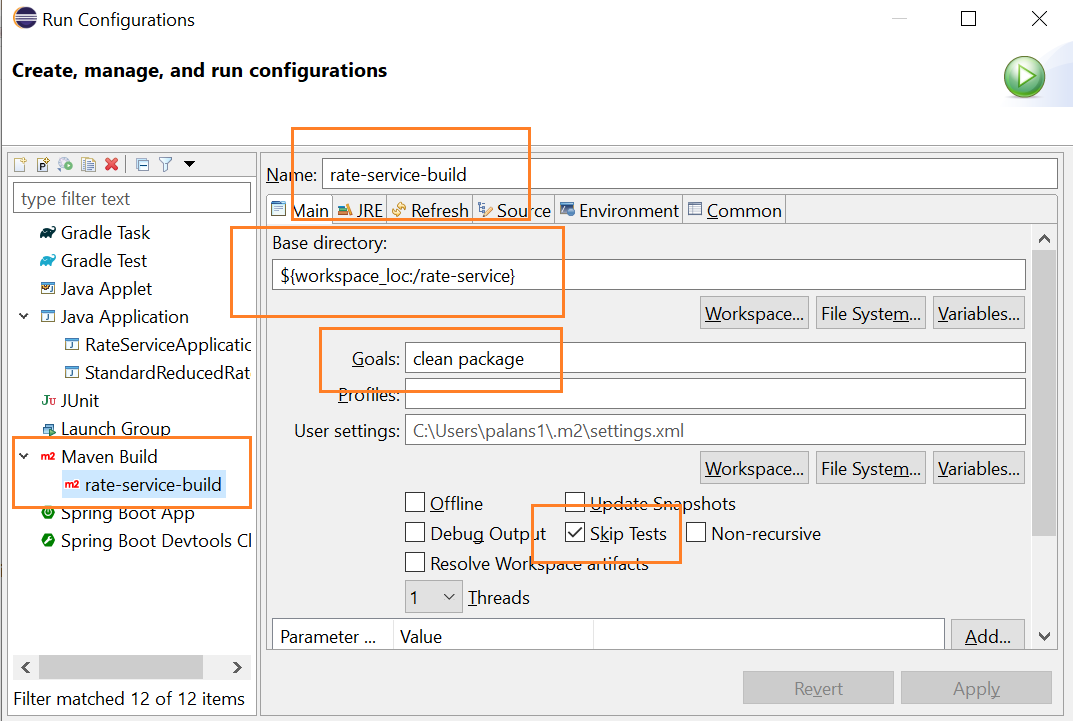
Select rate-service project -> run as -> run configurations



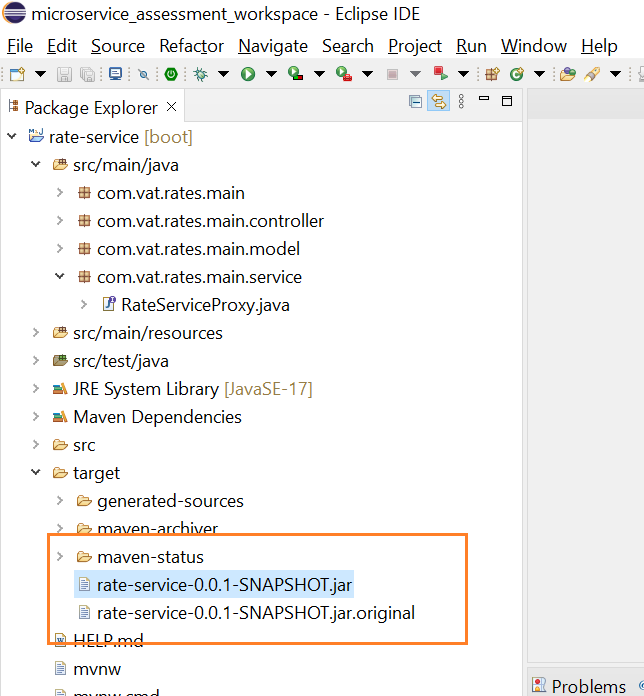
Create new maven build by double click and provide the name of the project build (ex:rate-service-build) and select the project workspace and input goals as “clean package” and check the skip tests



