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

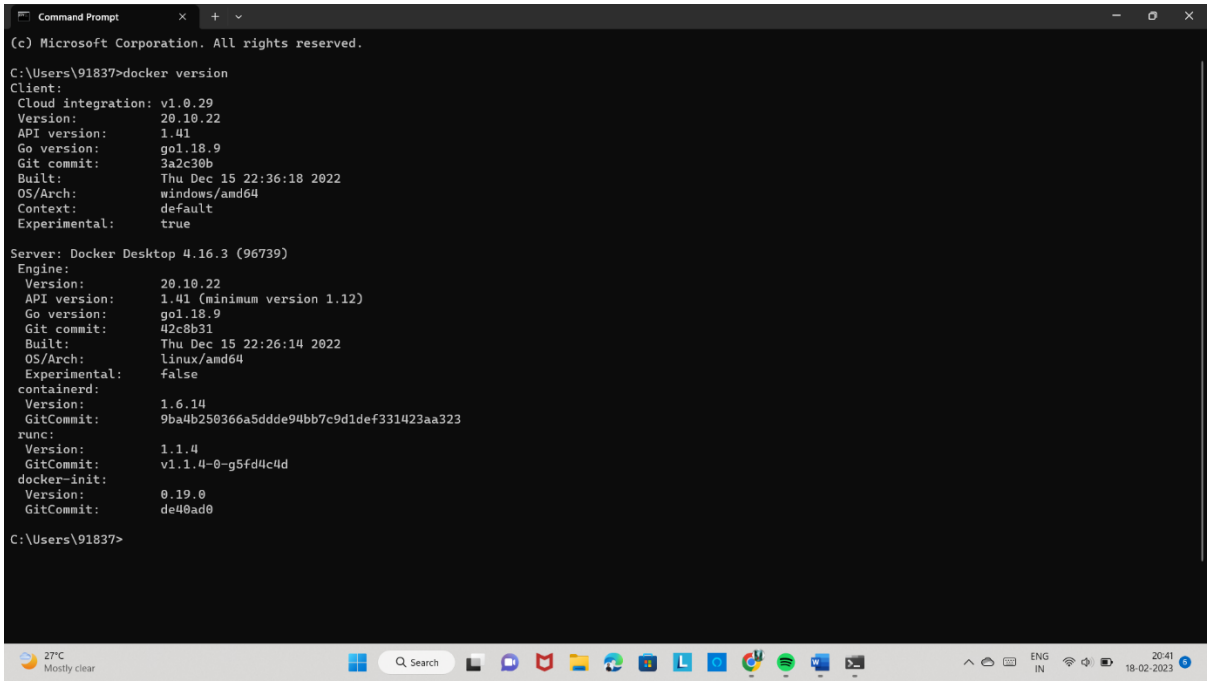
Steps to Pull any image from the docker hub and to Create its container:-

Step1:-At first, you have to install Docker image on your machine from the official Docker website.

Step2:-Check that if docker is installed correctly or not by using the command docker version in command prompt.

