* **Docker Assignment Submission Document**  
    
  Here I am using Docker in WSL Ubuntu machine   
  I have created a Folder called “docker-context “ and there I have written my web application In Java ,and Created a Dockerfile there   
  The folder structure in ubuntu looks like in the below image   
  A screenshot of a computer

  Description automatically generated  
  I Created the Simple Web Application with Java Springboot and the source code is in calculator folder .

**Docker File**

# -------------------- Build Stage --------------------

# Use a valid Maven official image with OpenJDK 11

FROM maven:3.6.3-openjdk-11 AS build

# Set the working directory inside the container

WORKDIR /app

# Copy the Maven project file (pom.xml) and download dependencies

COPY calculator/pom.xml .

# Copy the entire project source code into the container

COPY calculator/src ./src

# Package the application into a JAR file

RUN mvn clean package -DskipTests

# -------------------- Runtime Stage --------------------

# Use OpenJDK image for running the application

FROM openjdk:11-jre-slim

# Set the working directory inside the container

WORKDIR /app

# Copy the JAR file from the build stage to the runtime stage

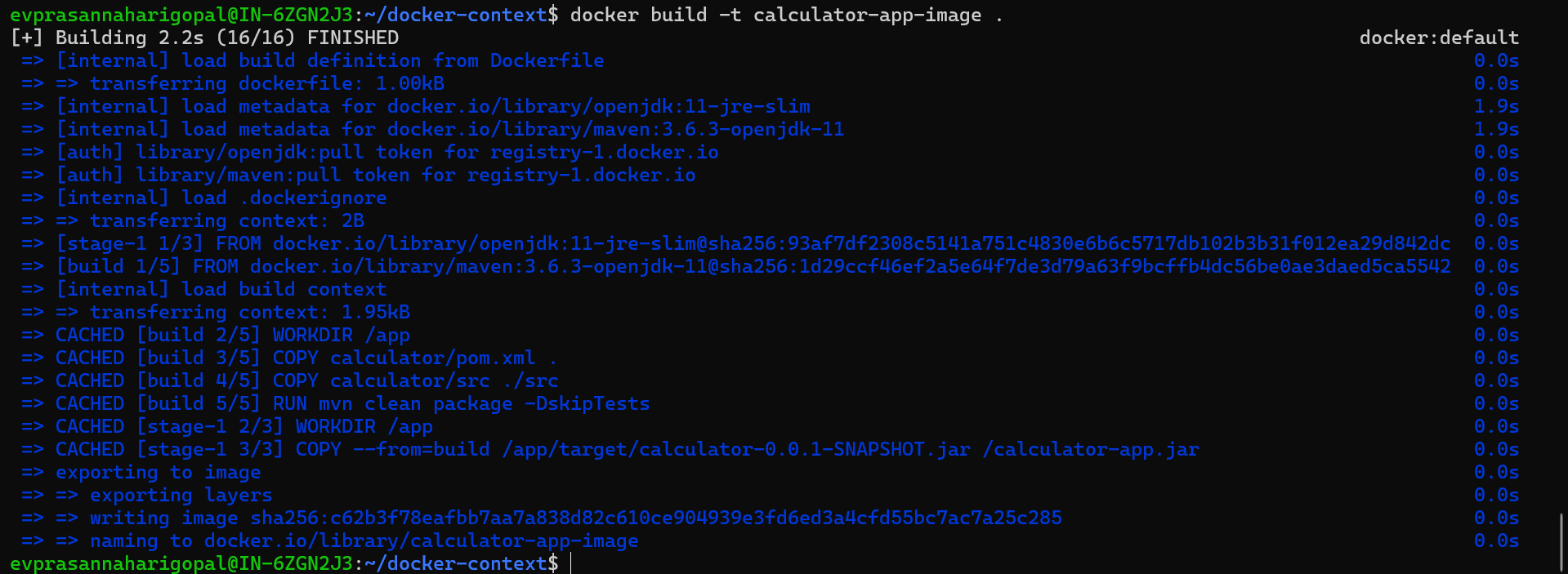
COPY --from=build /app/target/calculator-0.0.1-SNAPSHOT.jar /calculator-app.jar