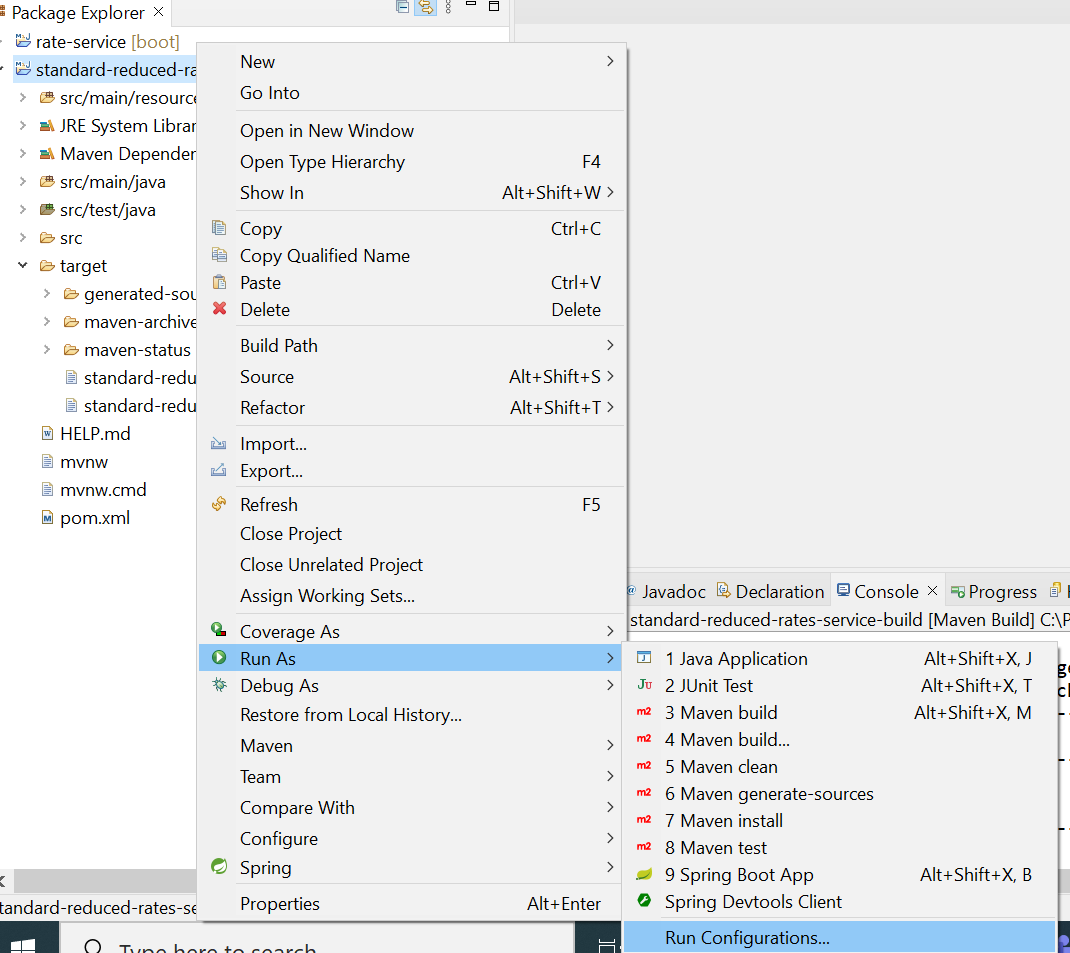
Apply and Run, once build success



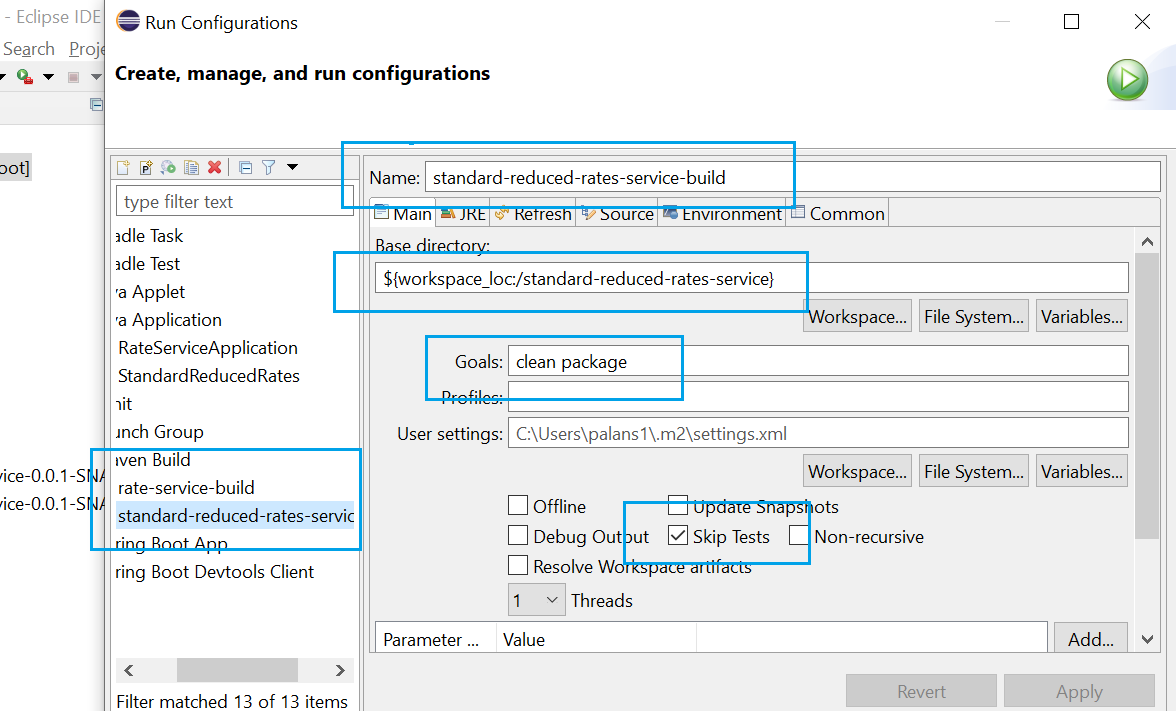
Go to project and refresh, expand the target folder from the project and can see the generated snapshot jar



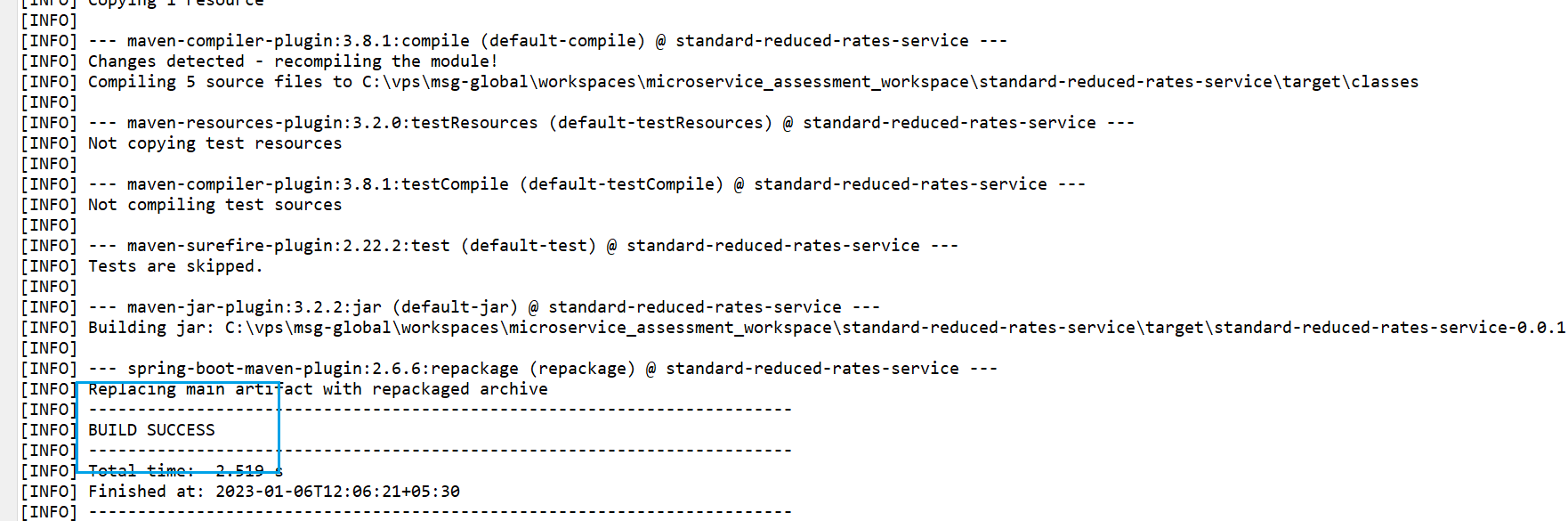
Select standard-reduced-rate-service project -> run as -> run configurations



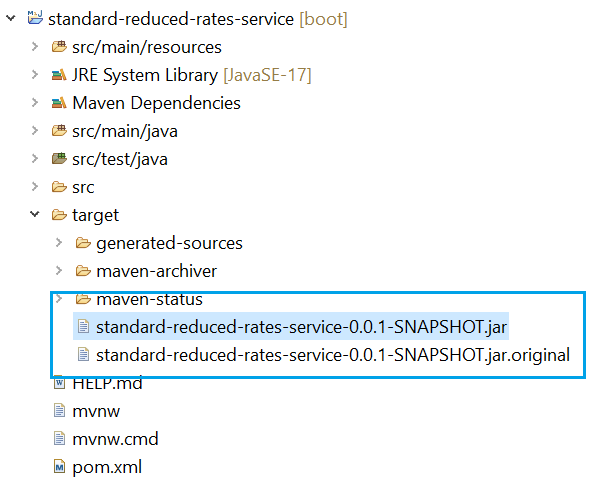
Create new maven build by double click and provide the name of the project build (ex: standard-reduced-rate-service-build) and select the project workspace and input goals as “clean package” and check the skip tests



