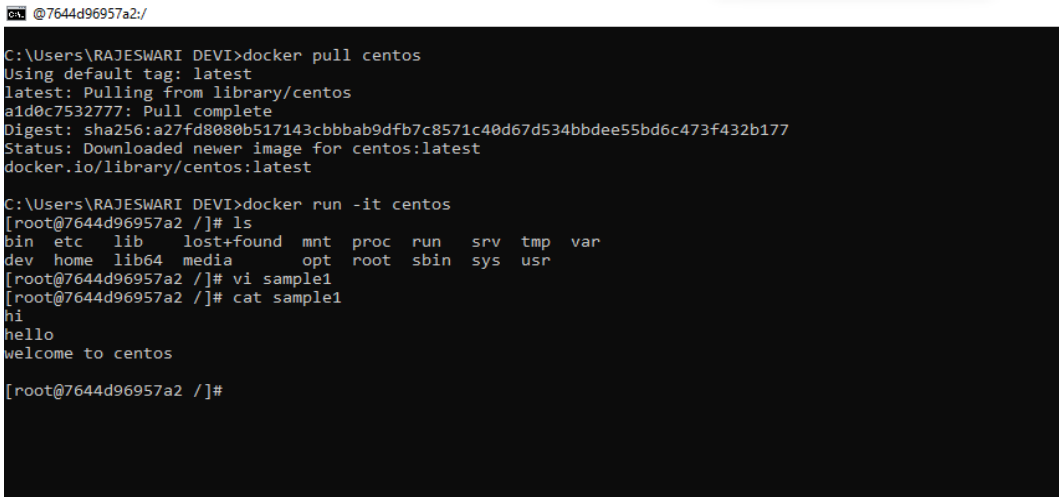
ASSIGNMENT 2:

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

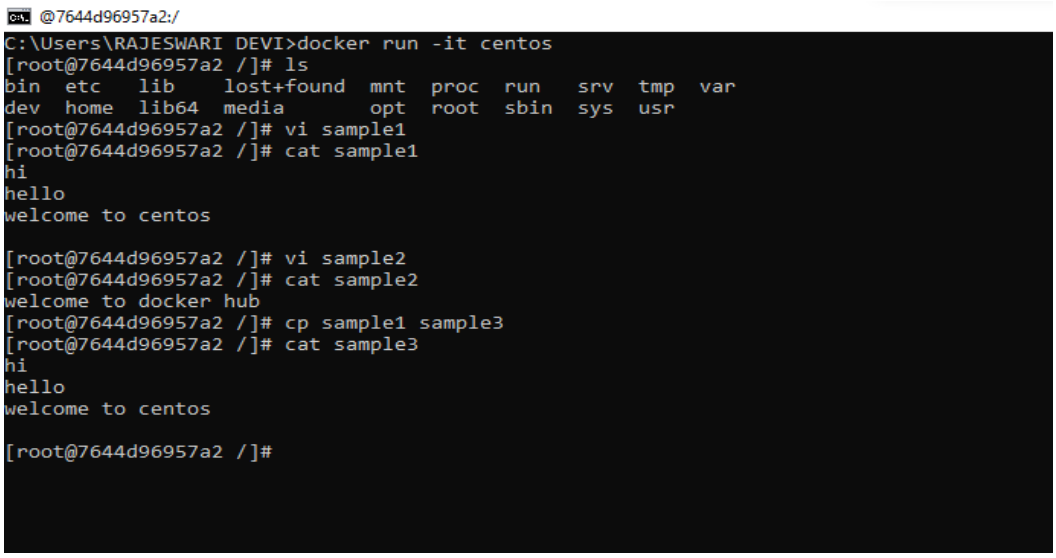
The '**docker pull**' is a Docker command to download a Docker image or a repository locally on the host from a public or private registry.

SYNTAX: docker pull [OPTIONS] NAME[:TAG|@DIGEST]



After pull an image from the docker hub we have to run it or execute by using below command

