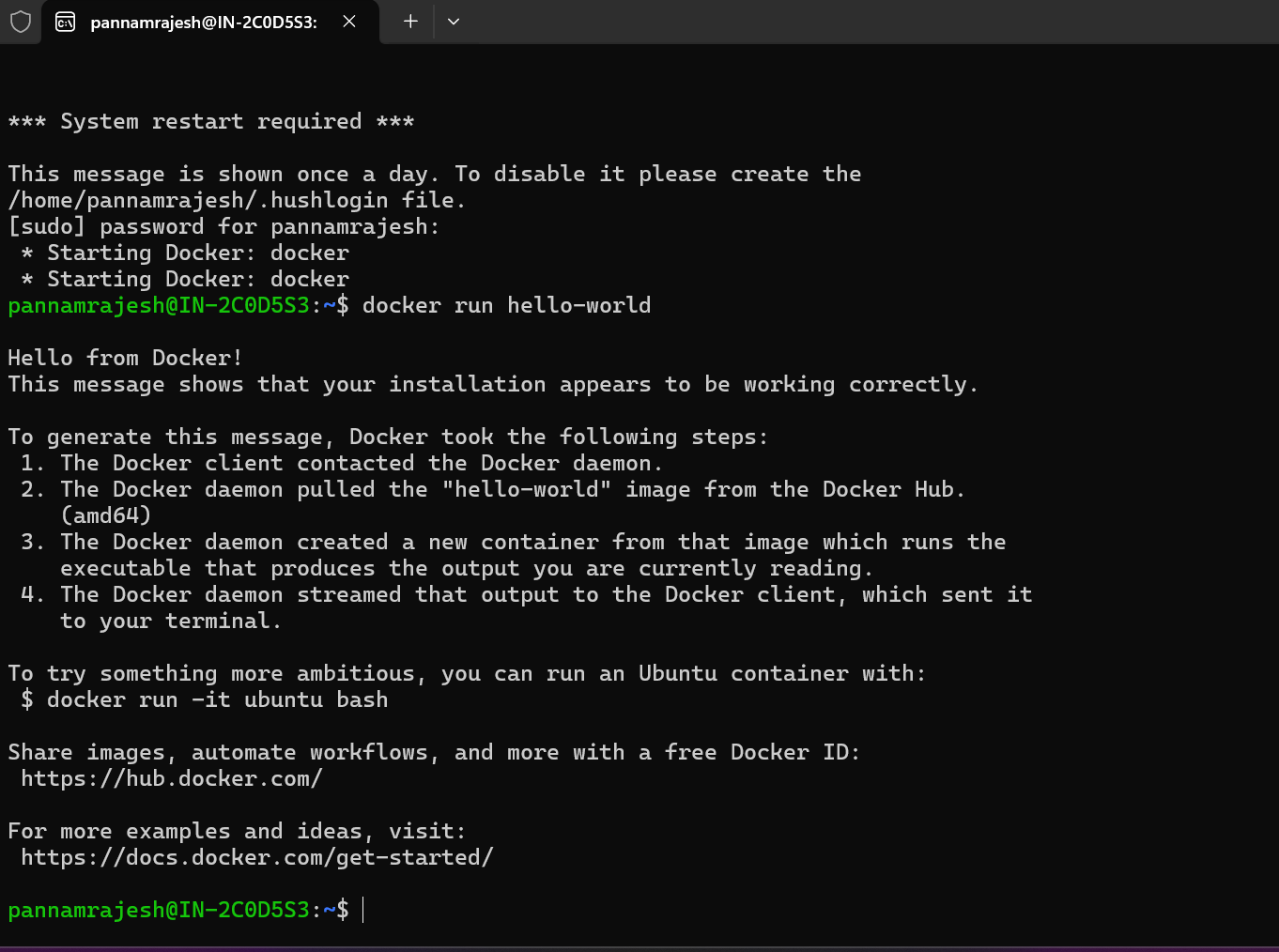
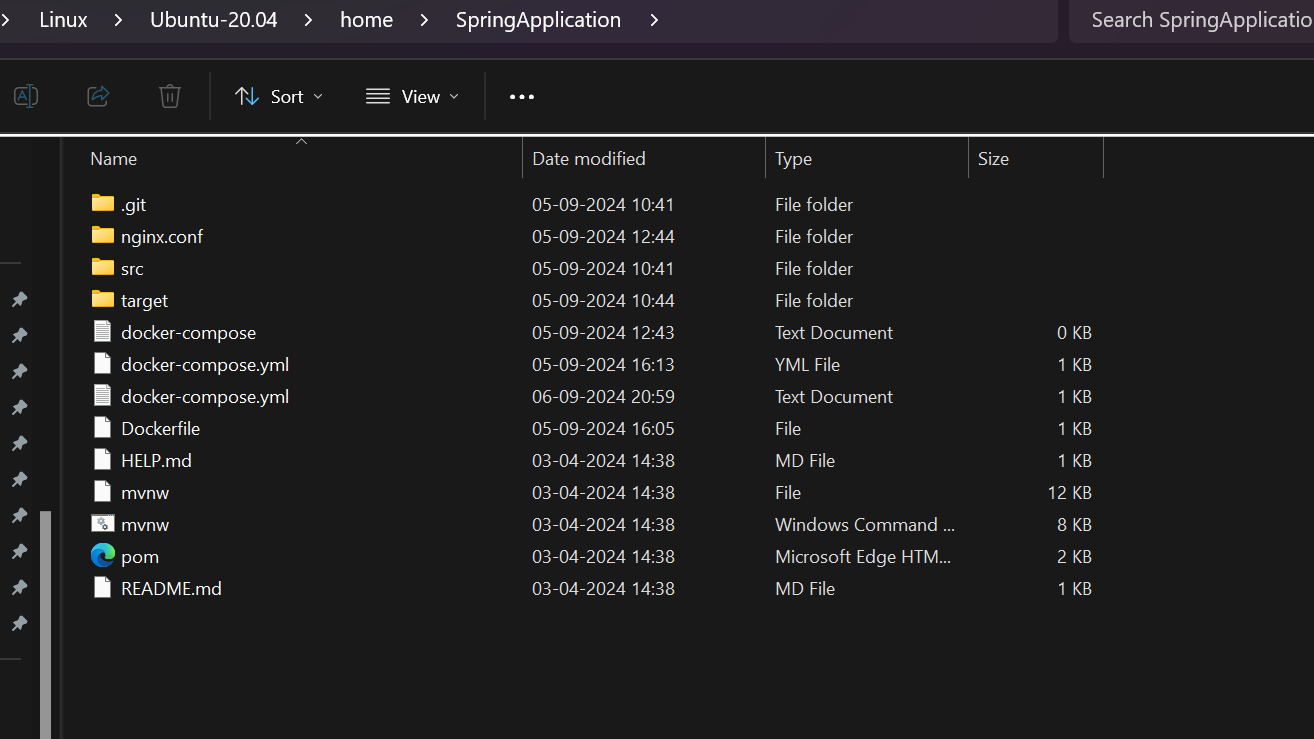
**Verify the installation by running a test container:**