# Expose the application port (change if necessary)

EXPOSE 8080

# Define the command to run the application

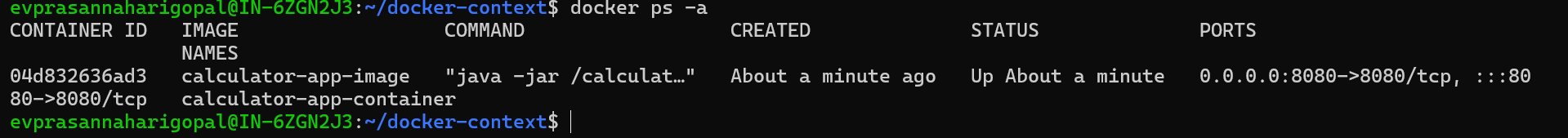
ENTRYPOINT ["java", "-jar", "/calculator-app.jar"]

**Build A Docker Image with Dockerfile**

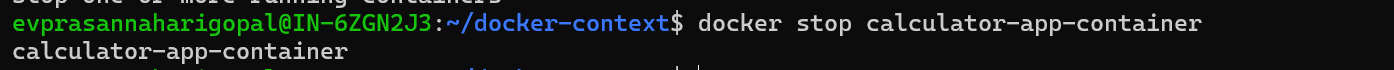
* **Create and run the Container using docker image**

