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.

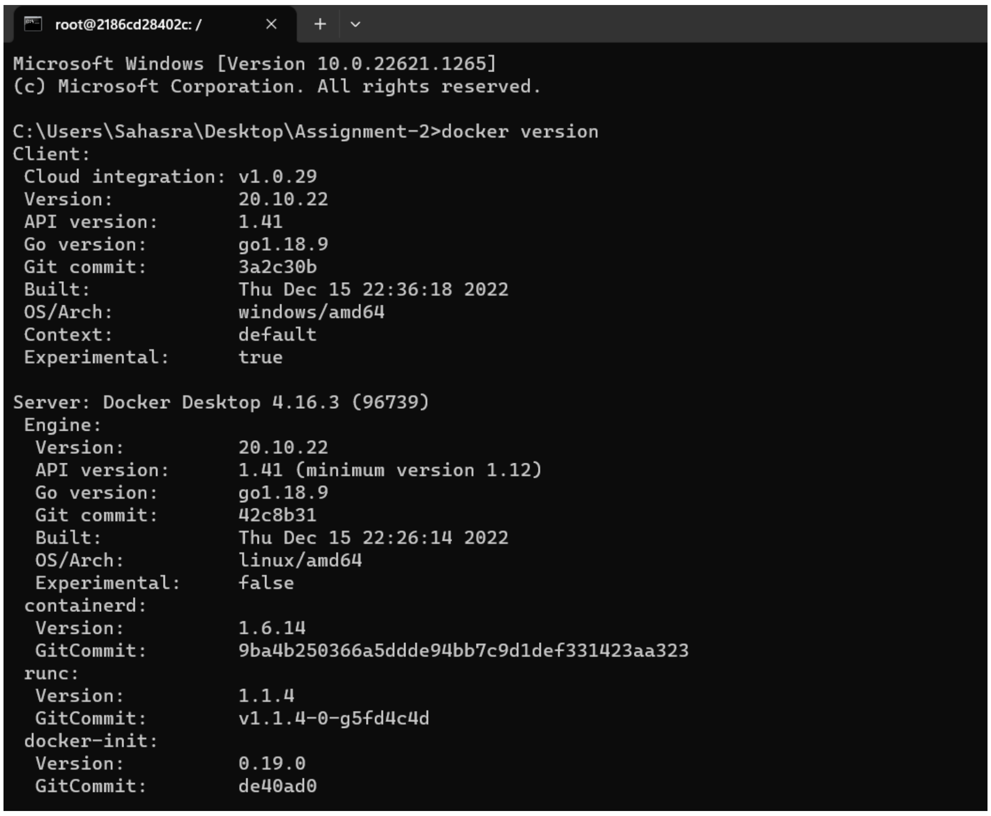
Docker container:

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 :

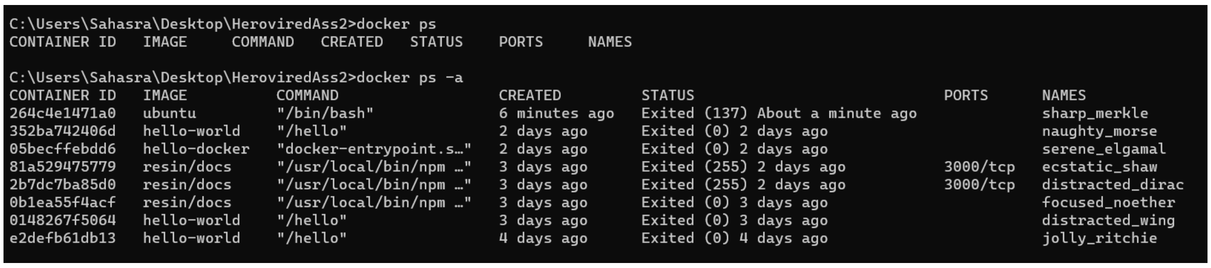
First,we need to check the version of the docker.



```
Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sahasra\Desktop\Assignment-2>docker version
Client:
 Cloud integration: v1.0.29
 Version:           20.10.22
 API version:       1.41
 Go version:        go1.18.9
 Git commit:        3a2c30b
 Built:             Thu Dec 15 22:36:18 2022
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Desktop 4.16.3 (96739)
 Engine:
  Version:          20.10.22
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.18.9
  Git commit:       42c8b31
  Built:            Thu Dec 15 22:26:14 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.14
  GitCommit:        9ba4b250366a5ddde94bb7c9d1def331423aa323
 runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

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.



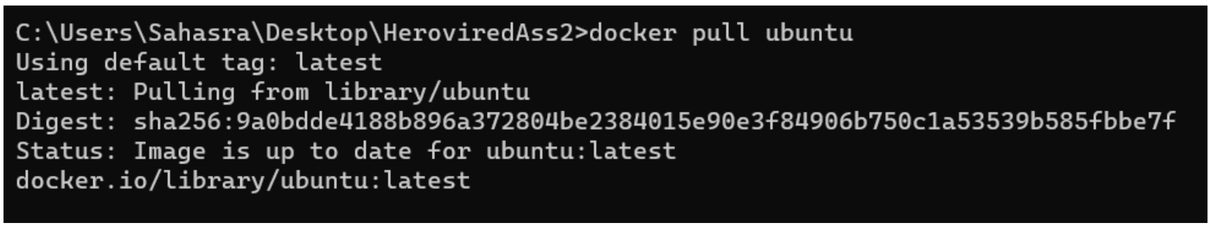
```
C:\Users\Sahasra\Desktop\HerovireddAss2>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