Apply and Run, once build success

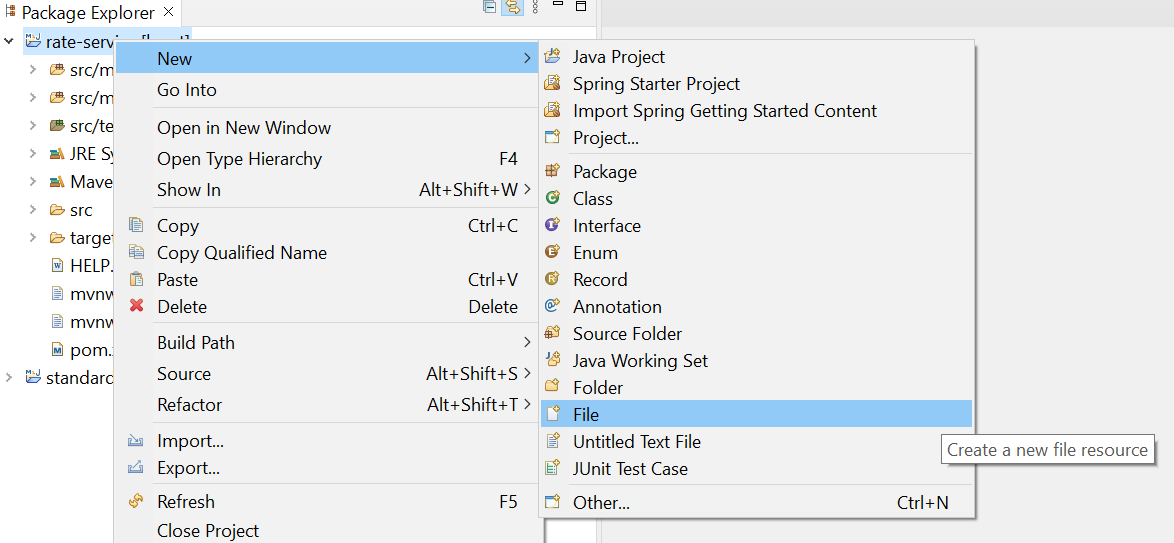


Go to project and refresh, expand the target folder from the project and can see the generated snapshot jar

