**ASSIGNMENT-2**

**Q1) Pull any image from the docker hub, create its container, and execute it showing the output.**

**Ans:**

**Docker Hub:**

Docker Hub is the world’s largest repository of container images and it allow us to share container images with our team,customers or with community.It is cloud-based repository that lets us to create,test,store and deploy the container images.

**Docker Image:**

It is a kind of ready to use software and read-only template crafted with source codes,libraries,dependencies,tools and other files that are needed for the software application to run successfully on any platform or operating system.

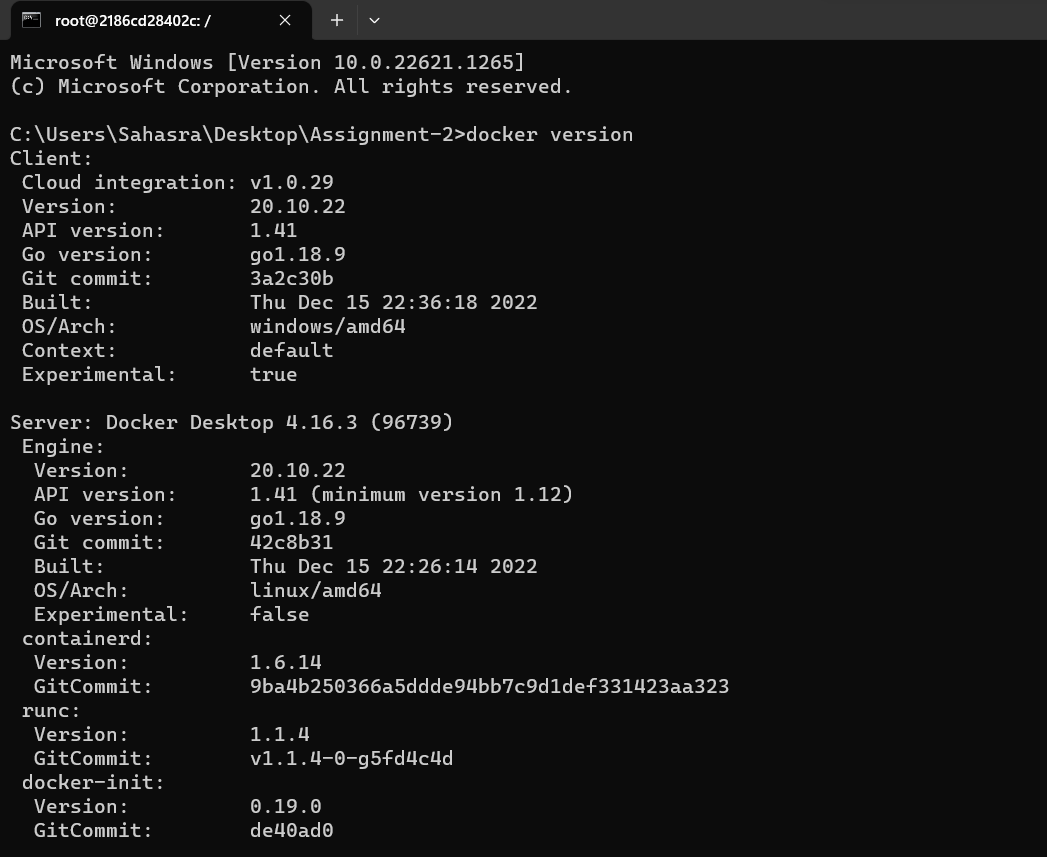
**Docker container:**

It is like a box which has the ability to run the docker images and it can be considered as cohensive software unit that contains code and all its dependencies so that application can run quickly and reliably.