C:\Users\Sahasra\Desktop\HerovireddAss2>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS                              PORTS      NAMES
264c4e1471a0   ubuntu         "/bin/bash"              6 minutes ago   Exited (137) About a minute ago                sharp_merkle
352ba742406d   hello-world    "/hello"                 2 days ago      Exited (0) 2 days ago                          naughty_morse
05becffebdd6   hello-docker   "docker-entrypoint.s…"   2 days ago      Exited (0) 2 days ago                          serene_elgamal
81a529475779   resin/docs     "/usr/local/bin/npm …"   3 days ago      Exited (255) 2 days ago             3000/tcp   ecstatic_shaw
2b7dc7ba85d0   resin/docs     "/usr/local/bin/npm …"   3 days ago      Exited (255) 2 days ago             3000/tcp   distracted_dirac
0b1ea55f4acf   resin/docs     "/usr/local/bin/npm …"   3 days ago      Exited (0) 3 days ago                          focused_noether
0148267f5064   hello-world    "/hello"                 3 days ago      Exited (0) 3 days ago                          distracted_wing
e2defb61db13   hello-world    "/hello"                 4 days ago      Exited (0) 4 days ago                          jolly_ritchie
```

(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).

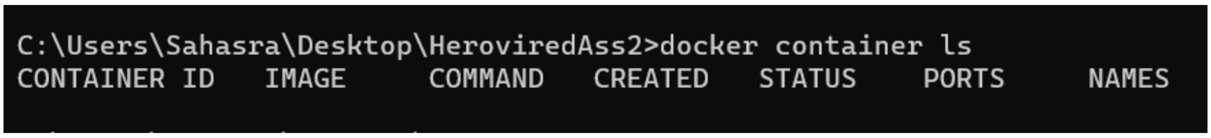
Docker pull <image_name>



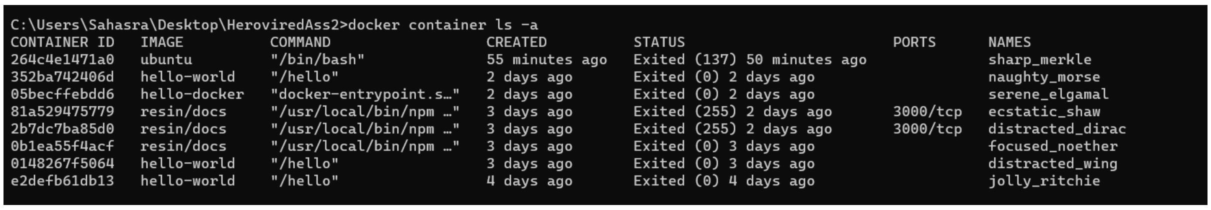
```
C:\Users\Sahasra\Desktop\HerovireddAss2>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:9a0bdde4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest
```

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 .