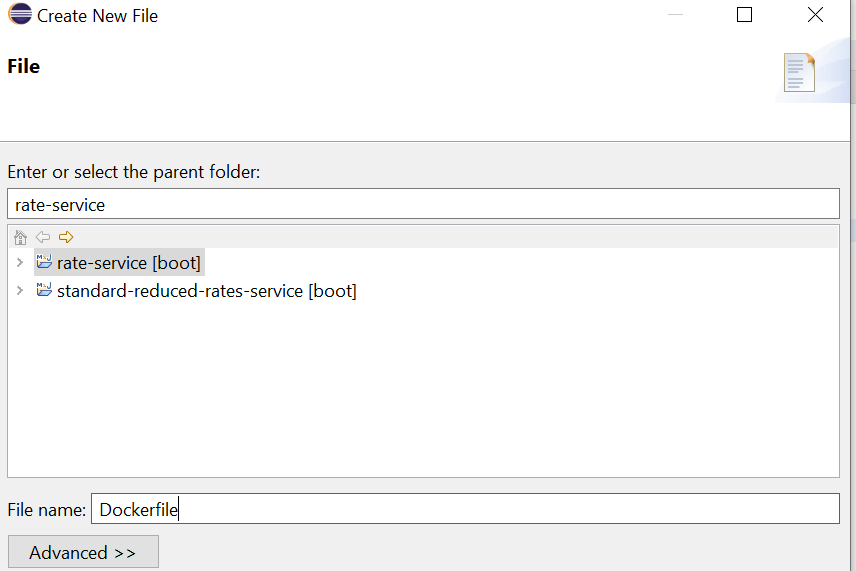


**Docker Image Creation & Container Start**

1. Create docker file by selecting project -> new file -> docker file name



Provide the file name as Dockerfile without extension





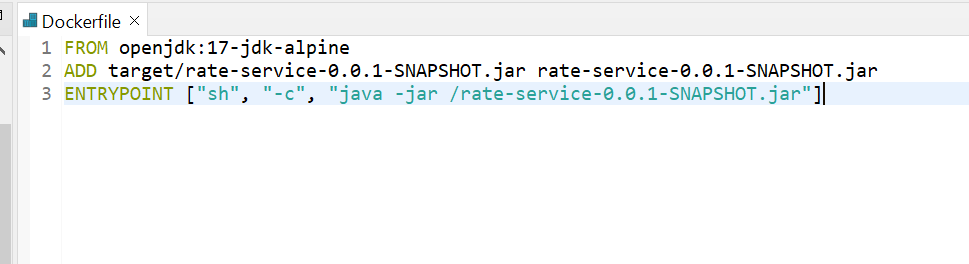
**DOCKER IMAGE – Automotive**

1. Create docker file and add below lines to create a image

FROM openjdk:17-jdk-alpine

ADD target/rate-service-0.0.1-SNAPSHOT.jar rate-service-0.0.1-SNAPSHOT.jar

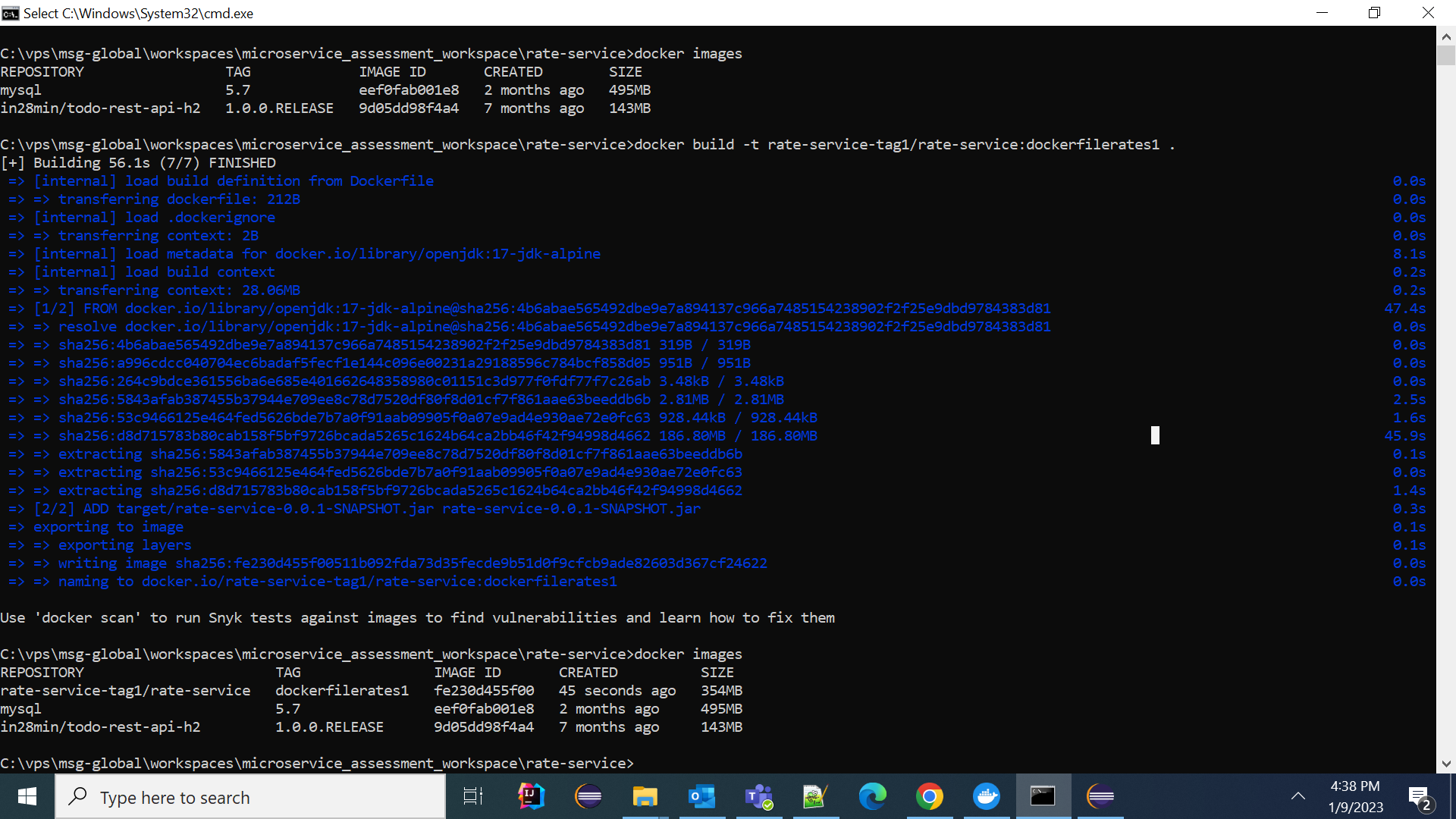
ENTRYPOINT ["sh", "-c", "java -jar /rate-service-0.0.1-SNAPSHOT.jar"]



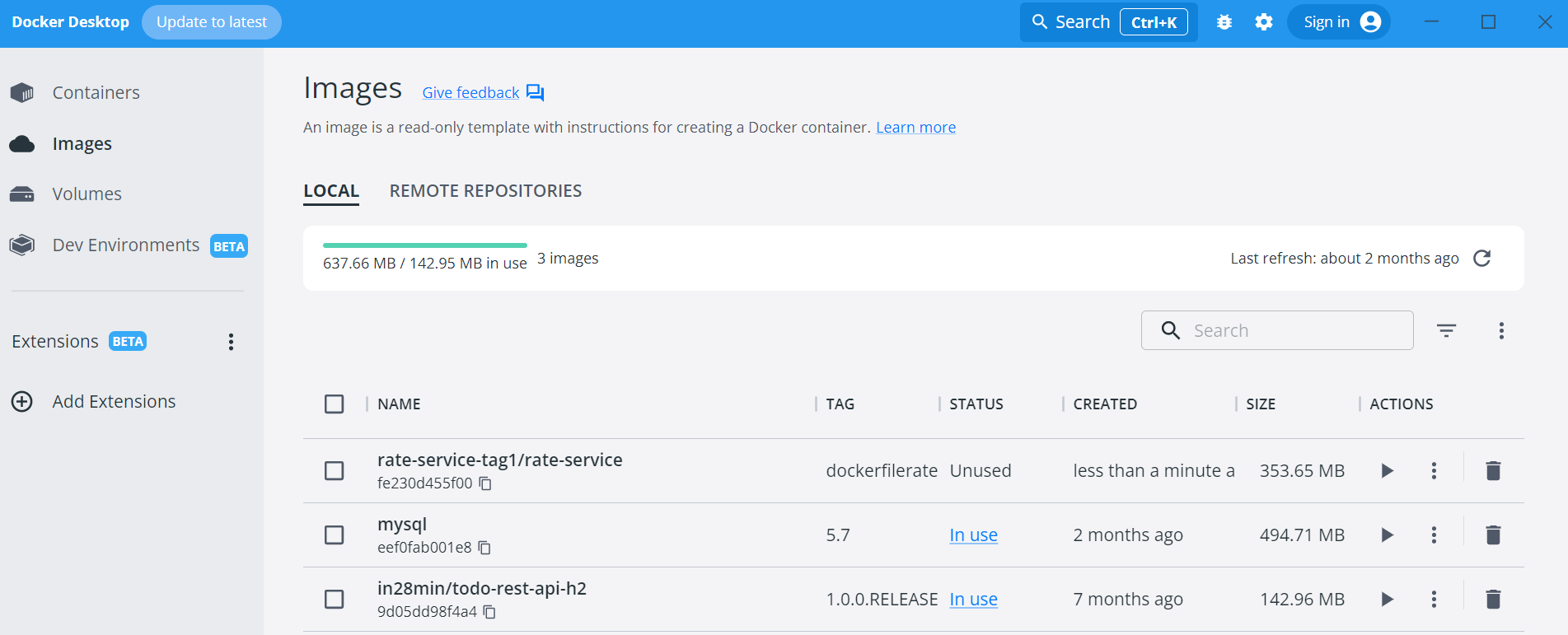
1. Go to docker console/command line

docker build -t <tag name>/<api name><tag> .

$ docker build -t rate-service-tag/rate-service:dockerfilerates .