**Docker Run** : The `docker run` command creates and starts a container from a specified Docker image. It pulls the image (if not available locally), creates the container, and executes the defined command inside it.

Here is a sample test container hello-world from docker.



**2. Create a simple web application :**

****

**3.** **Write a Dockerfile to containerize the application**

**Dockerfile :** A Dockerfile is a text file that contains a series of instructions to automate the process of building a Docker image. It defines the base image, environment setup, and commands needed to create the final image, such as installing dependencies, copying files, and specifying the container's default behavior

# Use OpenJDK 18 as the base image

FROM openjdk:18

# Expose port 8090 to allow external access to this port

EXPOSE 8090

# Copy the Spring Boot JAR file from the target directory to the /app directory in the container

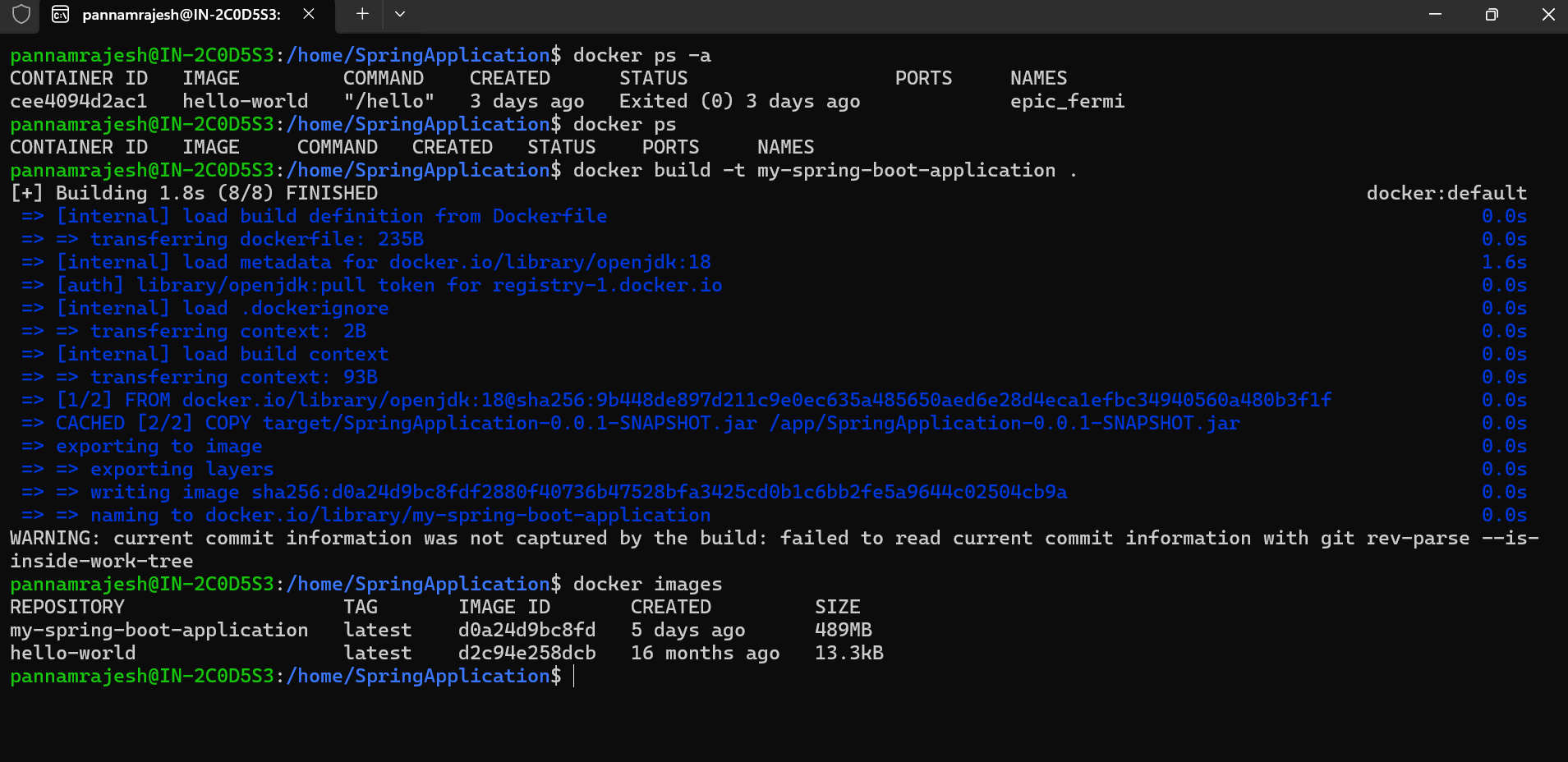
COPY target/SpringApplication-0.0.1-SNAPSHOT.jar /app/SpringApplication-0.0.1-SNAPSHOT.jar

# Define the command to run when the container starts

# This runs the Spring Boot application using the JAR file

ENTRYPOINT ["java", "-jar", "/app/SpringApplication-0.0.1-SNAPSHOT.jar"]