```
C:\Users\Sahasra\Desktop\HerovireddAss2>docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```



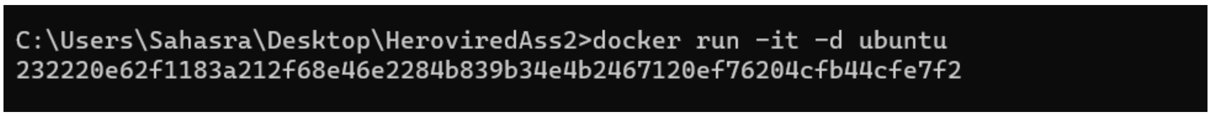
```
C:\Users\Sahasra\Desktop\HerovireddAss2>docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                         PORTS      NAMES
264c4e1471a0   ubuntu         "/bin/bash"              55 minutes ago   Exited (137) 50 minutes ago               sharp_merkle
352ba742406d   hello-world    "/hello"                 2 days ago       Exited (0) 2 days ago                     naughty_morse
05becffebdd6   hello-docker   "docker-entrypoint.s…"   2 days ago       Exited (0) 2 days ago                     serene_elgamal
81a529475779   resin/docs     "/usr/local/bin/npm …"   3 days ago       Exited (255) 2 days ago        3000/tcp   ecstatic_shaw
2b7dc7ba85d0   resin/docs     "/usr/local/bin/npm …"   3 days ago       Exited (255) 2 days ago        3000/tcp   distracted_dirac
0b1ea55f4acf   resin/docs     "/usr/local/bin/npm …"   3 days ago       Exited (0) 3 days ago                     focused_noether
0148267f5064   hello-world    "/hello"                 3 days ago       Exited (0) 3 days ago                     distracted_wing
e2defb61db13   hello-world    "/hello"                 4 days ago       Exited (0) 4 days ago                     jolly_ritchie
```

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

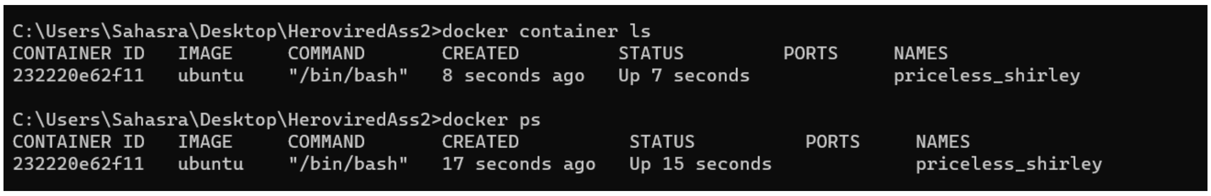


```
C:\Users\Sahasra\Desktop\HerovriedAss2>docker run -it -d ubuntu
232220e62f1183a212f68e46e2284b839b34e4b2467120ef76204cfb44cfe7f2
```

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



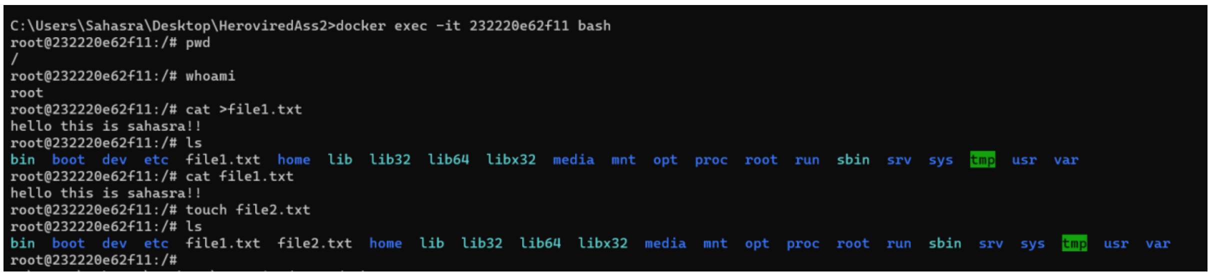
```
C:\Users\Sahasra\Desktop\HerovriedAss2>docker container ls
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS         PORTS     NAMES
232220e62f11   ubuntu    "/bin/bash"   8 seconds ago    Up 7 seconds             priceless_shirley

C:\Users\Sahasra\Desktop\HerovriedAss2>docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED           STATUS          PORTS     NAMES
232220e62f11   ubuntu    "/bin/bash"   17 seconds ago    Up 15 seconds             priceless_shirley
```

Step 7 :

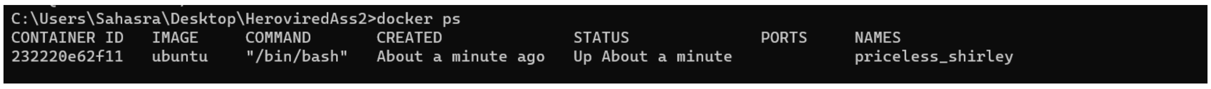
We can execute the container with the help of the command docker exec.Docker exec was used to runs a new command in a running container.