By using docker version command we know the version of a docker software

Command:-docker version



Step3:-To pull an image from the docker hub we use the command docker pull

Syntax:-docker pull image-name

Eg:- docker pull hello-world

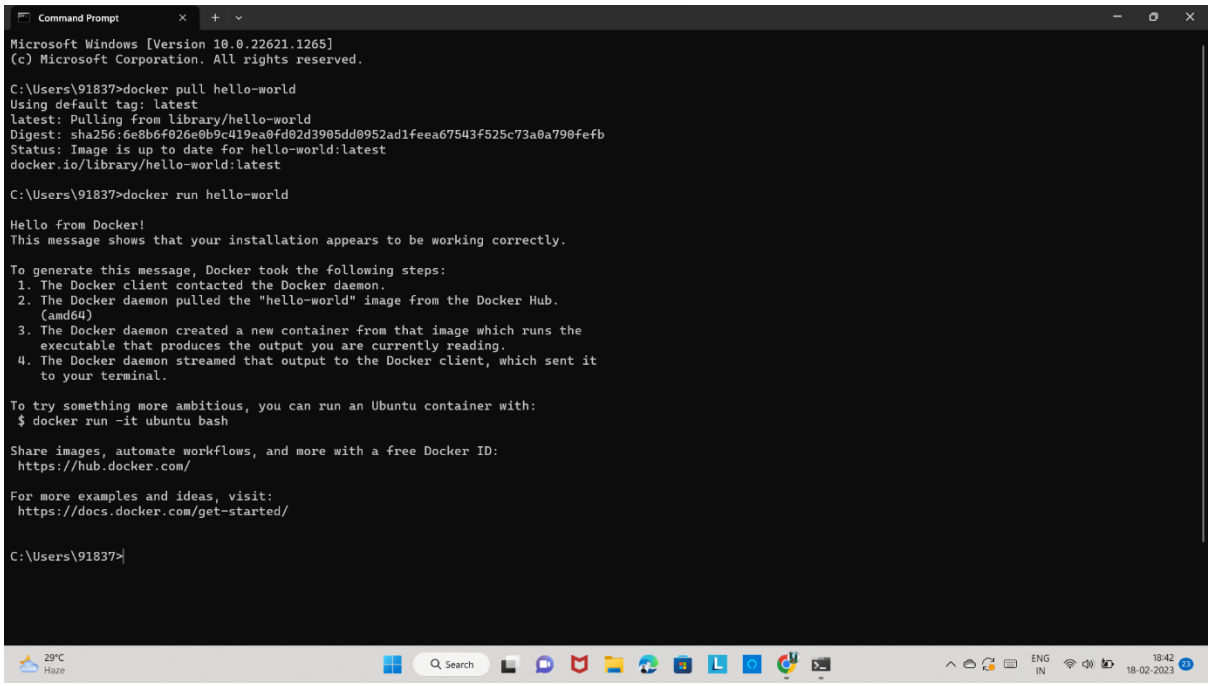
This command will download the “hello-world” image from the docker hub to your local machine.

Step4:-After the image is downloaded ,you can run the container using the command docker run.

Syntax:-docker run image-name

Eg:- docker run hello-world

This command will start a container using the “hello-world” image is working as expected.



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

Steps to create the basic java application,generate its image with necessary files and execute it with docker

Step1:-create a folder using the mkdir command

Syntax:-mkdir folder-name

Eg:-mkdir assignment-docker

After creating the folder ,go into the folder using the cd command.

Syntax:-cd folder-name

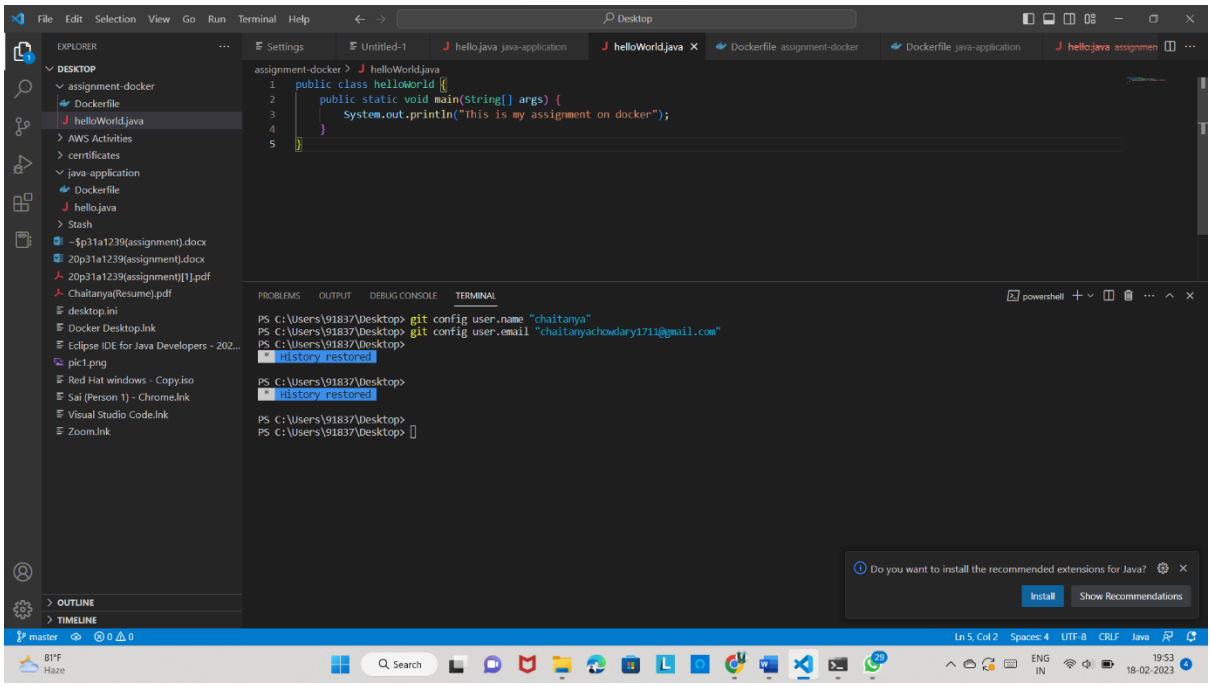
Eg:-cd assignment

Steps2:- run the “code .” in the command prompt to enter into the visual code.