**List of commands to show the containers**

**docker ps -a**

Lists the container  
   


**docker stop <container-name>**

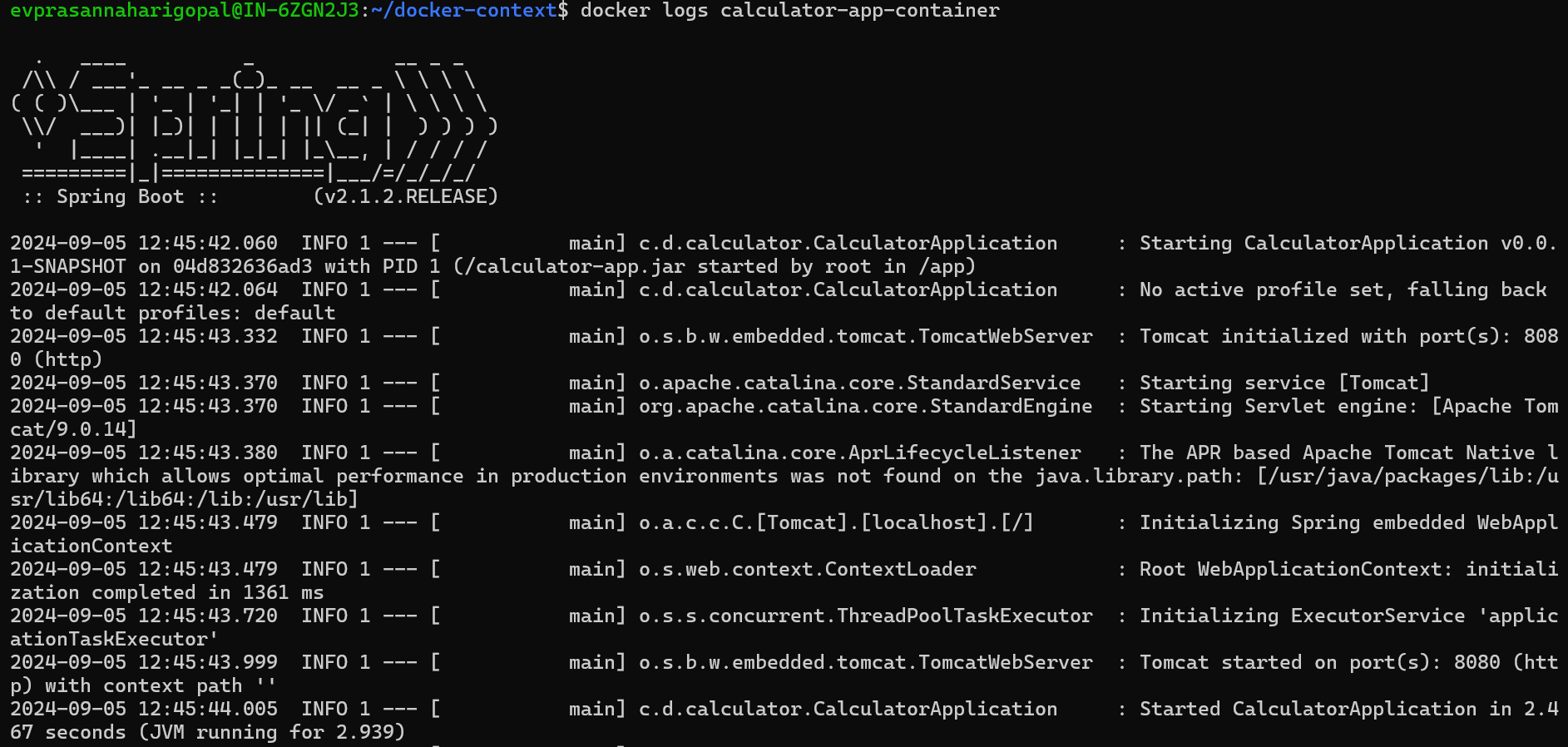
used to stop the container****

**docker start <container-name>**

Used to start the docker container  
A black screen with white text

Description automatically generated

**docker logs <container-name>**

Used to display the logs of that container

**Docker Multicontainer Application**

For this I have created A new folder “docker-multicontainer”, and there I have written my web application In Java ,and Created a “Dockerfile” and “docker-compose.yml”

The folder structure looks like this

A screenshot of a computer

Description automatically generated

**Docker-compose.yml**

This is my docker-compose.yml file

version: '3.8'

services:

calculator-container-1:

build: .

ports:

- "8081:8081"

calculator-container-2:

build: .

ports:

- "8082:8081"

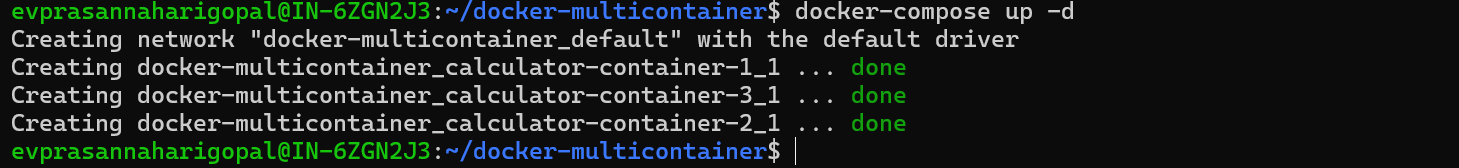
calculator-container-3:

build : .

ports:

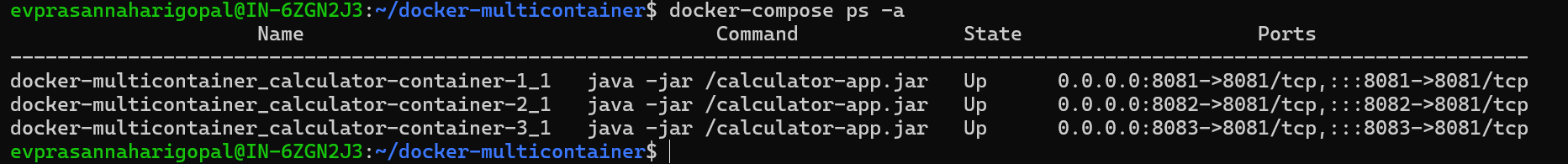
- "8083:8081"

Here the services are nothing but the containers ,I created multiple containers with single docker image and I ran them

I have created an image with **”docker-compose build “** command   
And now I am running the docker multi containers at once with   
**“docker-compose up -d “**  
here -d indicated detatched mode ,means the containers will run in the background  


**“docker-compose ps -a “**

A black screen with white text

Description automatically generatedList out the running containers using docker-compose ps -a****Tested the container services by hitting the curl request

**Used Docker Hub to Store My image**

I have logged into my docker hub using credentials

**docker login -u “username”**

asks for password enter the password and hit enter  
You will be logged in with valid credentials   
A screen shot of a computer

Description automatically generated  
and created a image with “docker tag <image-name> <username/image-name: tag-name>”  
used docker push <username/image-name: tag-name>  
Then my image got pushed into repository(docker hub)