**4.** **Build the Docker image and run a container from it.**

****

**5. Use Docker commands to list, start, stop, and remove containers**

**Ans :**

**List Running Containers:**

docker ps

**List All Containers:**

docker ps -a

**Start a docker Container:**

Docker start container name or id

**Stop a docker Container:**

docker stop <container-id-or-name>

**Remove a docker Container:**

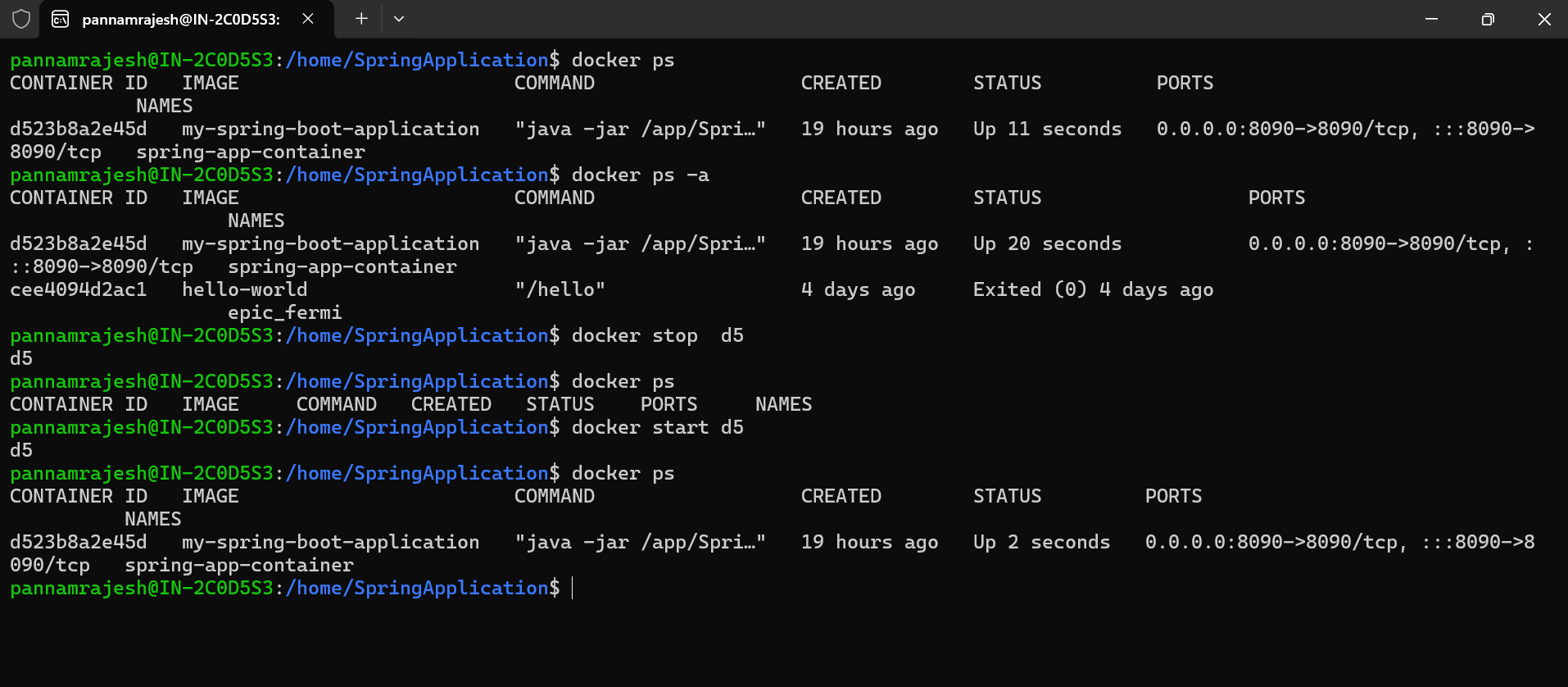
docker rm <container-id-or-name>

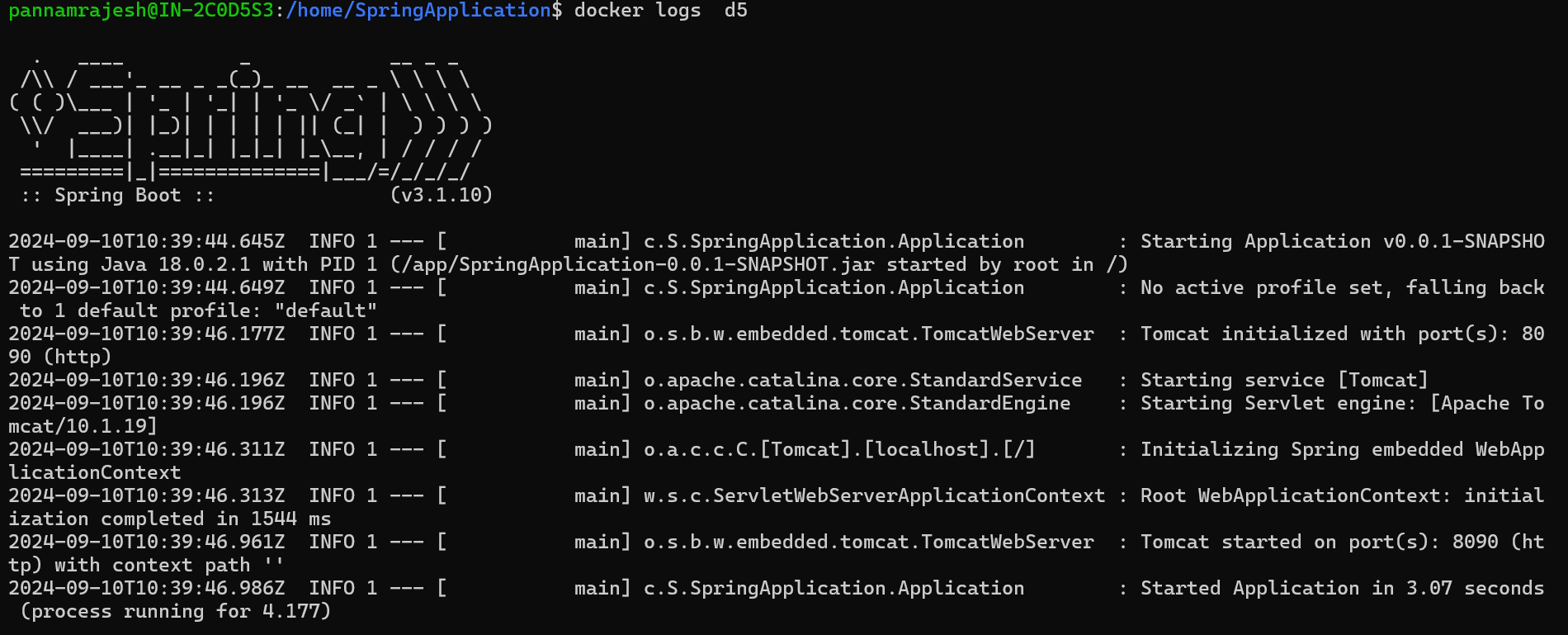
**Force Remove a docker Container:**

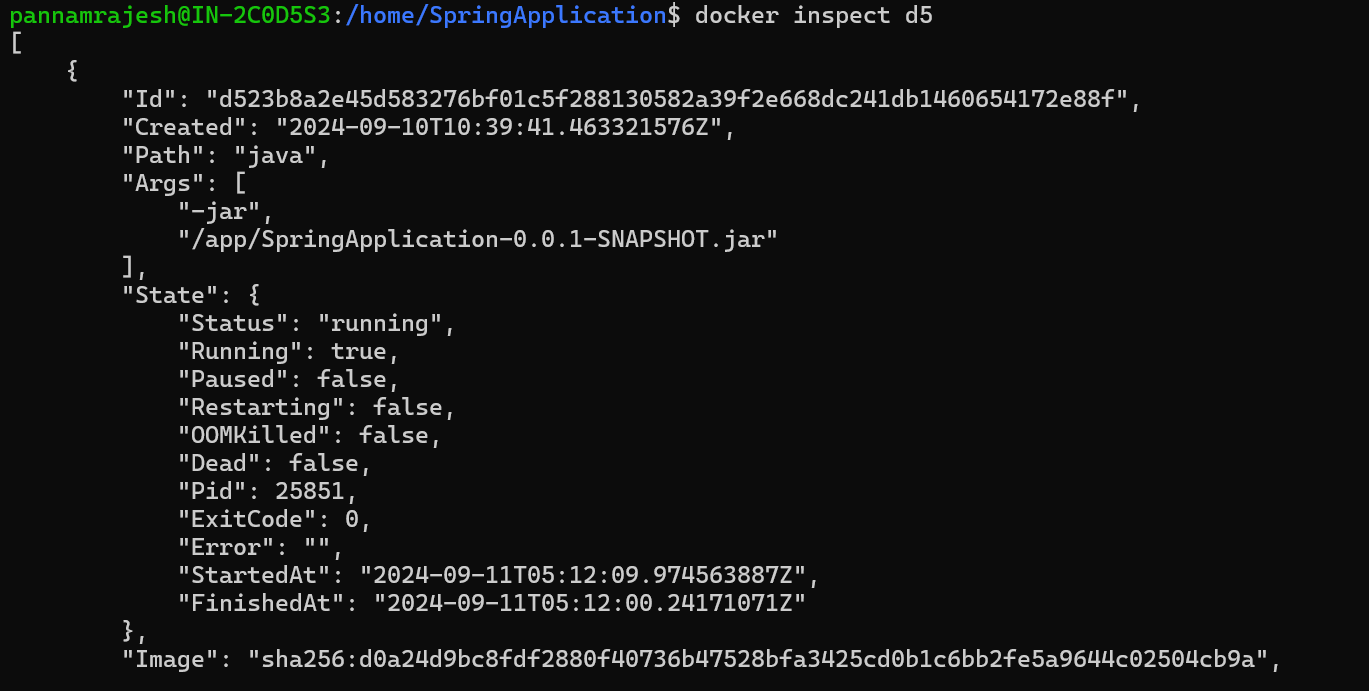
docker rm -f <container-id-or-name>

**Remove All Stopped Containers:**

docker container prune