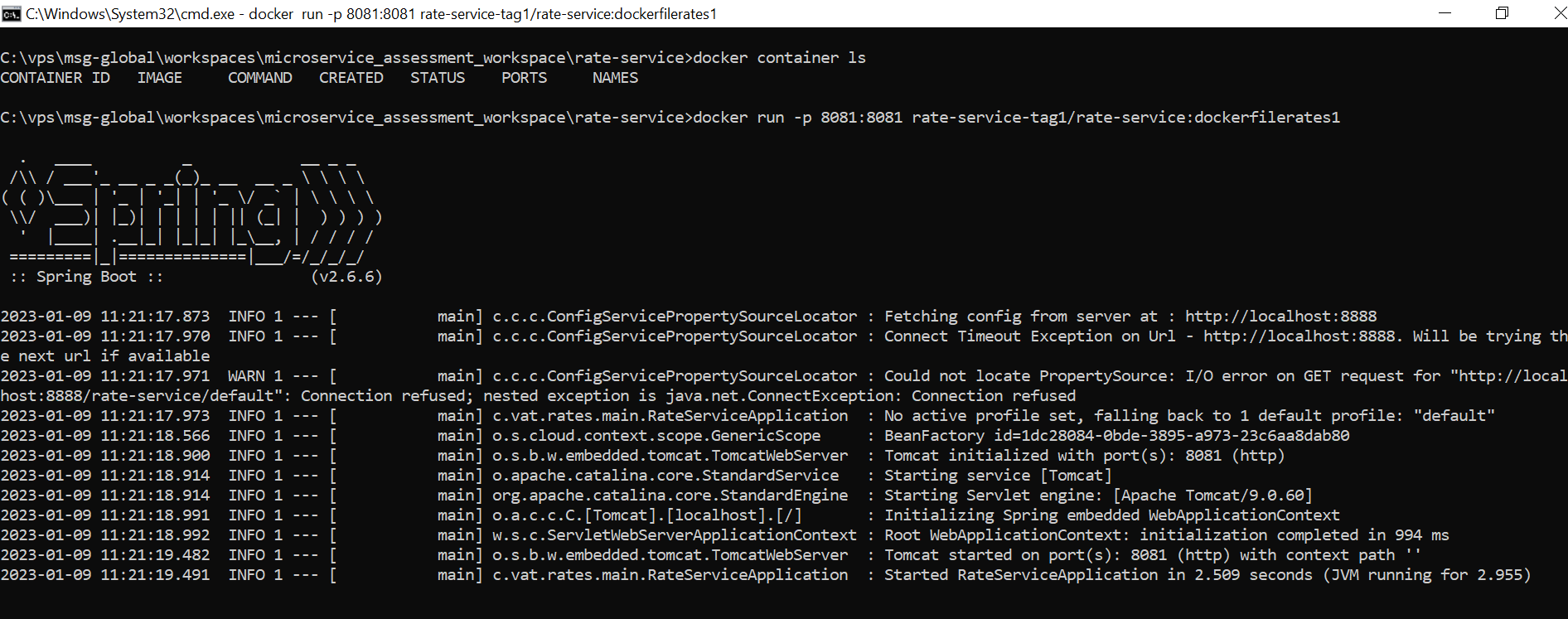


1. Go to Docker Desktop and click Images to list the created images

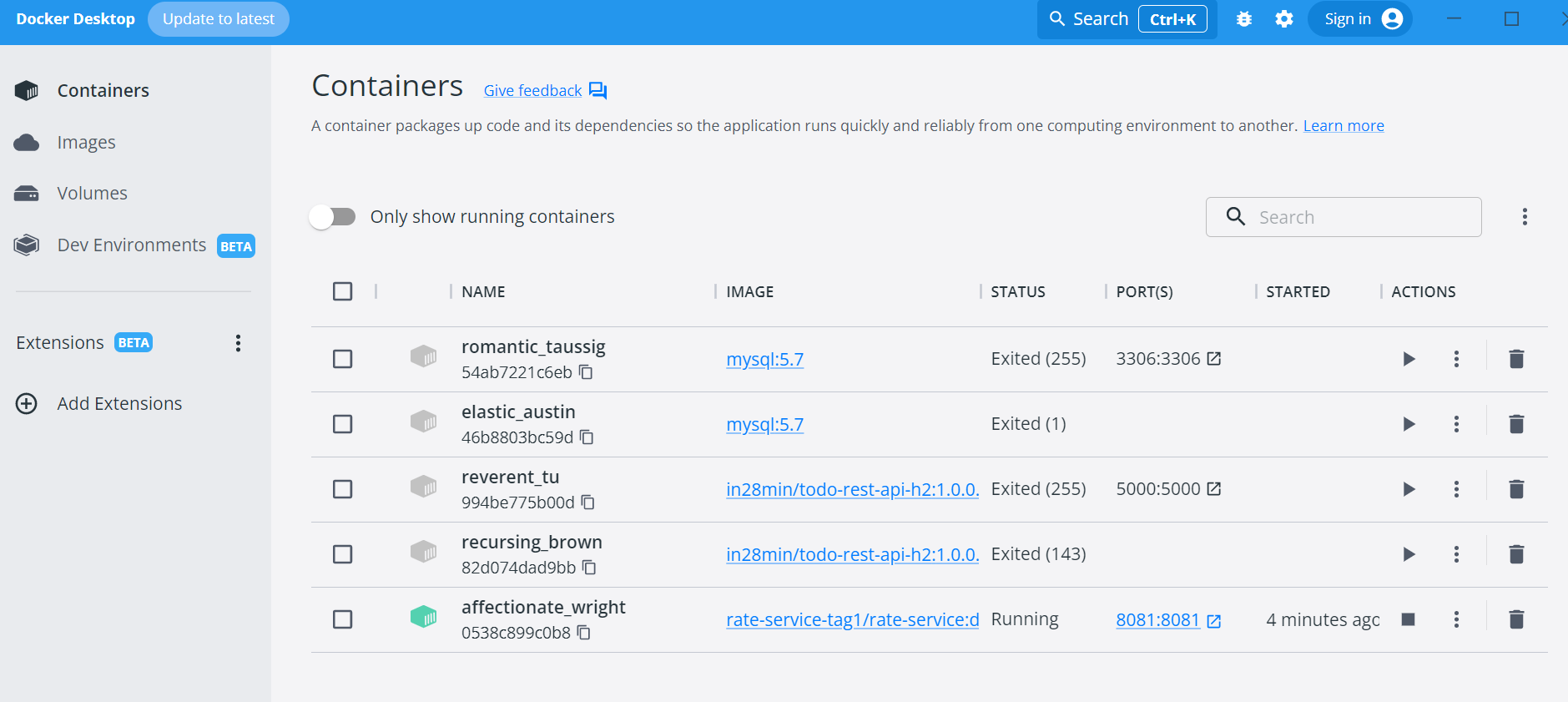


1. Command window execute the command

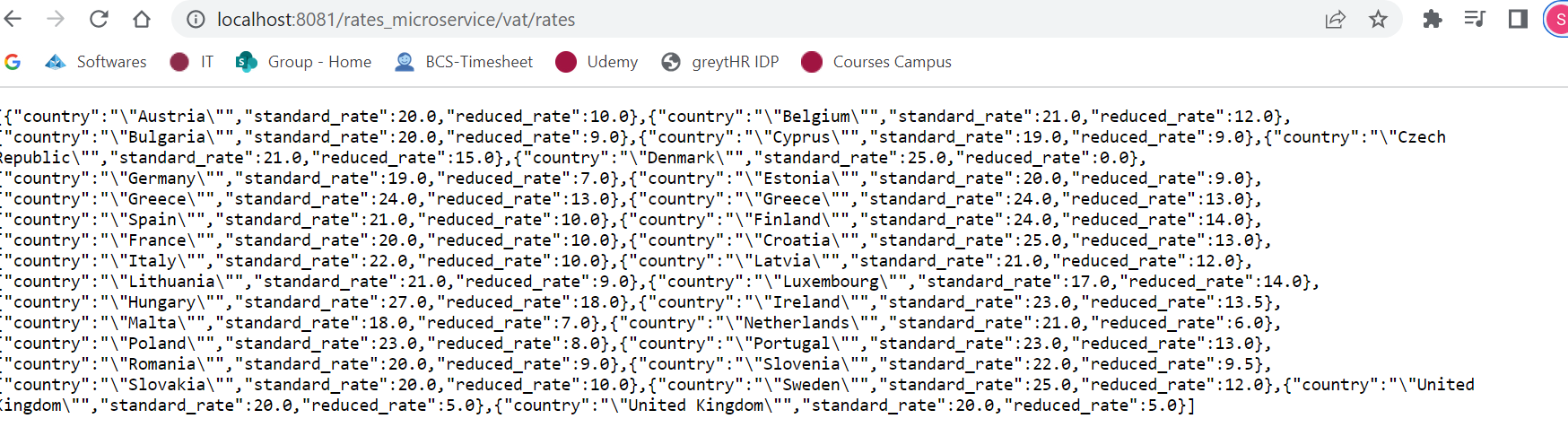
$ docker run -p 8081:8081 rate-service-tag/rate-service:dockerfilerates