**Pulling an image and executing it :**

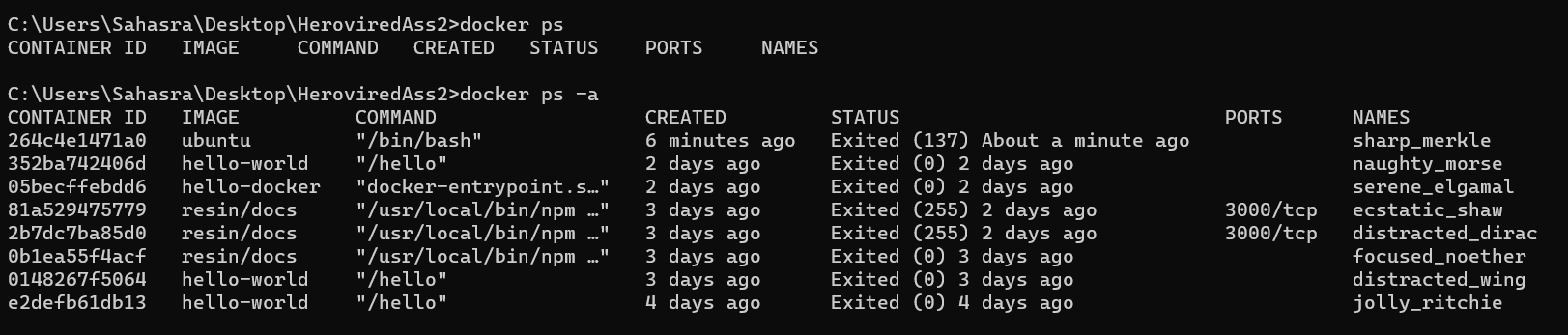
**Step 1 :**

First,we need to check the version of the docker.

****

**Step 2 :**

Check whether any container was in running state by using docker ps command.Docker ps command was used to list all the running containers and docker ps -a was used to list all the exited containers.

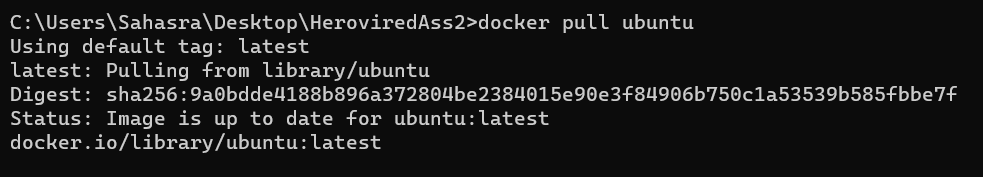
****

(Still now there is no containers in running state)

**Step 3 :**

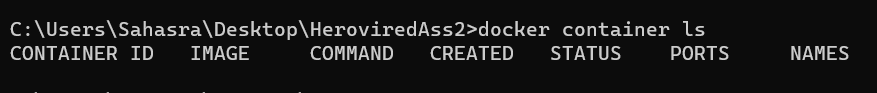
Now,we are pulling an image called Ubuntu from Docker Hub with the help of docker pull command.Docker pull command will the download the specified image from public repository(hub.docker.com).

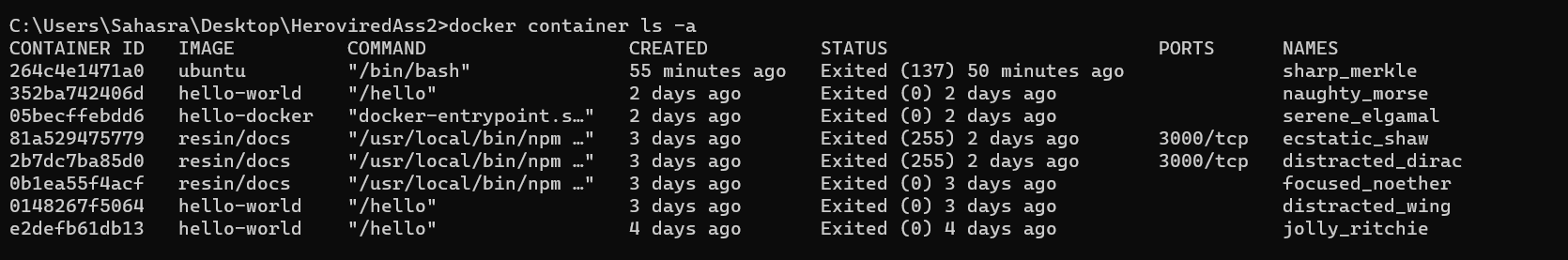
Docker pull <image\_name>

****

**Step 4 :**

After pulling the image,it doesn’t show that image when use command docker container ls.That means we need to create a container for the image that we have pulled .

****

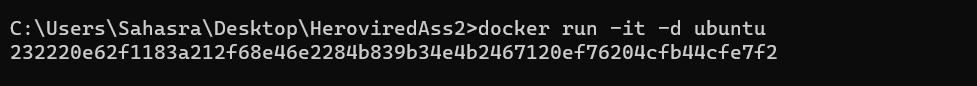
****

**Step 5 :**

To create a container for the pulled image,we can use docker run command.Docker run command will creates a writeable container layer over the specified image.