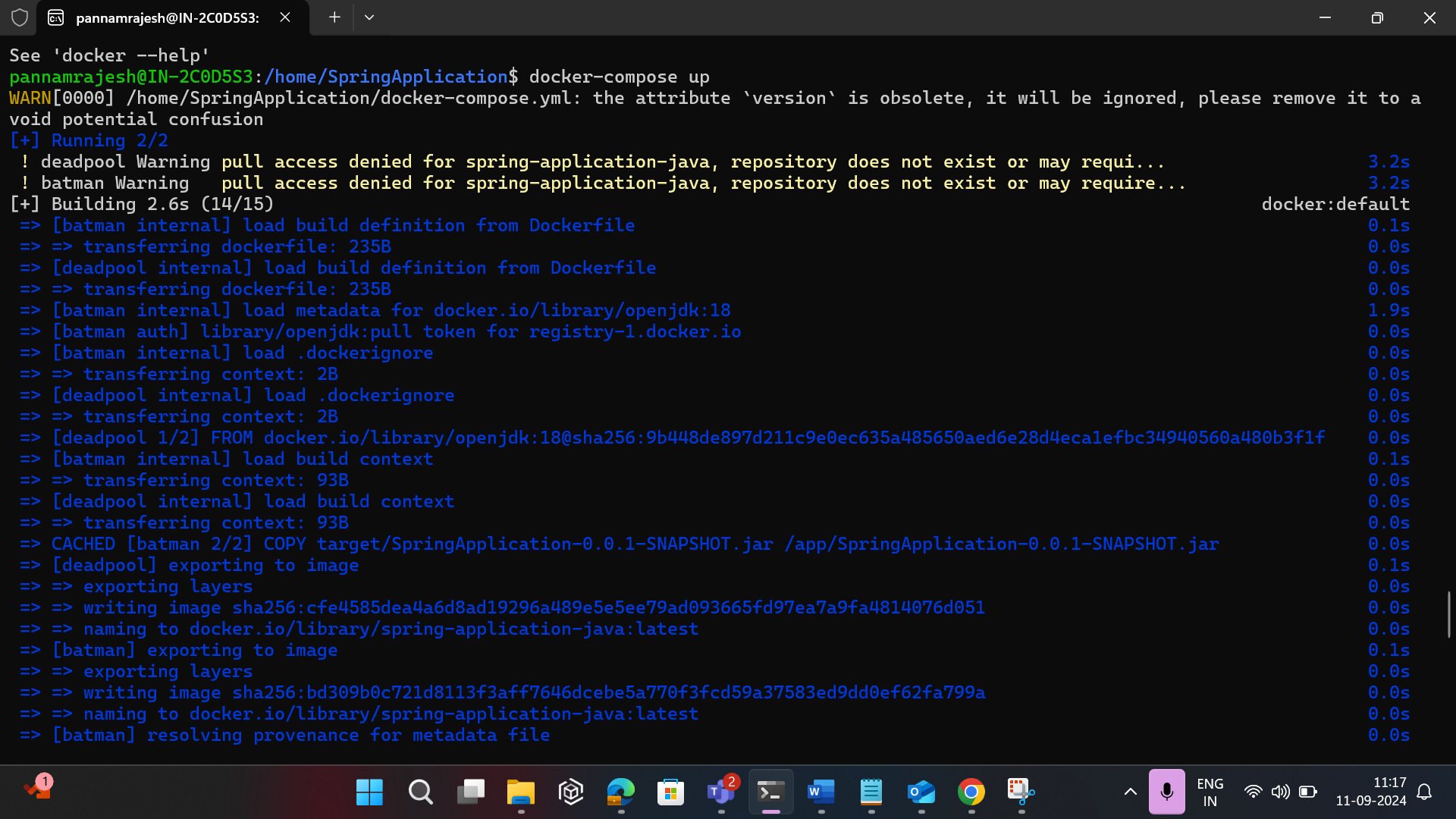


**6. Inspect running containers and view logs.**

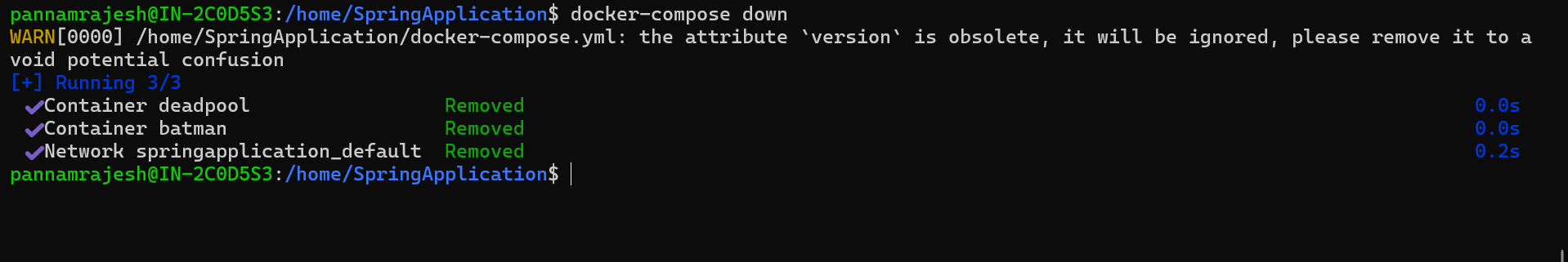
****

**7.** **Write a docker-compose.yml file to define a multi-container application**

Docker-Compose up : to run multiple containers

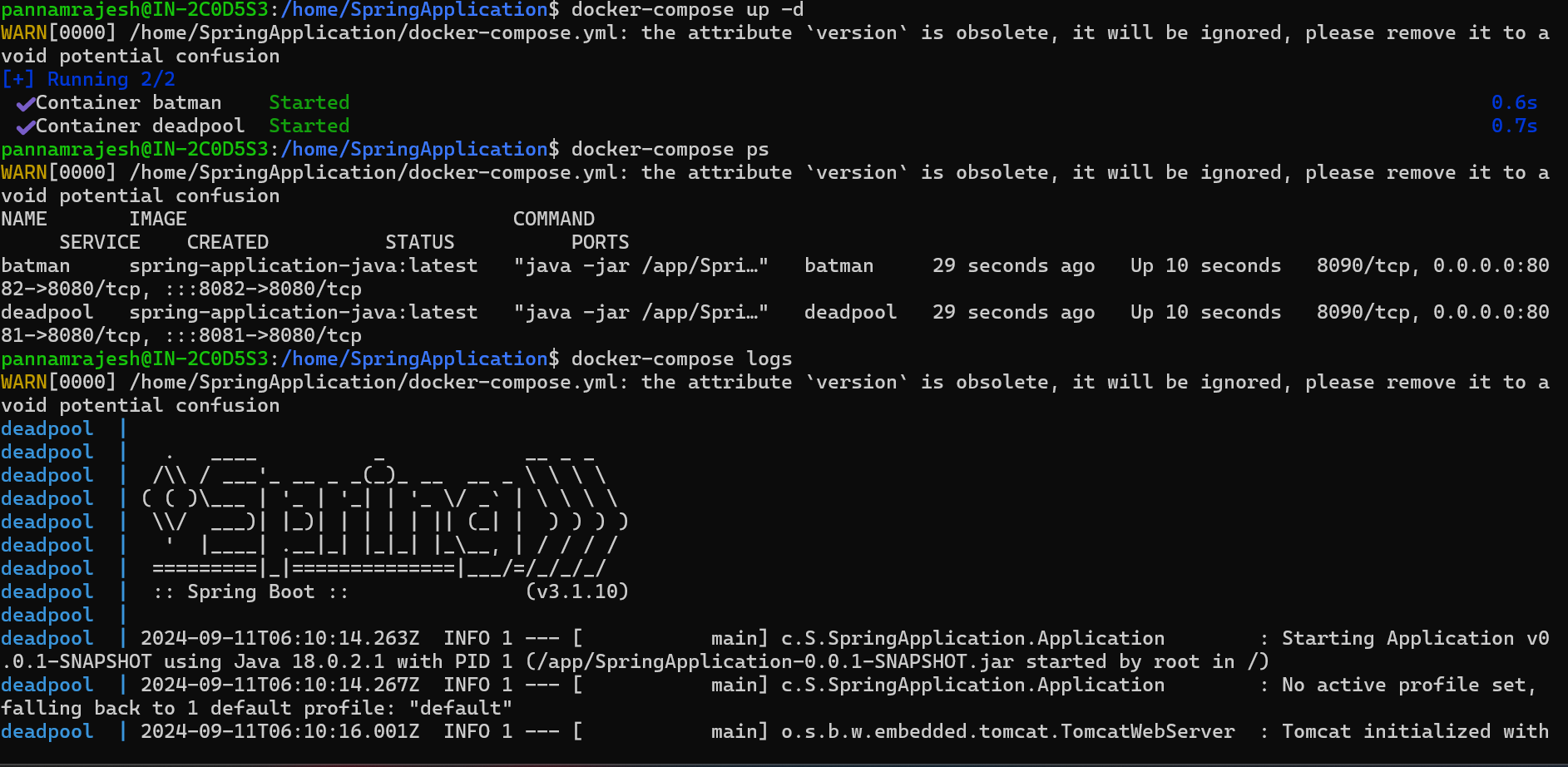