Step3:-

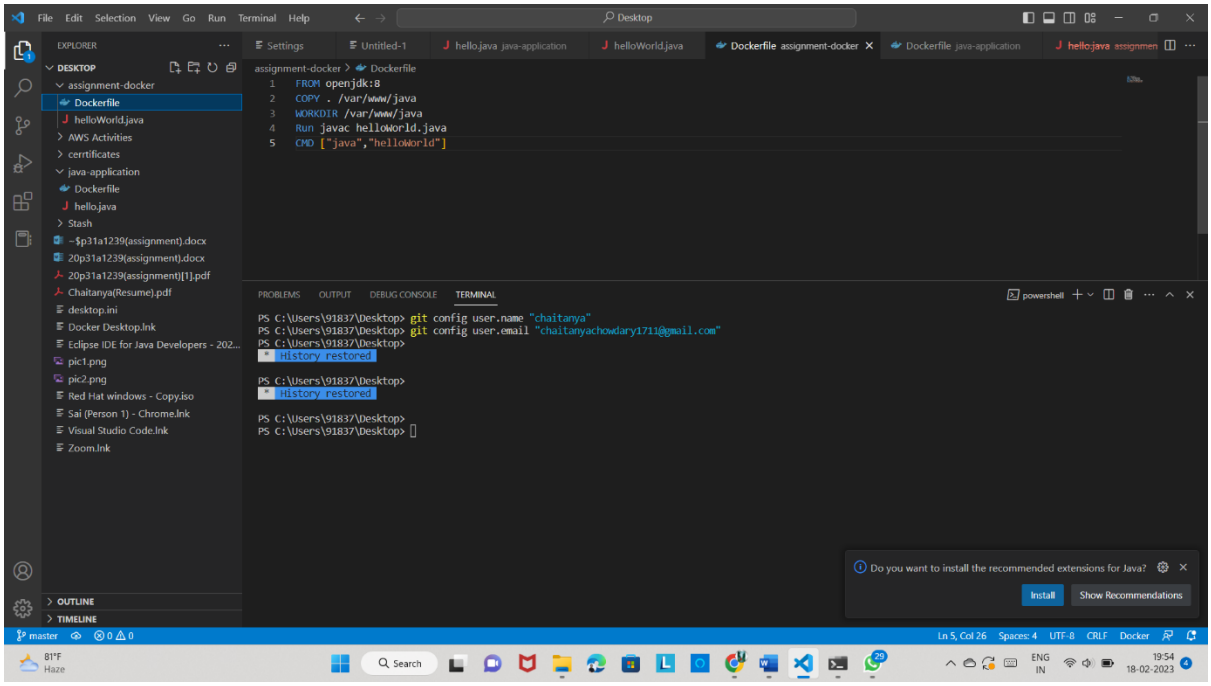
After enter into visual code create two files in the directory which was created by above.

In First file, write the java code which you want to execute and save the file with .java extension.



Step 4:- Create a Dockerfile in the same directory as your Java source code. The Dockerfile is a recipe for building a Docker image. Here's an example Dockerfile that will create an image that runs your Java application:

Note:-The name of this file has to save with “Dockerfile” only.



This Dockerfile uses the “openjdk:8”-jdk-alpine base image, which includes the Java runtime environment. It then copies the helloWorld.class file into the image and sets the command that will be run when the container is started.

Step5:-Create a docker image using the below command.

Syntax:-docker build -t image-name

Eg:-docker build -t assignment-java .