**Docker run -it -d <image\_name>**

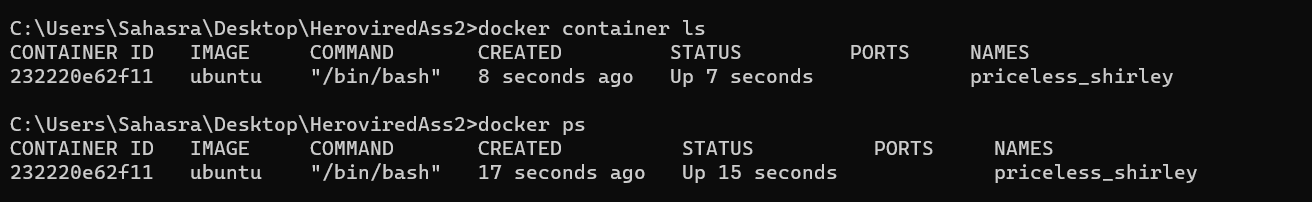
It will create a container for the specified image

****

**Step 6 :**

Now,we can see a container with ID 232220e62f11 of ubuntu image was in the running state.We can list the running containers using docker container ls or docker ps command.

docker container ls or docker ps

****

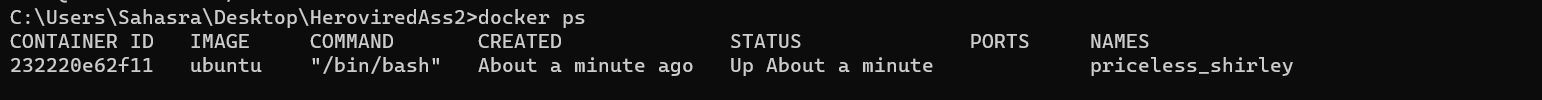
**Step 7 :**

We can execute the container with the help of the command docker exec.Docker exec was used to runs a new command in a running container.

Docker exec -it <container\_id> bash

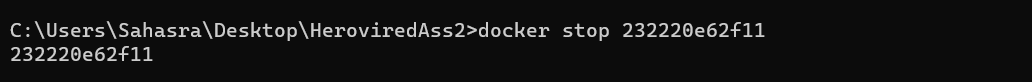
After doing the above command,it will enter into the running container.

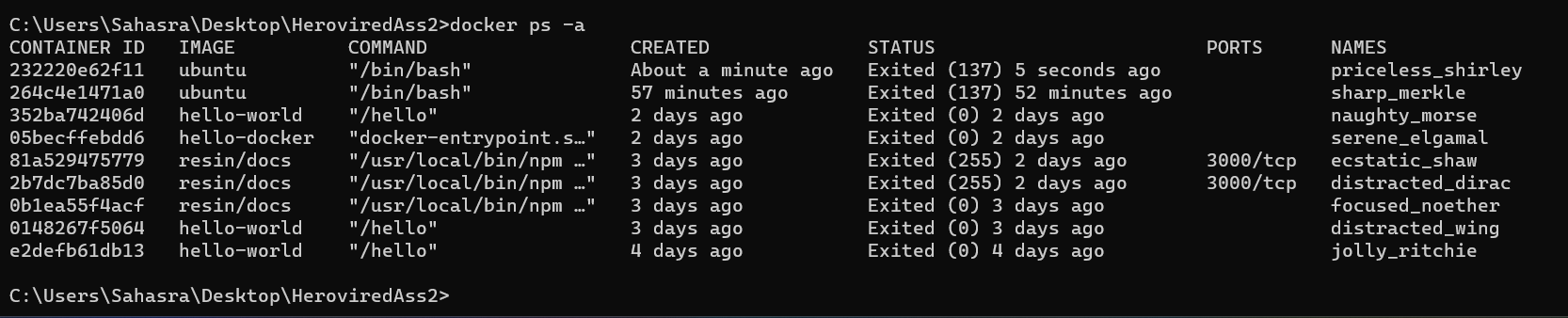
****

****

**Step 8 :**

And we can stop the container using docker stop.Docker stop command will stop the container.