**Docker-Compose down** : to stop Containers

****

**8 )Use Docker Compose to bring up the application and ensure all services are running correctly.**

**Docker-compose logs** to check all the logs ensure they are running without errors

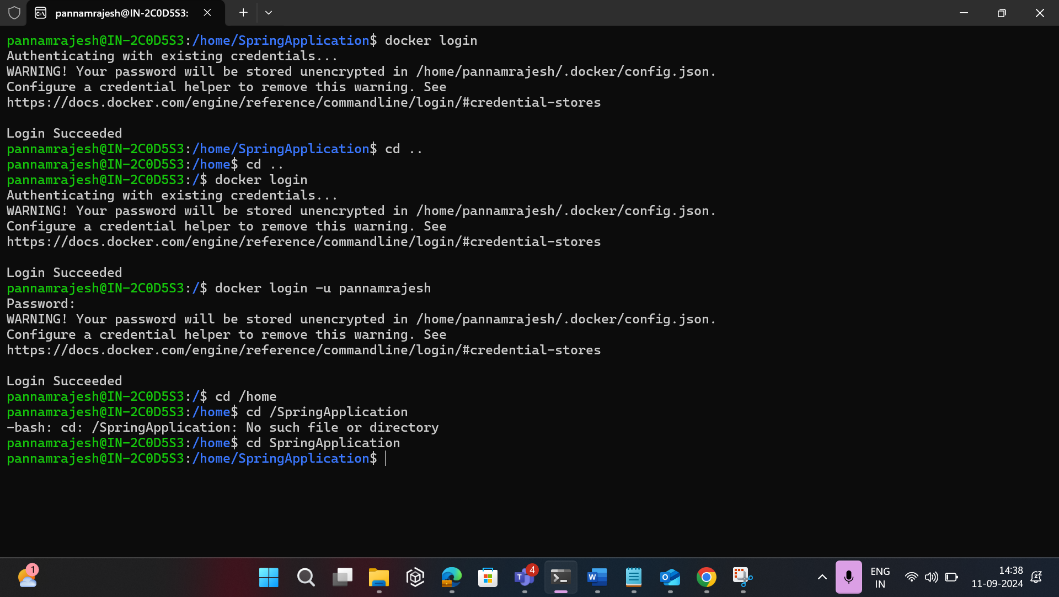
****

**To view logs in real-time:**

docker-compose logs -f