Syntax: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

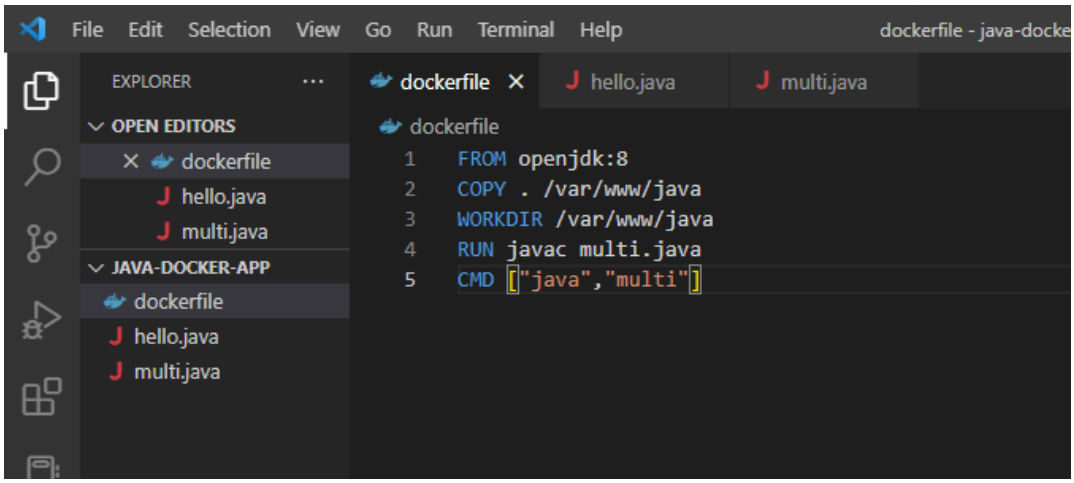


After that we have to create a sample1,sample 2 files .we have to copy file from sample1 .finally I removed sample2 file .

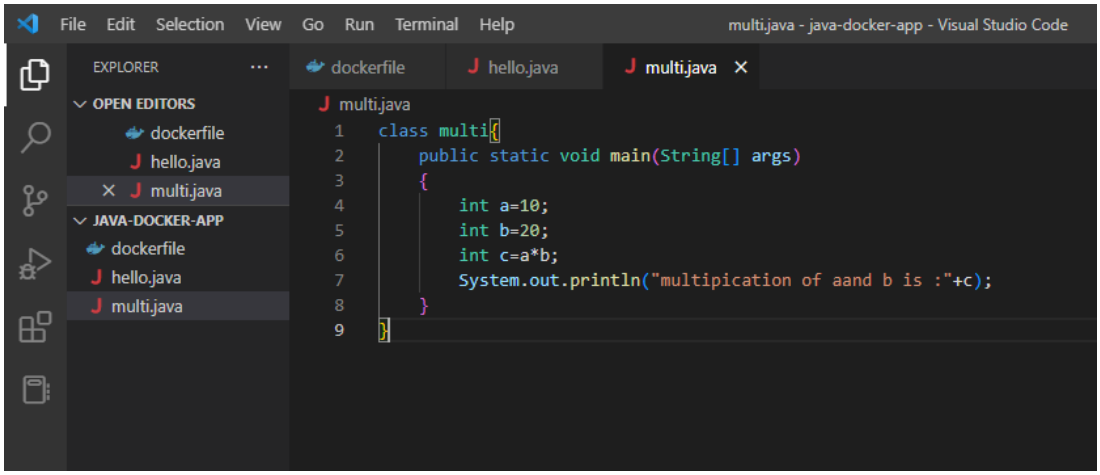
```
@7644d96957a2:/
welcome to centos

[root@7644d96957a2 /]# vi sample2
[root@7644d96957a2 /]# cat sample2
welcome to docker hub
[root@7644d96957a2 /]# cp sample1 sample3
[root@7644d96957a2 /]# cat sample3
hi
hello
welcome to centos

[root@7644d96957a2 /]# ls
bin  etc   lib    lost+found  mnt  proc  run      sample2  sbin  sys  usr
dev  home  lib64  media       opt  root  sample1  sample3  srv   tmp  var
[root@7644d96957a2 /]# rm sample2
rm: remove regular file 'sample2'? y
[root@7644d96957a2 /]# ls
bin  etc   lib    lost+found  mnt  proc  run      sample3  srv  tmp  var
dev  home  lib64  media       opt  root  sample1  sbin     sys  usr
[root@7644d96957a2 /]#
```

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

First I have created a directory called java-docker-app. In that app now I am creating a java application and a Dockerfile