****

****

**Q2) Create the basic java application, generate its image with necessary files, and execute it with docker.**

**Ans:**

**Docker Image:**

It is a kind of ready to use software and read-only template crafted with source codes,libraries,dependencies,tools and other files that are needed for the software application to run successfully on any platform or operating system.

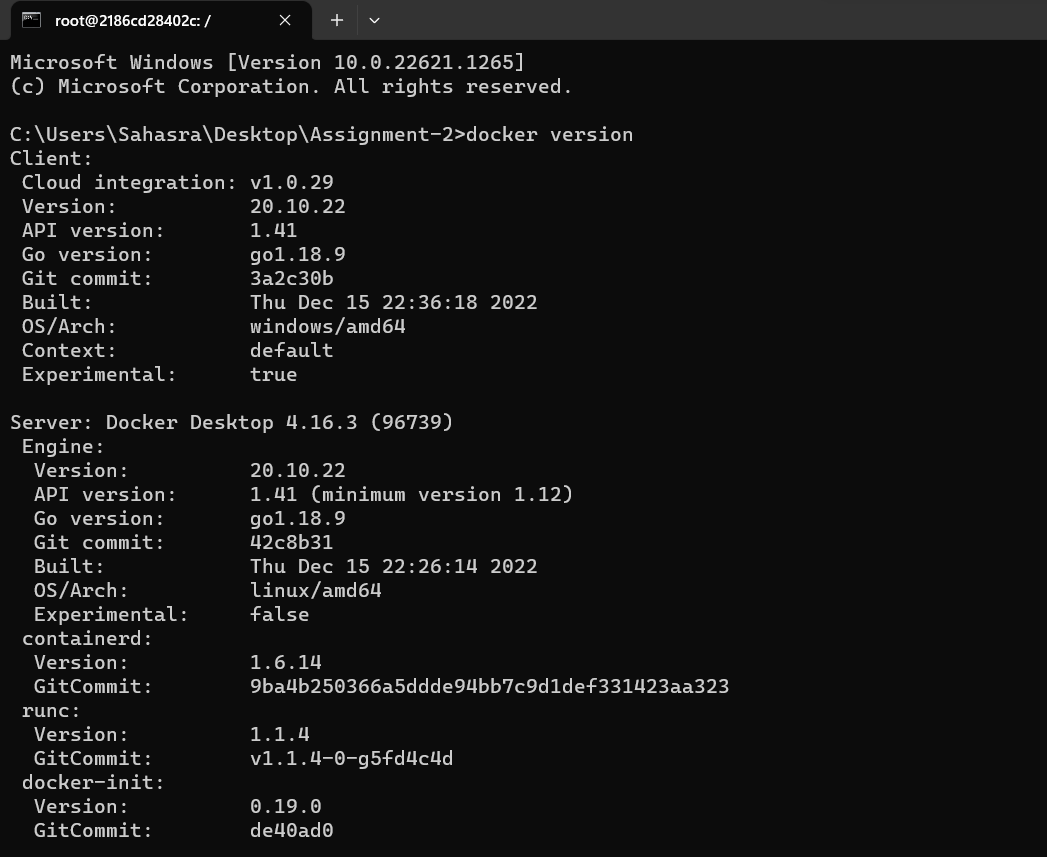
**Docker container:**

It is like a box which has the ability to run the docker images and it can be considered as cohensive software unit that contains code and all its dependencies so that application can run quickly and reliably.

Now,we are creating a java application and running by using the docker.

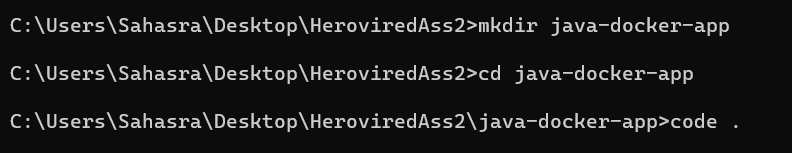
**Step 1 :**

First,we need to check the version of the docker.

****

**Step 2 : Creating a directory**

Now,we are creating a directory with the name of java-docker-app

****

**Step 3 :**

Now,open vscode and open java-docker-app.And now create a new java file Hello.java and also Dockerfile.Dockerfile is a simple text with the set of commands or instructions and it is a script that used the docker platform to generate containers automatically.