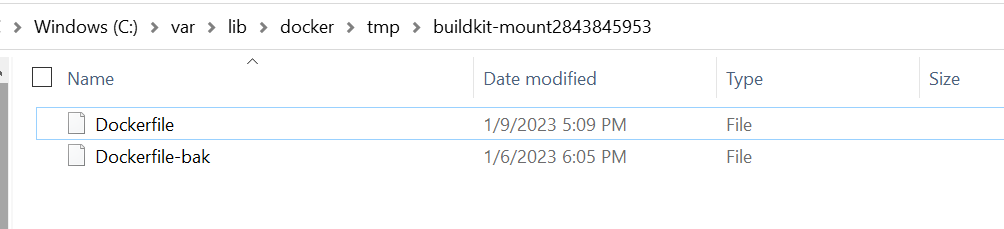
Make sure in docker desktop also running the service



URL: <http://localhost:8081/rates_microservice/vat/rates>



Ref:



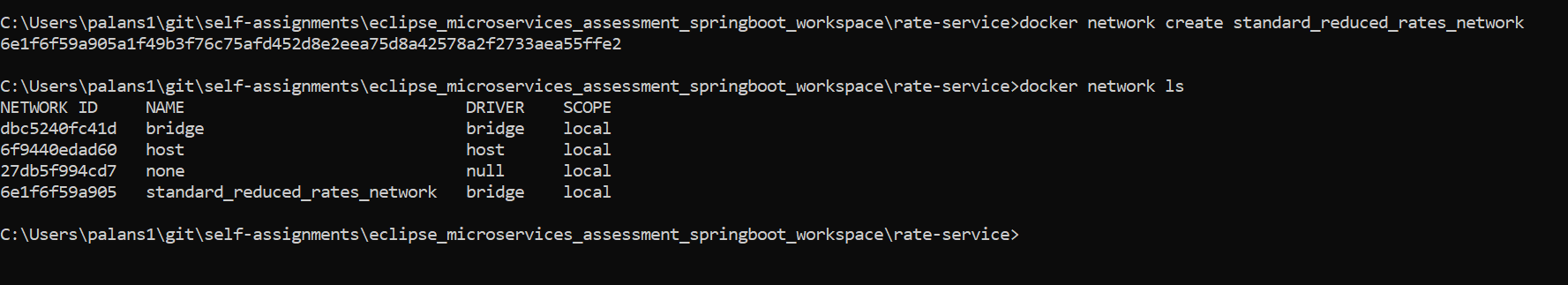
-> docker build -t rate-service-tag/rate-service:dockerfilerates .

-> docker build -t standard-reduced-rates-service-tag/standard-reduced-rates-service:dockerfilestandard-reduced-rates-service .

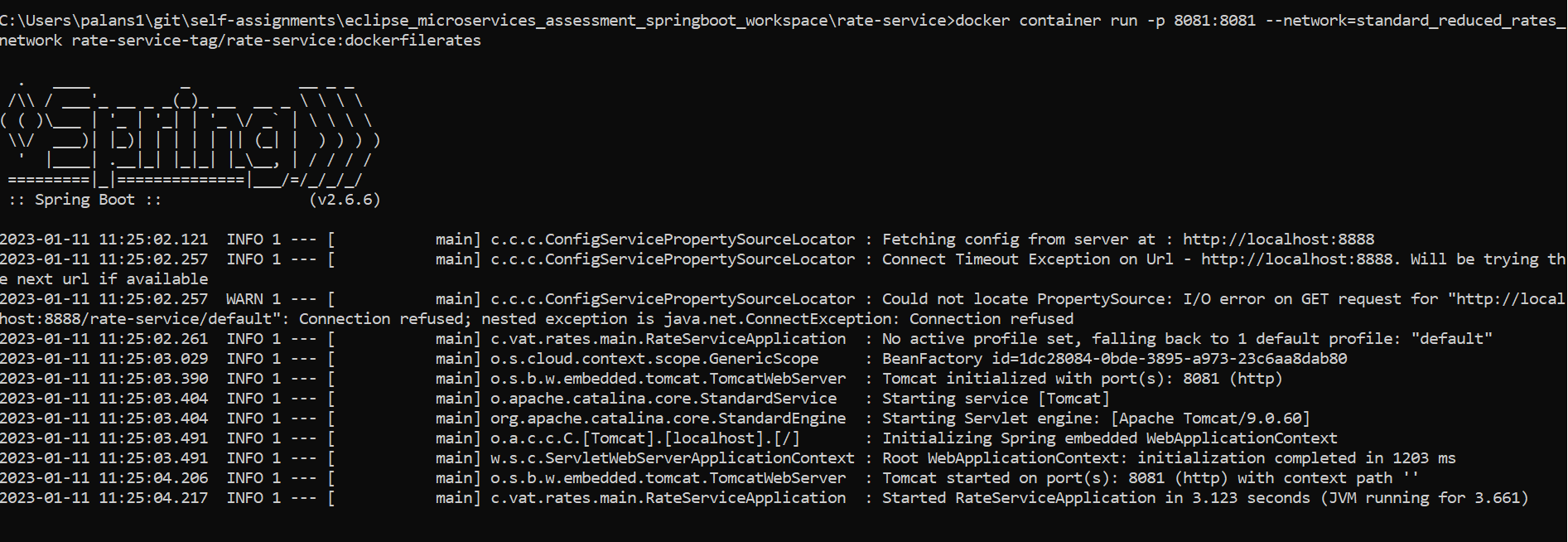
-> docker network create standard\_reduced\_rates\_network