Docker exec -it <container_id> bash

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



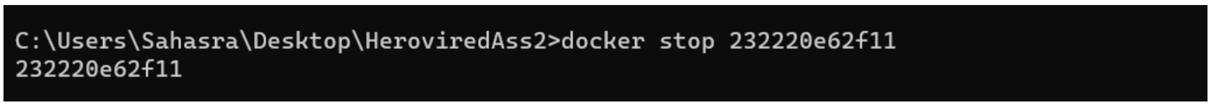
```
C:\Users\Sahasra\Desktop\HerovriedAss2>docker exec -it 232220e62f11 bash
root@232220e62f11:/# pwd
/
root@232220e62f11:/# whoami
root
root@232220e62f11:/# cat >file1.txt
hello this is sahasra!!
root@232220e62f11:/# ls
bin  boot  dev  etc  file1.txt  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@232220e62f11:/# cat file1.txt
hello this is sahasra!!
root@232220e62f11:/# touch file2.txt
root@232220e62f11:/# ls
bin  boot  dev  etc  file1.txt  file2.txt  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@232220e62f11:/#
```



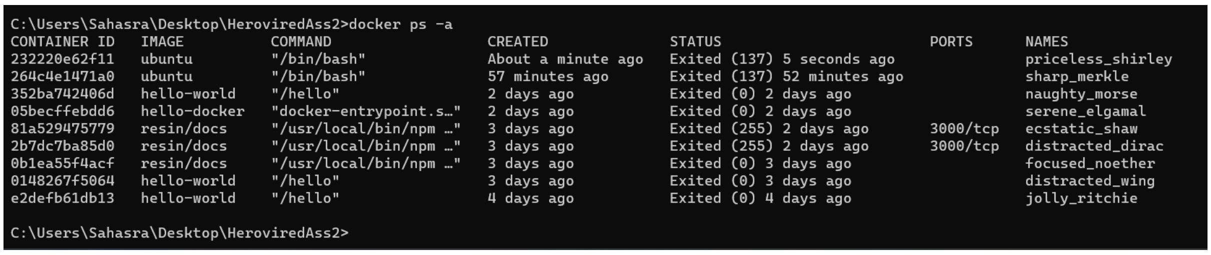
```
C:\Users\Sahasra\Desktop\HerovriedAss2>docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED              STATUS              PORTS     NAMES
232220e62f11   ubuntu    "/bin/bash"   About a minute ago   Up About a minute             priceless_shirley
```

Step 8 :

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



```
C:\Users\Sahasra\Desktop\HerovriedAss2>docker stop 232220e62f11
232220e62f11
```



```
C:\Users\Sahasra\Desktop\HerovriedAss2>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS                       PORTS      NAMES
232220e62f11   ubuntu         "/bin/bash"              About a minute ago   Exited (137) 5 seconds ago              priceless_shirley
264c4e1471a0   ubuntu         "/bin/bash"              57 minutes ago       Exited (137) 52 minutes ago             sharp_merkle
352ba742406d   hello-world    "/hello"                 2 days ago           Exited (0) 2 days ago                   naughty_morse
05becffebdd6   hello-docker   "docker-entrypoint.s…"   2 days ago           Exited (0) 2 days ago                   serene_elgamal
81a529475779   resin/docs     "/usr/local/bin/npm …"   3 days ago           Exited (255) 2 days ago      3000/tcp   ecstatic_shaw
2b7dc7ba85d0   resin/docs     "/usr/local/bin/npm …"   3 days ago           Exited (255) 2 days ago      3000/tcp   distracted_dirac
0b1ea55f4acf   resin/docs     "/usr/local/bin/npm …"   3 days ago           Exited (0) 3 days ago                   focused_noether
0148267f5064   hello-world    "/hello"                 3 days ago           Exited (0) 3 days ago                   distracted_wing
e2defb61db13   hello-world    "/hello"                 4 days ago           Exited (0) 4 days ago                   jolly_ritchie

C:\Users\Sahasra\Desktop\HerovriedAss2>
```

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.

