Graded Assignment on Docker:

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

Docker is a software platform to create, test and deploy applications in an isolated environment. Docker uses a container to package up an application with all of the parts it needs including, libraries and dependencies. It allows applications to use the kernel and other resources of the host operating system this will boost the performance and reduce the size of the application. Docker Hub is a centralized repository service that allows you to store container images and share them with your team. You can use Pull and Push commands to upload and download images to and from the Docker Hub.

Docker version

Step-1: Verify the Docker version and also log in to Docker Hub. docker version docker login

```
D:\pradeep>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: Pull the Image from Docker Hub. ...

```
D:\pradeep>docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
bb263680fed1: Pull complete
258f176fd226: Pull complete
a0bc35e70773: Pull complete
077b9569ff86: Pull complete
3082a16f3b61: Pull complete
7e9b29976cce: Pull complete
Digest: sha256:6650513efd1d27c1f8a5351cbd33edf85cc7e0d9d0fcb4ffb23d8fa89b601ba8
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

Step-3: Next, create a new nginx container from the downloaded image and expose it on port 80 using the following command.

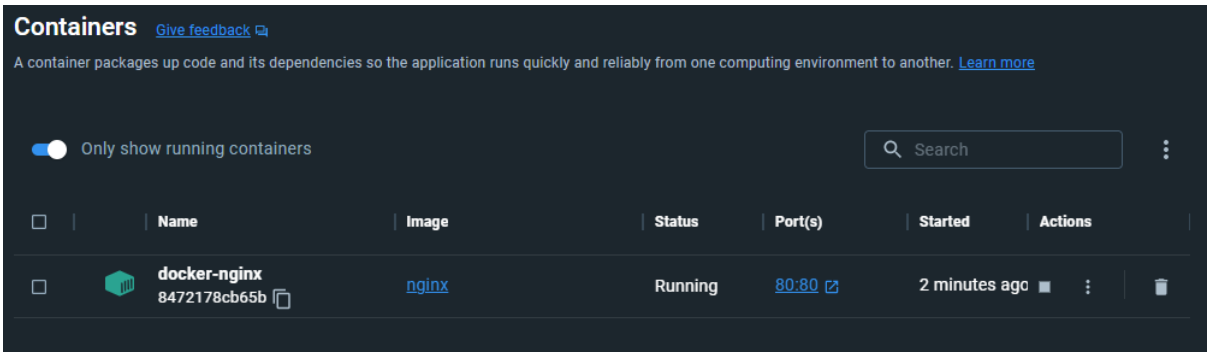
```
D:\pradeep>docker run --name docker-nginx -p 80:80 -d nginx
8472178cb65b2681fbdcda8ad883daa6e16c6618ee318ff88816a78f0ac1c559
```

```
D:\pradeep>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                NAMES
8472178cb65b   nginx     "/docker-entrypoint.…"   7 seconds ago   Up 6 seconds   0.0.0.0:80->80/tcp   docker-nginx
```

Step-5: Connect to Container Terminal…

```
D:\pradeep>docker exec -it docker-nginx /bin/bash
root@8472178cb65b:/# apt update
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8183 kB]
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [226 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [14.6 kB]
Fetched 8632 kB in 18s (469 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
root@8472178cb65b:/#
```

Docker Desktop



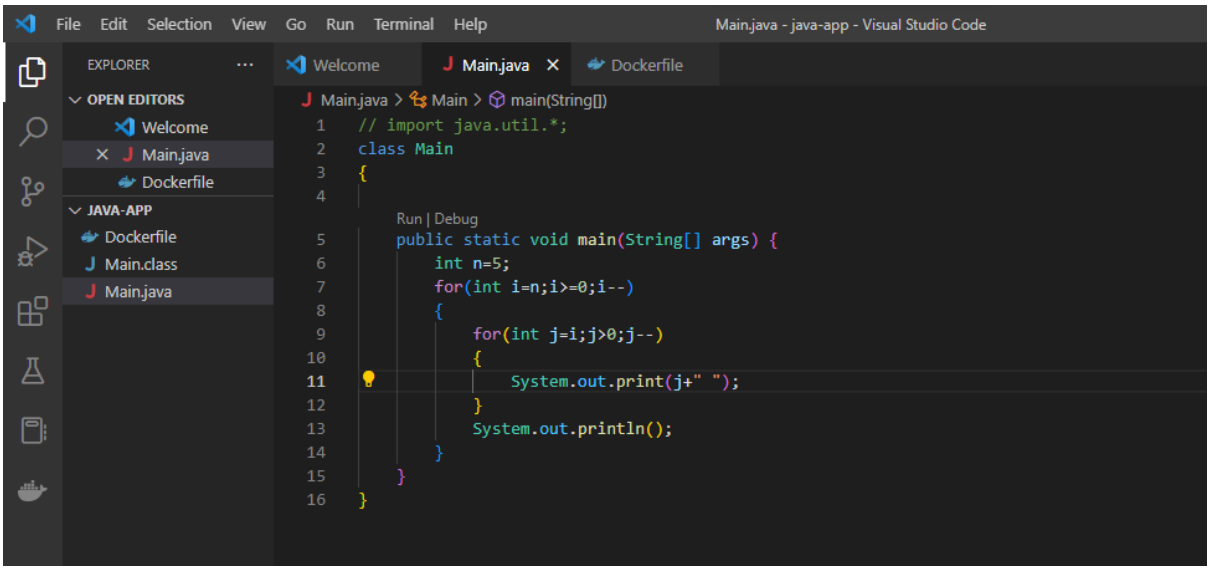
Q2) Create the basic java application, generate its image with necessary files, and execute it with docker. Creating the basic java application.