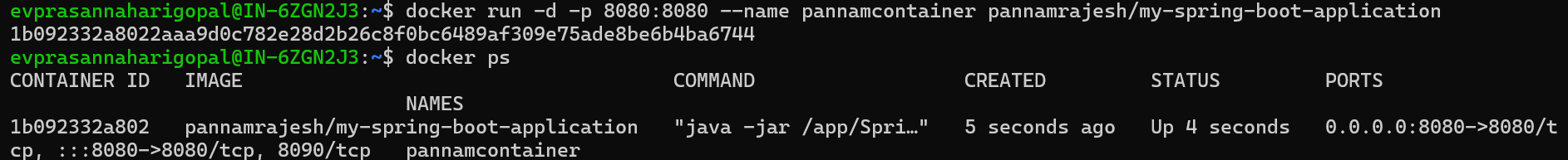
A black screen with white text

Description automatically generated

The screenshot is referring to the image in docker hub   
A screenshot of a computer

Description automatically generated

**Pull the image from Remote repository and run the container**

I take a pull of my friends application in which it was in public repository and I ran the container with that image   


**Bonus Of Assignment**

* Extend the multi-container application to include a database service and configure the web application to interact with the database.

Here I have created a new folder in Ubuntu named “docker-bonus” and I have placed my Source code application and “Dockerfile” and “docker-compose.yml”,Here I created a database and interacted with that database ,basically my application is calculator and I am storing the information like operand A and Operand B and Operation name and Result of that operation   
**Dockerfile**

# -------------------- Build Stage --------------------

# Use a valid Maven official image with OpenJDK 11

FROM maven:3.6.3-openjdk-11 AS build

# Set the working directory inside the container

WORKDIR /multicontainer

# Copy the Maven project file (pom.xml) and download dependencies

COPY calculator/pom.xml .

# Copy the entire project source code into the container

COPY calculator/src ./src

# Package the application into a JAR file

RUN mvn clean package -DskipTests

# -------------------- Runtime Stage --------------------

# Use OpenJDK image for running the application

FROM openjdk:11-jre-slim

# Set the working directory inside the container

WORKDIR /multicontainer

# Copy the JAR file from the build stage to the runtime stage

COPY --from=build /multicontainer/target/calculator-0.0.1-SNAPSHOT.jar /calculator-app.jar

# Expose the application port (change if necessary)

EXPOSE 8085

# Define the command to run the application

ENTRYPOINT ["java", "-jar", "/calculator-app.jar"]

**Docker-compose.yml**

version: '3.8'

services:

app:

image: calculator-app-bonus:latest

build:

context: .

dockerfile: Dockerfile

ports:

- "8085:8085" # Map to your Spring Boot server port

depends\_on:

- mysql

environment:

SPRING\_DATASOURCE\_URL: jdbc:mysql://mysql:3306/calculatordb

SPRING\_DATASOURCE\_USERNAME: root

SPRING\_DATASOURCE\_PASSWORD: root

SPRING\_JPA\_HIBERNATE\_DDL\_AUTO: update

mysql:

image: mysql:8.0

container\_name: mysqldb

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: calculatordb

ports:

- "3306:3306"

volumes:

- mysql\_data:/var/lib/mysql # Persisting the MySQL data

volumes:

mysql\_data: # Named volume for database persistence

**docker-compose build**

**A computer screen shot of blue text

Description automatically generated**

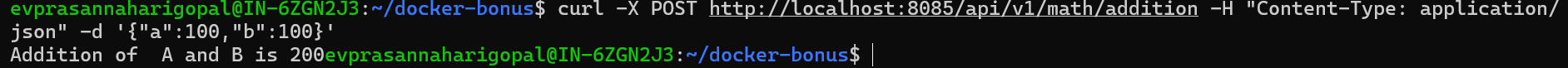
**A screen shot of a computer program

Description automatically generateddocker-compose up -d**

**docker-compose logs   
A screenshot of a computer program

Description automatically generated**

**Testing :**

****I send a Curl request ,The result is below

**  
  
Volume data**

A screen shot of a computer

Description automatically generated

**database information in volumes (mysql db)  
A screenshot of a computer

Description automatically generated**