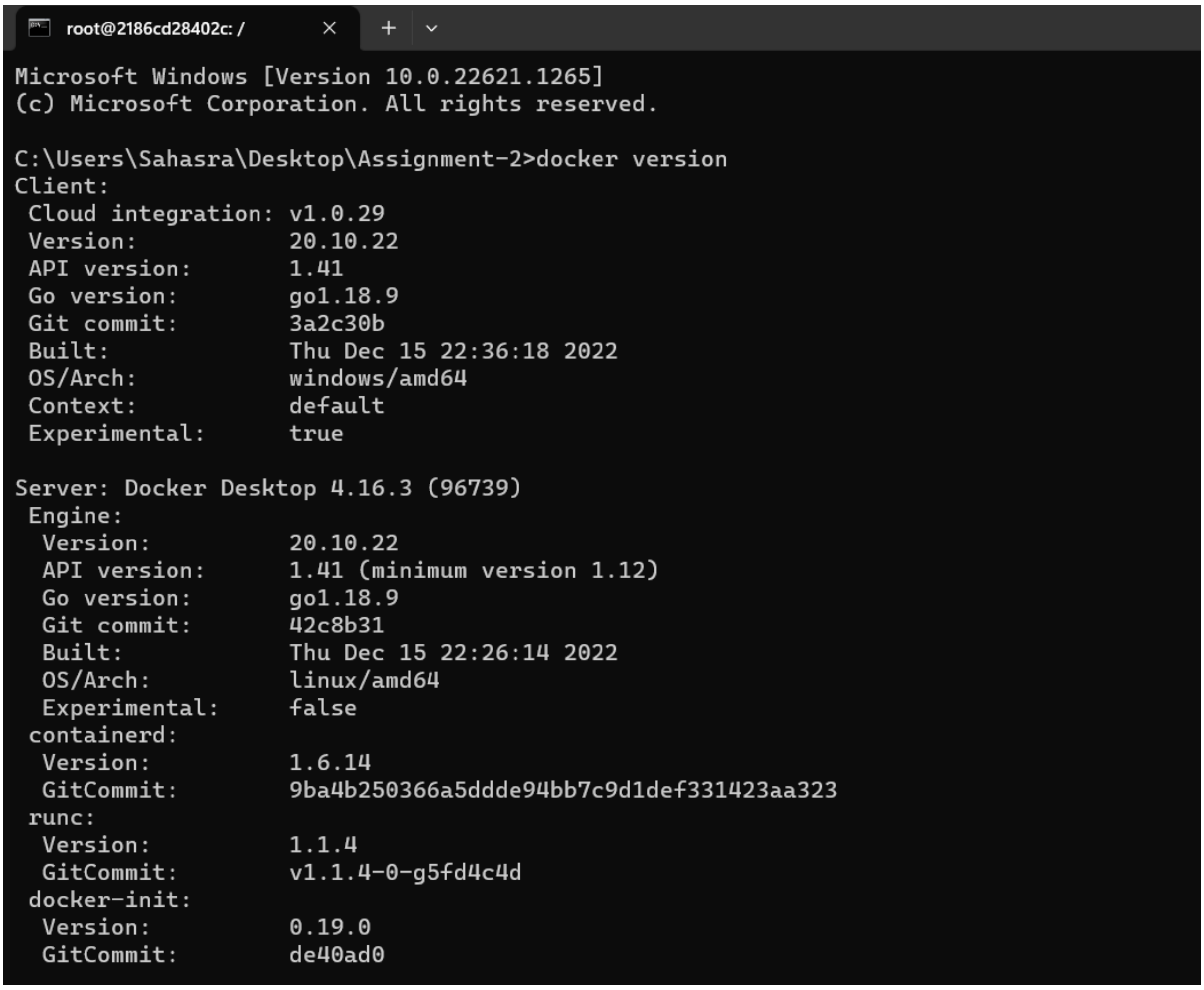
Docker container:

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 :