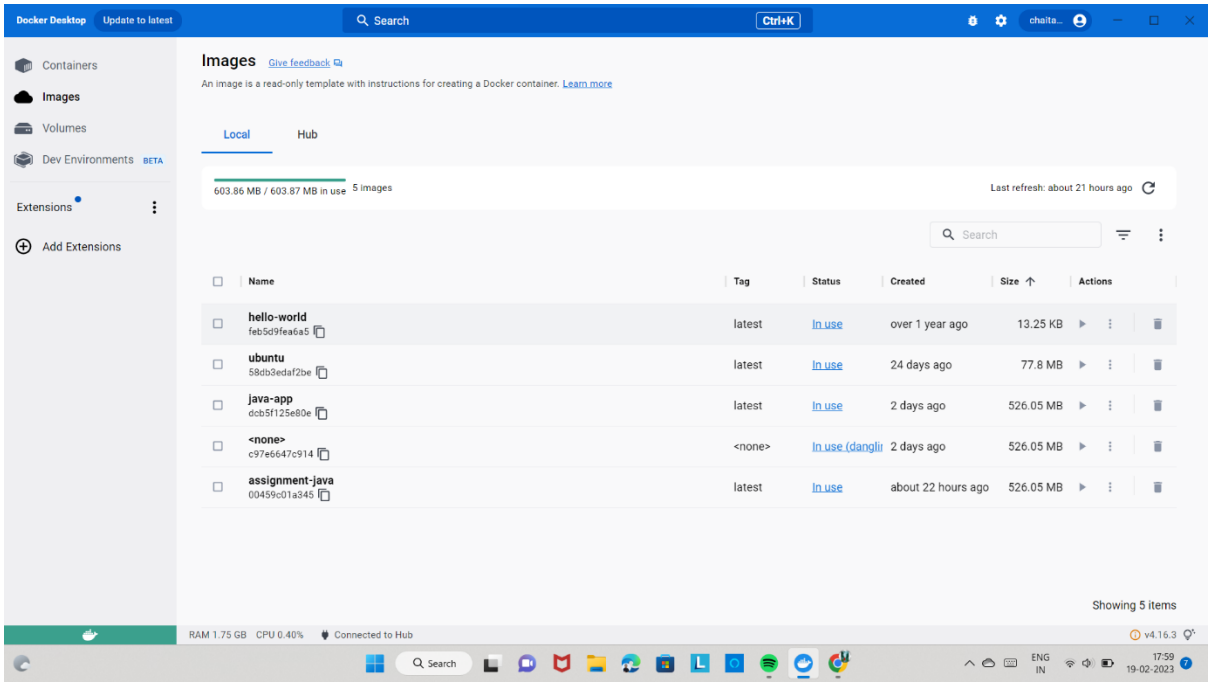
```
C:\Users\91837\Desktop\assignment-docker>docker build -t assignment-java .
[+] Building 4.5s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                 0.0s
 => => transferring dockerfile: 150B                                 0.0s
 => [internal] load .dockerignore                                    0.0s
 => => transferring context: 2B                                      0.0s
 => [internal] load metadata for docker.io/library/openjdk:8         2.4s
 => [auth] library/openjdk:pull token for registry-1.docker.io       0.0s
 => [internal] load build context                                    0.0s
 => => transferring context: 342B                                    0.0s
 => CACHED [1/4] FROM docker.io/library/openjdk:8@sha256:86e863cc5721  0.0s
 => [2/4] COPY . /var/www/java                                       0.0s
 => [3/4] WORKDIR /var/www/java                                      0.0s
 => [4/4] RUN javac helloWorld.java                                  1.7s
 => exporting to image                                               0.1s
 => => exporting layers                                              0.1s
 => => writing image sha256:00459c01a3458fc422462408b2abe356ac047edb0  0.0s
 => => naming to docker.io/library/assignment-java                   0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

Step6:-To Run the docker container use the below command.

Syntax:-docker run image-name

Eg:- docker run assignment.java

```
C:\Users\91837\Desktop\assignment-docker>docker run assignment-java
This is my assignment on docker
```


Docker Desktop
Update to latest
Search
Ctrl+K
Containers
Images
Volumes
Dev Environments
BETA
Extensions
Add Extensions
Images
Give feedback
An image is a read-only template with instructions for creating a Docker container. Learn more
Local
Hub
603.86 MB / 603.87 MB in use
5 images
Last refresh: about 21 hours ago
Search
Name
Tag
Status
Created
Size
Actions
hello-world
feb5d9fea6a5
latest
In use
over 1 year ago
13.25 KB
ubuntu
58db3edaf2be
latest
In use
24 days ago
77.8 MB
java-app
dcb5f125e80e
latest
In use
2 days ago
526.05 MB
<none>
c97e6647c914
<none>
In use (dangli
2 days ago
526.05 MB
assignment-java
00459c01a345
latest
In use
about 22 hours ago
526.05 MB
Showing 5 items
RAM 1.75 GB
CPU 0.40%
Connected to Hub
v4.16.3
Search
ENG
IN
17:59
19-02-2023