**Hello.java:**

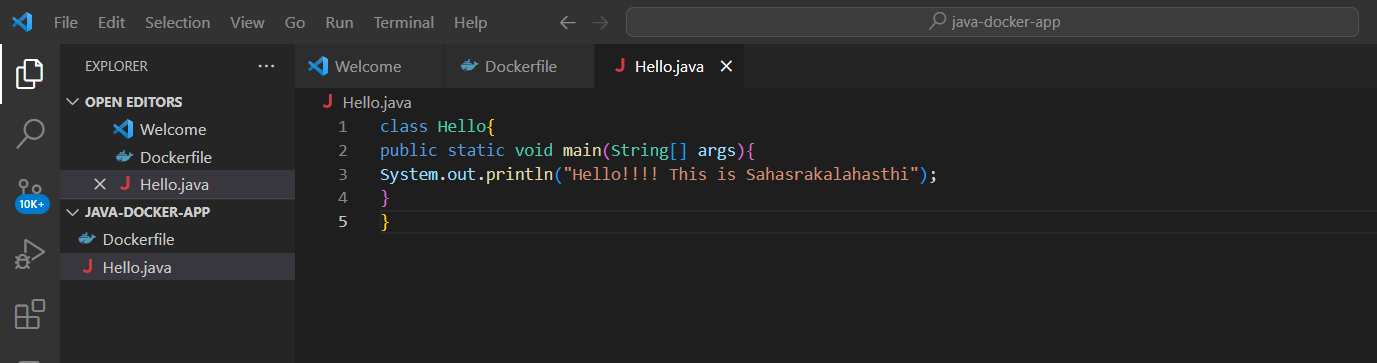
class Hello{

    public static void main(String[] args){

        System.out.println("Hello!!!! This is Sahasrakalahasthi");