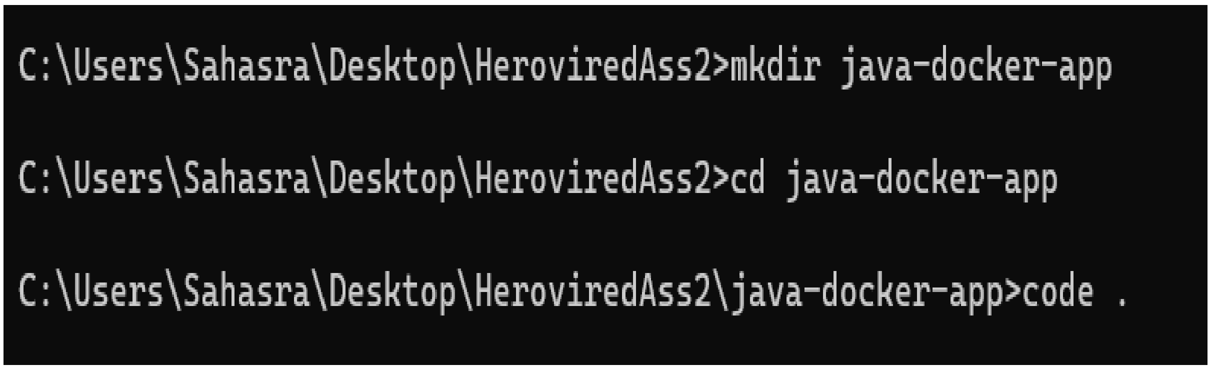
First,we need to check the version of the docker.



```
root@2186cd28402c: /

Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sahasra\Desktop\Assignment-2>docker version
Client:
 Cloud integration: v1.0.29
 Version:           20.10.22
 API version:       1.41
 Go version:        go1.18.9
 Git commit:        3a2c30b
 Built:             Thu Dec 15 22:36:18 2022
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Desktop 4.16.3 (96739)
 Engine:
  Version:          20.10.22
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.18.9
  Git commit:       42c8b31
  Built:            Thu Dec 15 22:26:14 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.14
  GitCommit:        9ba4b250366a5ddde94bb7c9d1def331423aa323
 runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

Step 2 : Creating a directory

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



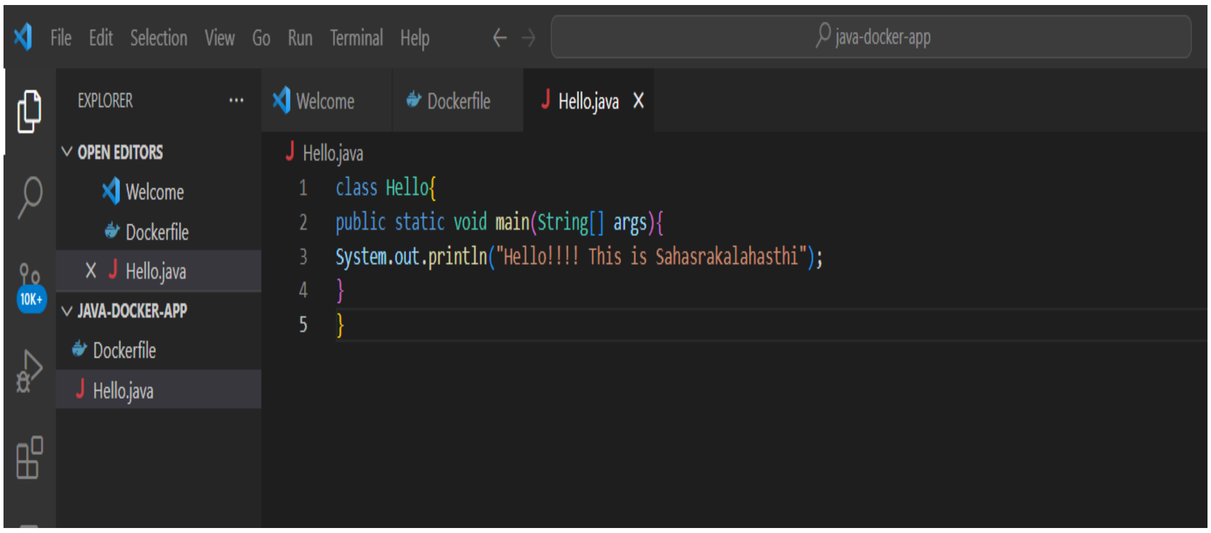
```
C:\Users\Sahasra\Desktop\HerovirediAss2>mkdir java-docker-app