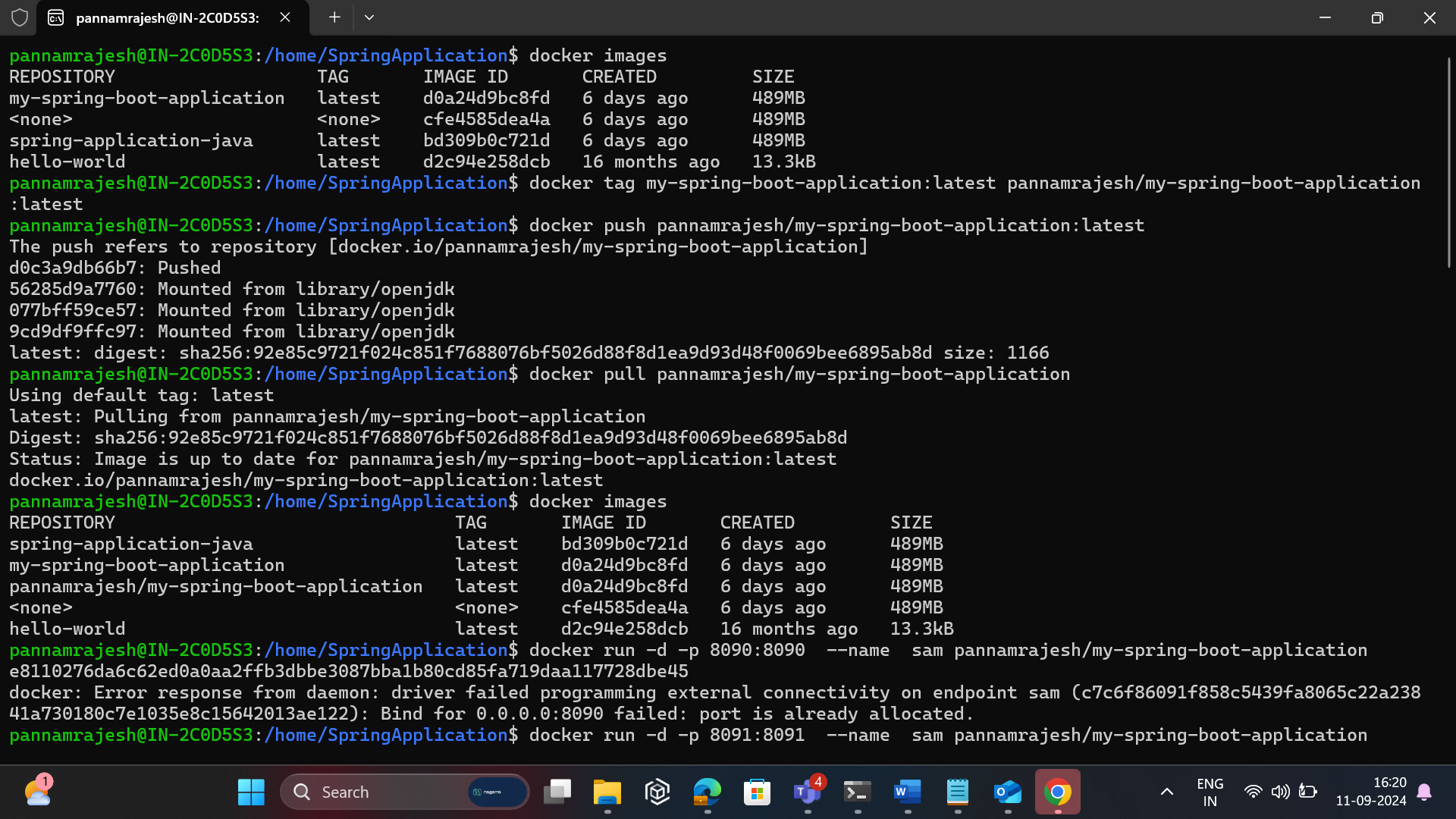
**9 Create a private Docker registry or use Docker Hub.**

Docker login -u <Username> password : provide username and password to login to docker hub.

****

**10. Push your Docker images to the registry.**

**And**

**Pull the images from the registry and run them locally.  
**