    }

}

****

**Dockerfile:**

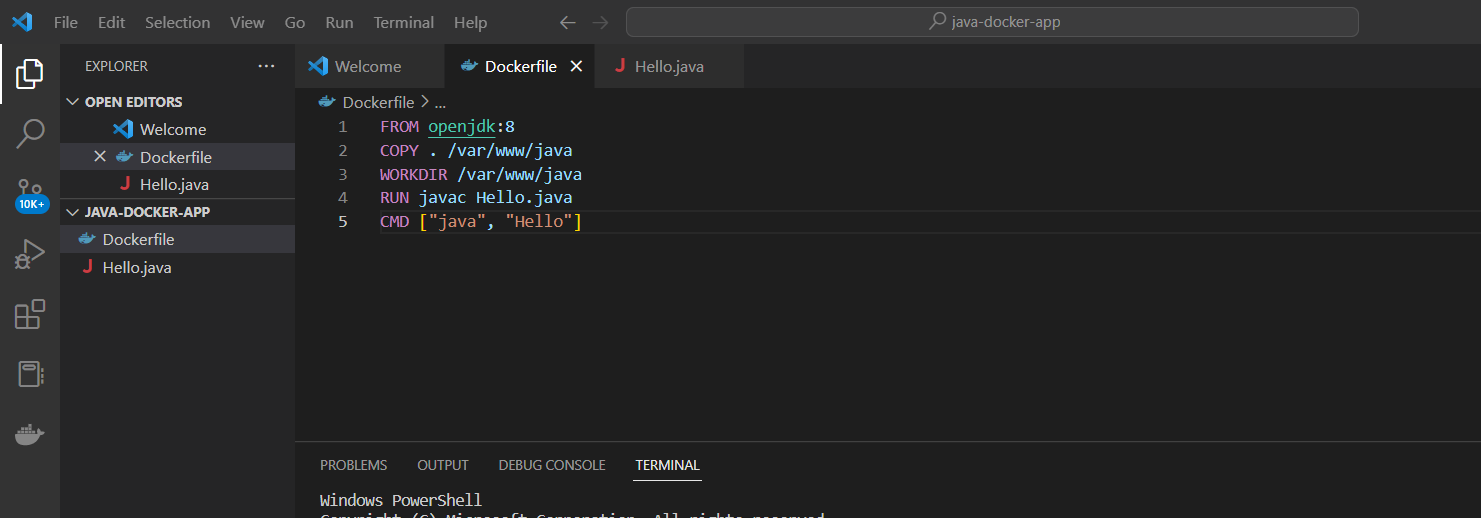
FROM openjdk:8

COPY . /var/www/java

WORKDIR /var/www/java

RUN javac Hello.java

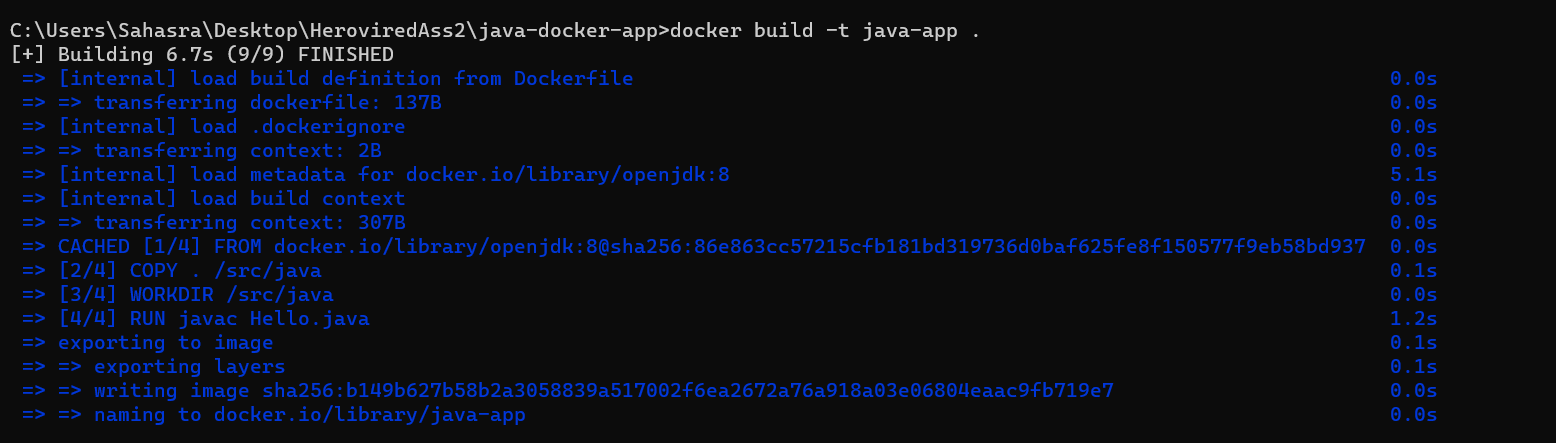
CMD ["java", "Hello"]

****

**Step 4 :**

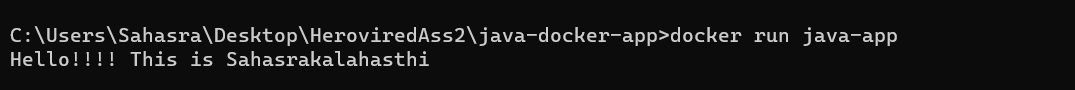
After creating a docker file,we are creating an image by using the command docker build.Docker build will create an image with the name given.

Docker build -t <image\_name> .

****

**Step 5 :**

Now,after creating an image successfully,we can run docker by using run command.docker run command will run the java application.It will show the output of the Hello.java file.

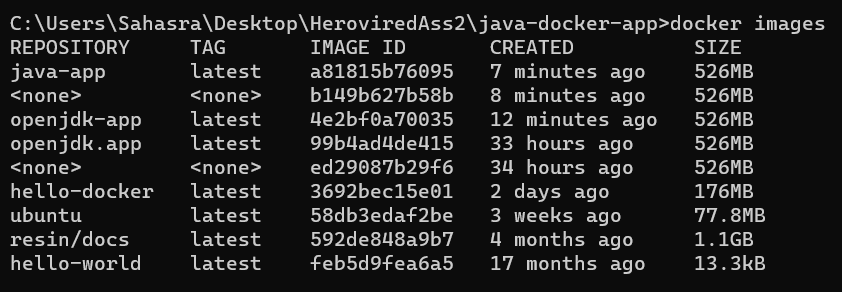
****

Here,we can see that after running the java-app it produced an output of “**Hello!!!!This is sahasrakalahasthi**”

**Step 6 :**

We can list the running containers using docker images or docker container ls.

Docker images or docker container ls

****

**Git repository link:**

https://github.com/sahasrakalahasthi/Herovired\_Assignments