C:\Users\Sahasra\Desktop\HerovirediAss2>cd java-docker-app

C:\Users\Sahasra\Desktop\HerovirediAss2\java-docker-app>code .
```

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:

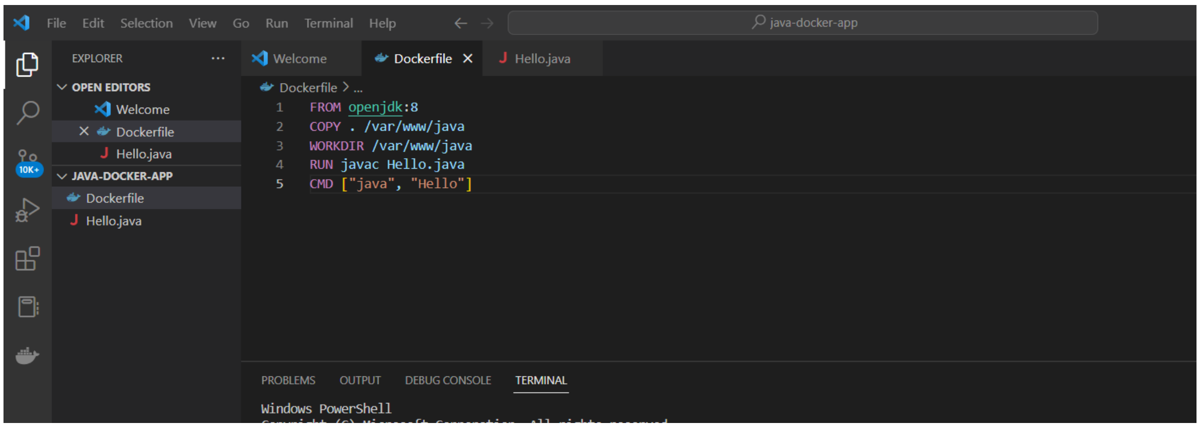
```java
class Hello{
    public static void main(String[] args){
        System.out.println("Hello!!!! This is Sahasrakalahasthi");
    }
}
```



```java
class Hello{
public static void main(String[] args){
System.out.println("Hello!!!! This is Sahasrakalahasthi");
}
}
```

Dockerfile:

```dockerfile
FROM openjdk:8
COPY . /var/www/java
WORKDIR /var/www/java
RUN javac Hello.java
CMD ["java", "Hello"]
```



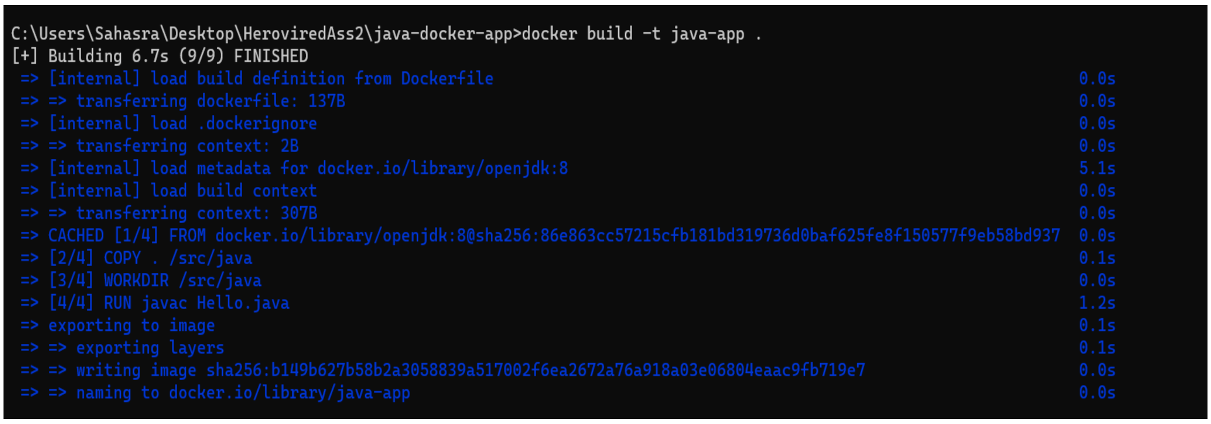
```
FROM openjdk:8
COPY . /var/www/java
WORKDIR /var/www/java
RUN javac Hello.java
CMD ["java", "Hello"]
```

```
Windows PowerShell
```

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.

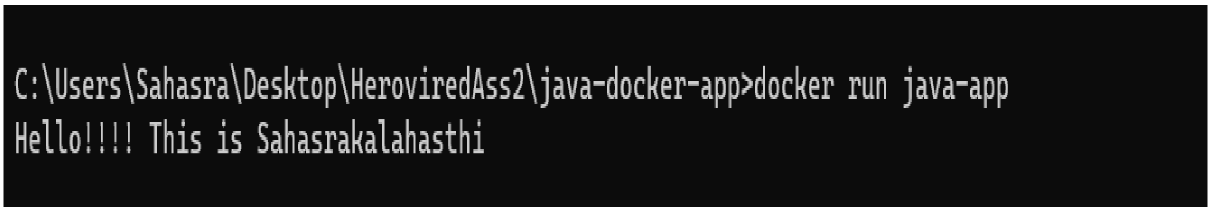
Docker build -t <image_name> .