-> docker network ls

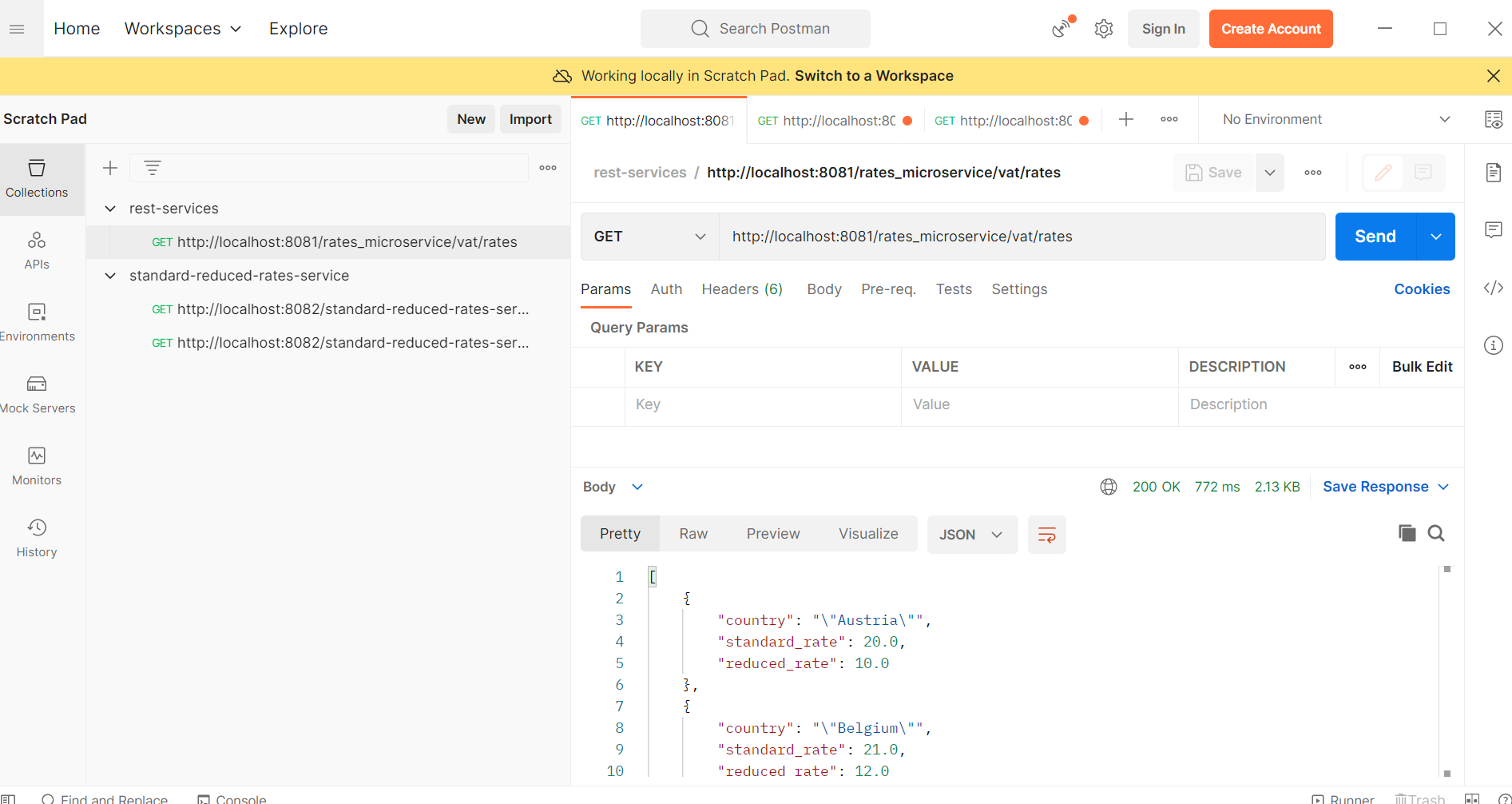
Create custom network after images creation



Run rate-service



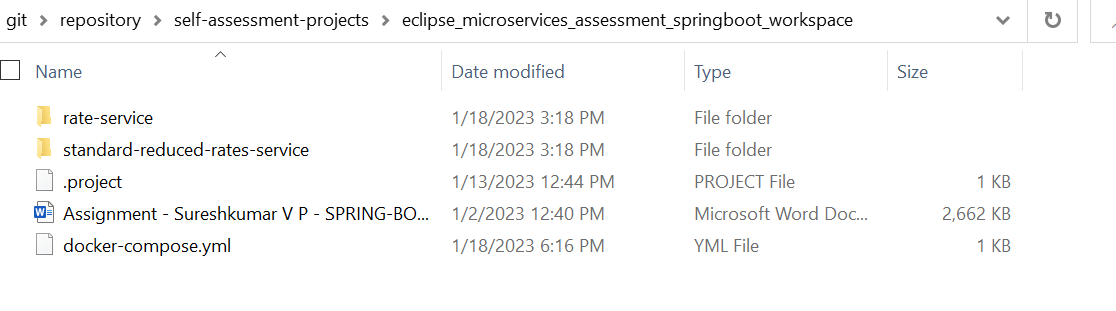
<http://localhost:8081/rates_microservice/vat/rates>



DOCKER COMPOSE

-------------- (make sure build the images in docker)

1. create docker-compose.yml in outside of projects



docker-compose.yml

version: '2.13.0'

services:

rate-service:

image: rate-service-tag/rate-service:dockerfilerates

ports:

- "8081:8081"

restart: always

networks:

- standard\_reduced\_rates\_network

standard-reduced-rate-service:

image: standard-reduced-rates-service-tag/standard-reduced-rates-service:dockerfilestandard-reduced-rates-service

ports:

- "8082:8082"

restart: always

depends\_on:

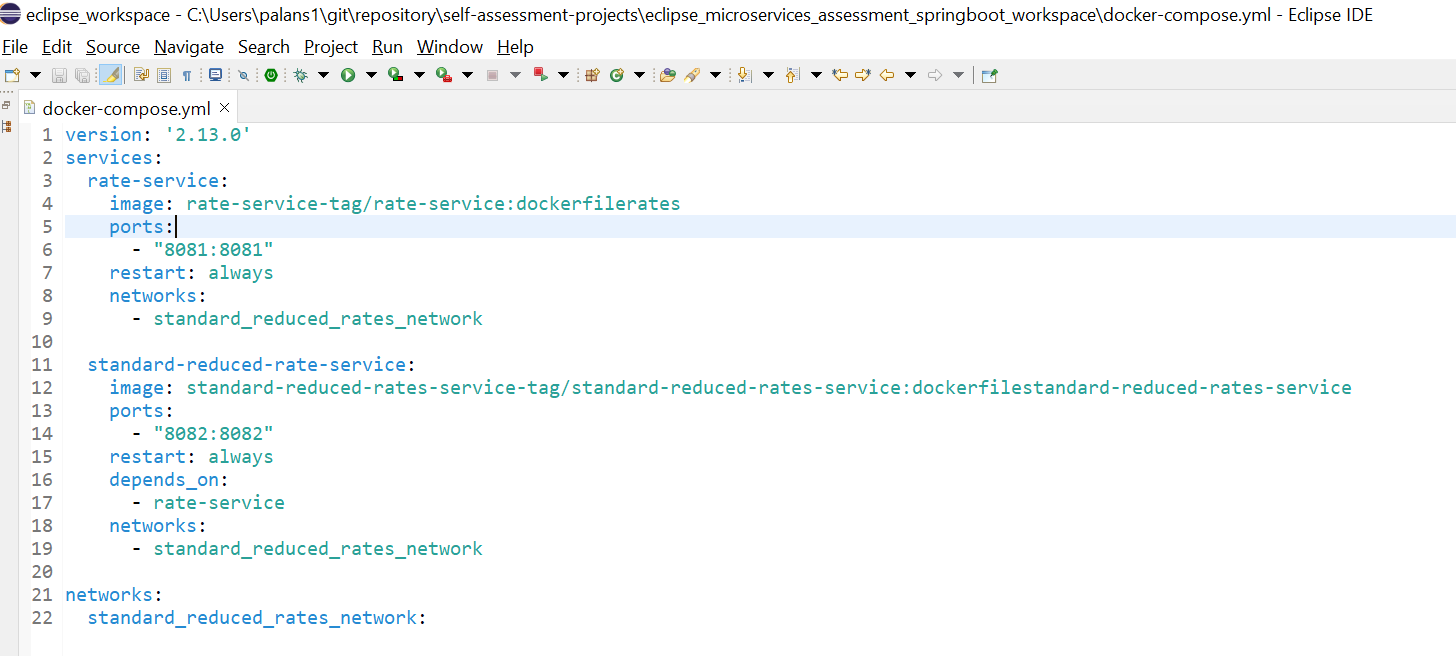
- rate-service

networks:

- standard\_reduced\_rates\_network

networks:

standard\_reduced\_rates\_network:



2. run the docker compose

- $ docker-compose up

3. run the application

**To avoid connection refused error use below steps of commands in docker:**

ISSUE-1: -> when connecting msg office network, localhost URL submit which invokes external end point point it throws SSL Handshake exception  
SOL: -> Switch to msg internet network then this issue resolved

ISSUE-2: -> when running the application to communicate between localhost services via docker it throws Connection Refused error  
SOL: -> Below points should take care to resolve it

1. After creating docker image of service1 (rateservice with port 8081) --please note the image name for mapping to calling from others services  
2. Run the docker run command for service1 (Ex:rateservice)   
   docker run -p 8081:8081 --name rateservice --net standard\_reduced\_rates\_network rate-service-tag/rate-service:dockerfilerates  
   --make sure docker execution command orders like flag, port, service name, then image name  
3. In service2 (standardservice with port 8082) update service1 name in the URL of calling service   
   --instead of localhost it should be [http://rateservice:8081/<service1](http://rateservice:8081/%3cservice1) context path name>  
4. After creating docker image of service2, run the docker run command for service2  
   docker run --net standard\_reduced\_rates\_network -p 8082:8082 --name standardservice  standard-reduced-rates-service-tag/standard-reduced-rates-service:dockerfilestandard-reduced-rates-service  
   --make sure the docker execution command orders like network name, port, service name, then the image name  
**Note**:- Both service should be in same network, we can create new  custom network and use it in docker run command

GIT

1. Go to local path where to create new git local repo, execute the command line as

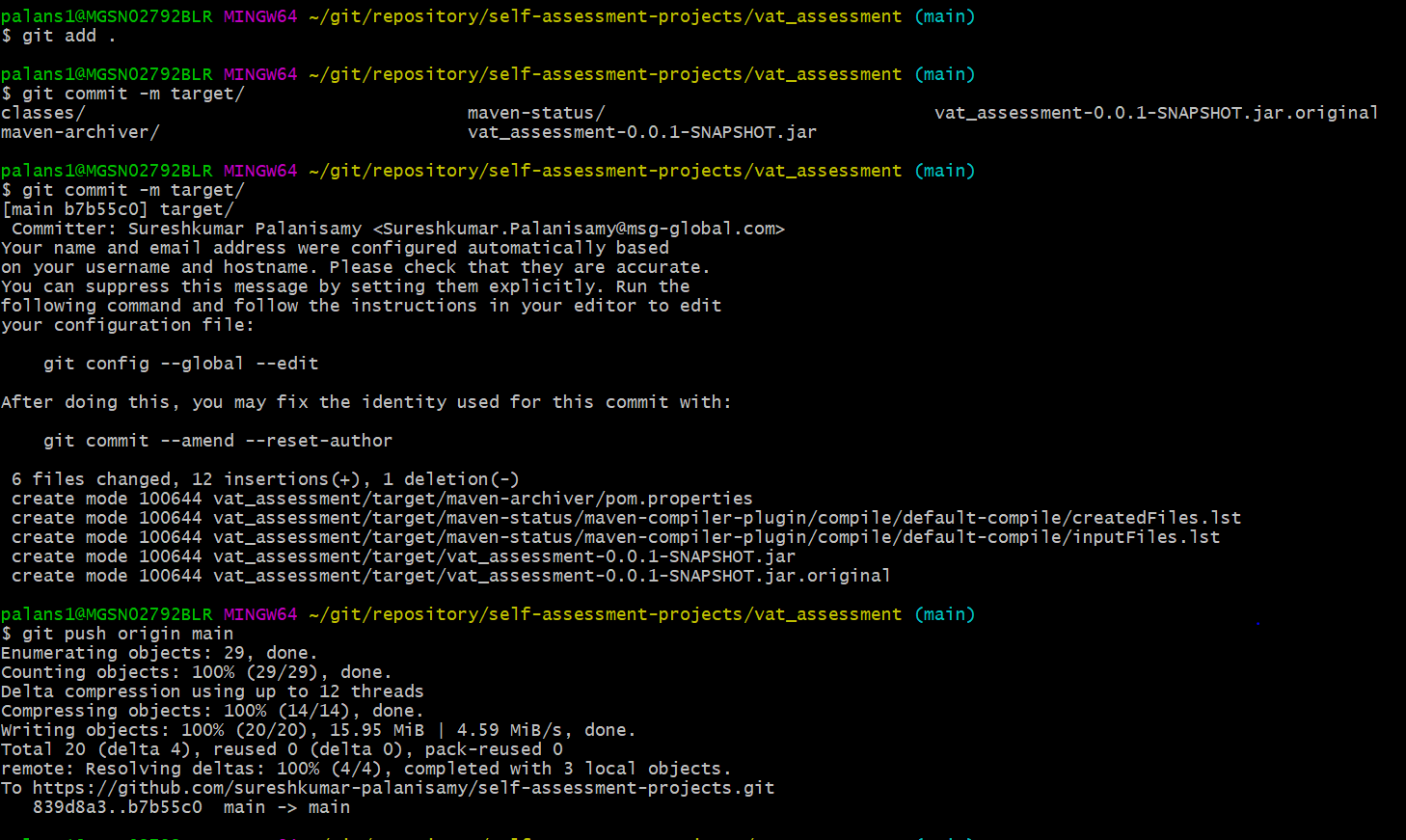
$ git clone [https://github.com/<username>/self-assessment-projects.git](https://github.com/%3cusername%3e/self-assessment-projects.git)

1. Once local git repo created then in eclipse import as general project and select from local projects to import
2. After import success, then select the project and convert into maven project and doe the maven update
3. If any modification done in the project then to commit follow below commands, go to the project folder do the git bash

$ git add .

$ git commit -m target/ (Ex: changed folder or file name)

$ git push origin main



Check the changes reflects in git hub