Step1: Create a folder and two files.

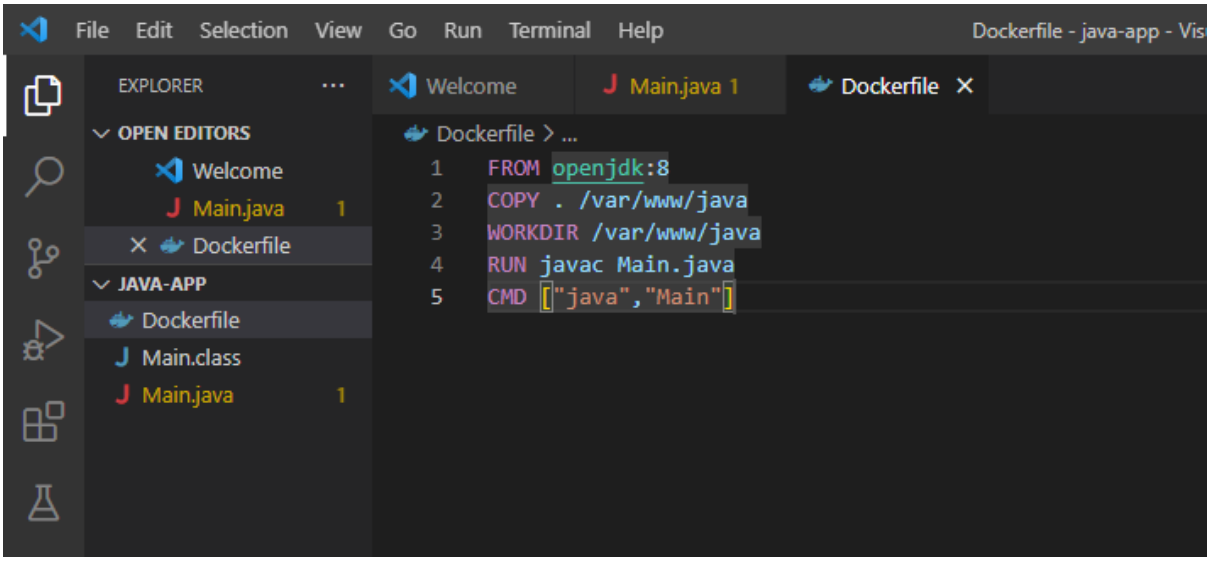
1.mains.java

2.Docker

Step2: Create a java file, and save it as a mains.java



Step3:Create a Docker file



Step5:Now create an image by following the below command. we must log in as root in order to create an image. In the following command,java-app is the name of the image. We can have any name for our docker image.

```
D:\pradeep\java-app>javac Main.java
```

```
D:\pradeep\java-app>docker build -t java-app .
[+] Building 3.8s (9/9) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.1s
 => => transferring dockerfile: 31B                                                                                 0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load metadata for docker.io/library/openjdk:8                                                        1.4s
 => [internal] load build context                                                                                   0.0s
 => => transferring context: 1.37kB                                                                                 0.0s
 => CACHED [1/4] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452cb8  0.0s
 => [2/4] COPY . /var/www/java                                                                                      0.1s
 => [3/4] WORKDIR /var/www/java                                                                                     0.1s
 => [4/4] RUN javac Main.java                                                                                       1.7s
 => exporting to image                                                                                              0.1s
 => => exporting layers                                                                                             0.1s
 => => writing image sha256:f4d86e433db0ad3d06d1638426790545150ba8f6910bf470d7a73537d5798635                        0.0s
 => => naming to docker.io/library/java-app                                                                         0.0s

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

```
D:\pradeep\java-app>docker run java-app
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```