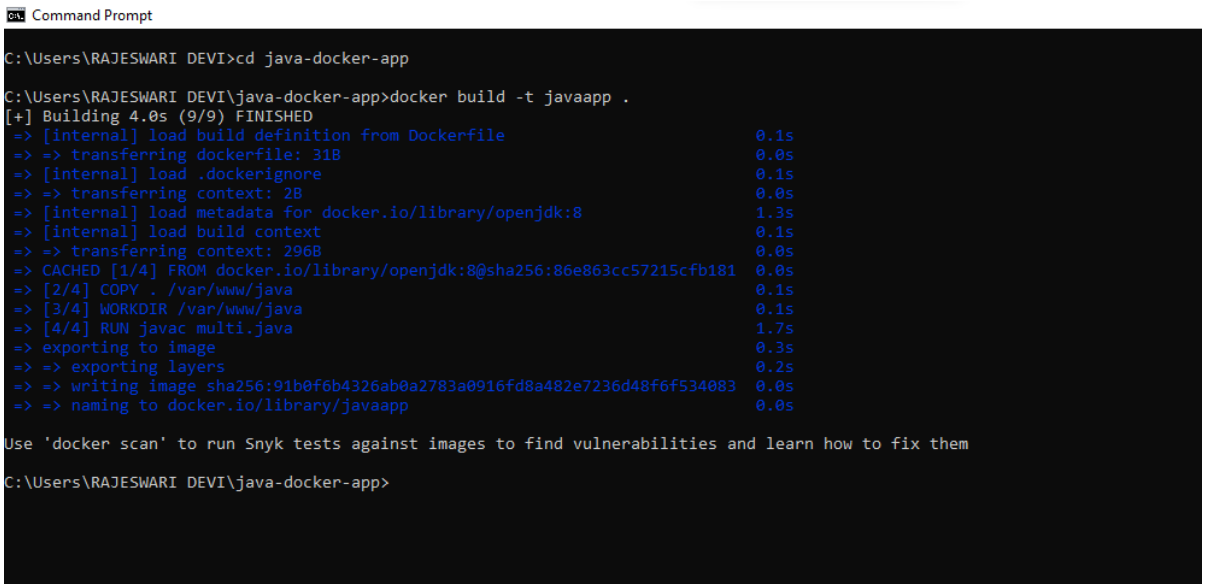
Now I created multi.java file



Now go to command prompt , I have change the directory to the java-docker-app.

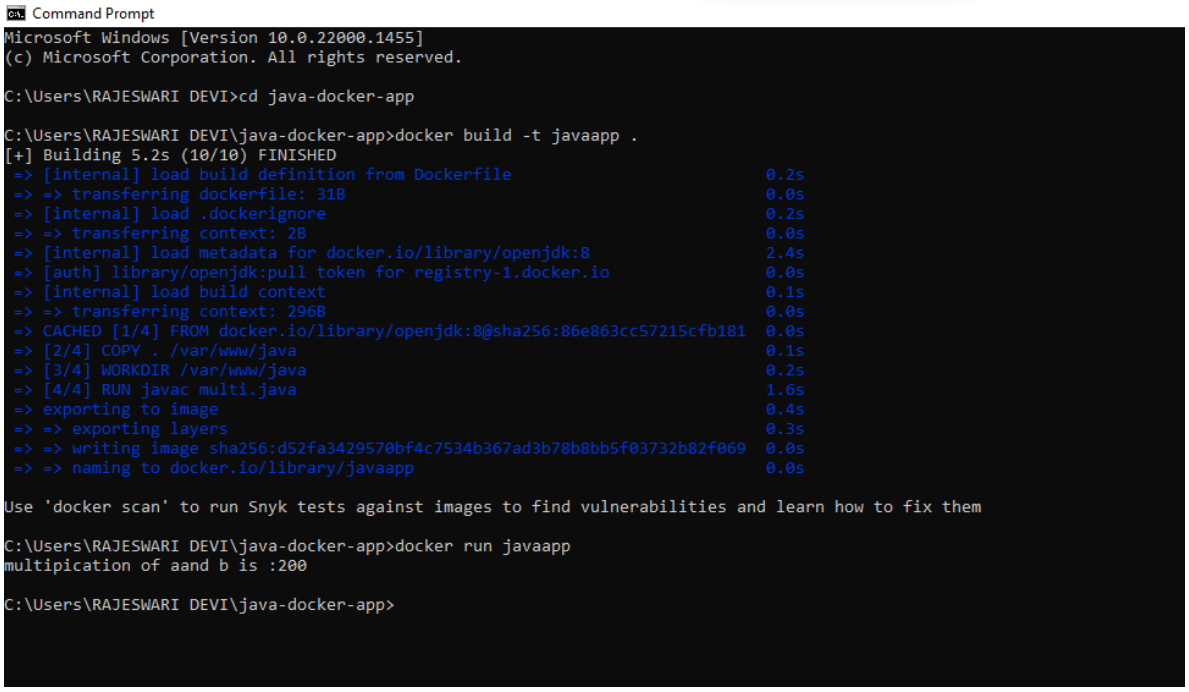
The build command is used to build an image from a Dockerfile, but the command has to be run in the same directory as the Dockerfile.

SYNTAX: docker build <options> <directory path or URL>



The docker run command first creates a writeable container layer over the specified image, and then starts it using the specified command.

Syntax: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]


Command Prompt
Microsoft Windows [Version 10.0.22000.1455]
(c) Microsoft Corporation. All rights reserved.
C:\Users\RAJESWARI DEVI>cd java-docker-app
C:\Users\RAJESWARI DEVI\java-docker-app>docker build -t javaapp .
[+] Building 5.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile 0.2s
=> => transferring dockerfile: 31B 0.0s
=> [internal] load .dockerignore 0.2s
=> => 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 0.1s
=> => transferring context: 296B 0.0s
=> CACHED [1/4] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb181 0.0s
=> [2/4] COPY . /var/www/java 0.1s
=> [3/4] WORKDIR /var/www/java 0.2s
=> [4/4] RUN javac multi.java 1.6s
=> exporting to image 0.4s
=> => exporting layers 0.3s
=> => writing image sha256:d52fa3429570bf4c7534b367ad3b78b8bb5f03732b82f069 0.0s
=> => naming to docker.io/library/javaapp 0.0s
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
C:\Users\RAJESWARI DEVI\java-docker-app>docker run javaapp
multipication of aand b is :200
C:\Users\RAJESWARI DEVI\java-docker-app>