```
C:\Users\Sahasra\Desktop\HerovirediAss2\java-docker-app>docker build -t java-app .
[+] Building 6.7s (9/9) FINISHED
 => [internal] load build definition from Dockerfile                                                 0.0s
 => => transferring dockerfile: 137B                                                                 0.0s
 => [internal] load .dockerignore                                                                    0.0s
 => => transferring context: 2B                                                                      0.0s
 => [internal] load metadata for docker.io/library/openjdk:8                                         5.1s
 => [internal] load build context                                                                    0.0s
 => => transferring context: 307B                                                                    0.0s
 => CACHED [1/4] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937  0.0s
 => [2/4] COPY . /src/java                                                                           0.1s
 => [3/4] WORKDIR /src/java                                                                          0.0s
 => [4/4] RUN javac Hello.java                                                                       1.2s
 => exporting to image                                                                               0.1s
 => => exporting layers                                                                              0.1s
 => => writing image sha256:b149b627b58b2a3058839a517002f6ea2672a76a918a03e06804eaac9fb719e7         0.0s
 => => naming to docker.io/library/java-app                                                          0.0s
```

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.



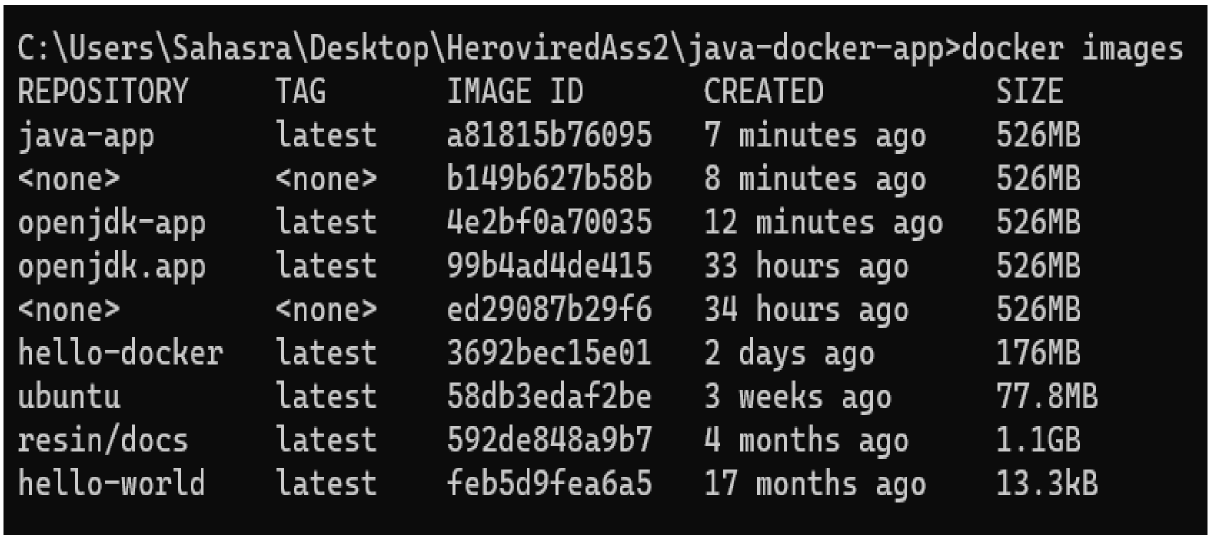
```
C:\Users\Sahasra\Desktop\HerovirediAss2\java-docker-app>docker run java-app
Hello!!!! This is Sahasrakalahasthi
```

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



```
C:\Users\Sahasra\Desktop\HerovirdedAss2\java-docker-app>docker images
REPOSITORY     TAG       IMAGE ID       CREATED          SIZE
java-app       latest    a81815b76095   7 minutes ago    526MB
<none>         <none>    b149b627b58b   8 minutes ago    526MB
openjdk-app    latest    4e2bf0a70035   12 minutes ago   526MB
openjdk.app    latest    99b4ad4de415   33 hours ago     526MB
<none>         <none>    ed29087b29f6   34 hours ago     526MB
hello-docker   latest    3692bec15e01   2 days ago       176MB
ubuntu         latest    58db3edaf2be   3 weeks ago      77.8MB
resin/docs     latest    592de848a9b7   4 months ago     1.1GB
hello-world    latest    feb5d9fea6a5   17 months ago    13.3kB
```

Git repository link:

https://github.com/sahasrakalahasthi/Herovired_Assignments