Graded Assignment on Docker

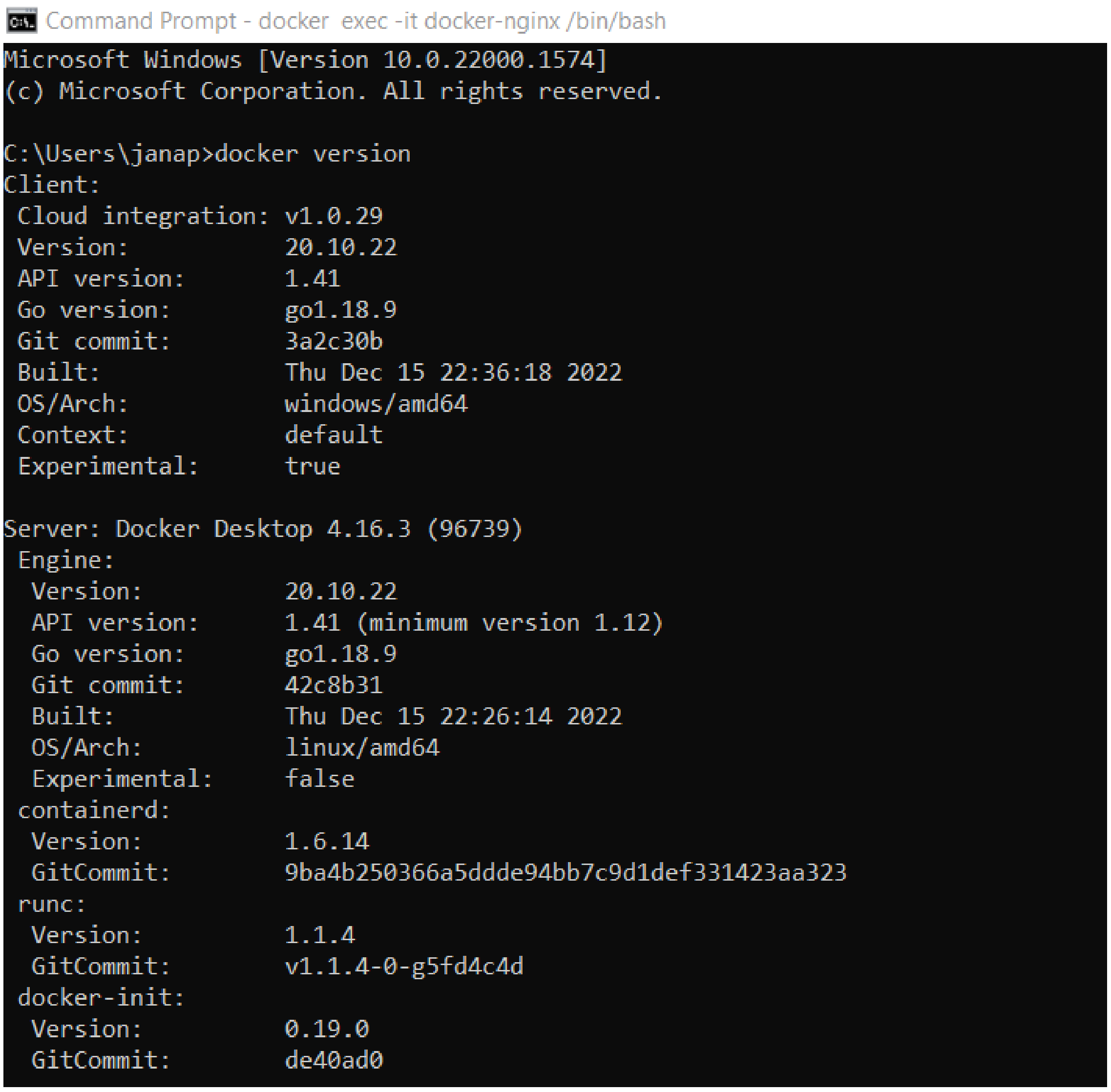
Q1) Pull any image from the docker hub, create its container, and execute it showing the output.

Docker:

- Docker provides a way to run applications securely isolated in a container, packaged with all its dependencies and libraries.
- 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.

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

Docker version: This command is used to get the currently installed version of docker

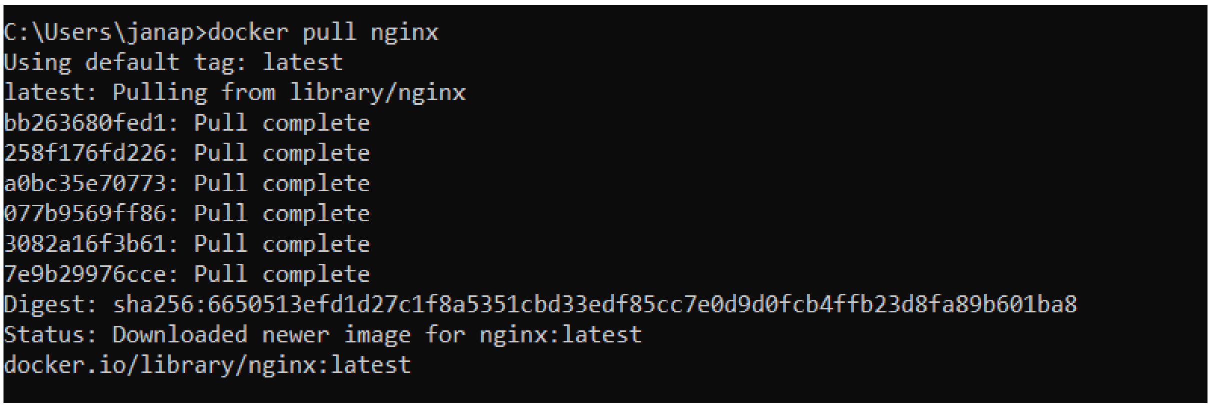


Command Prompt - docker exec -it docker-nginx /bin/bash

```
Microsoft Windows [Version 10.0.22000.1574]
(c) Microsoft Corporation. All rights reserved.

C:\Users\janap>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.

<u>Docker Hub:</u> 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.

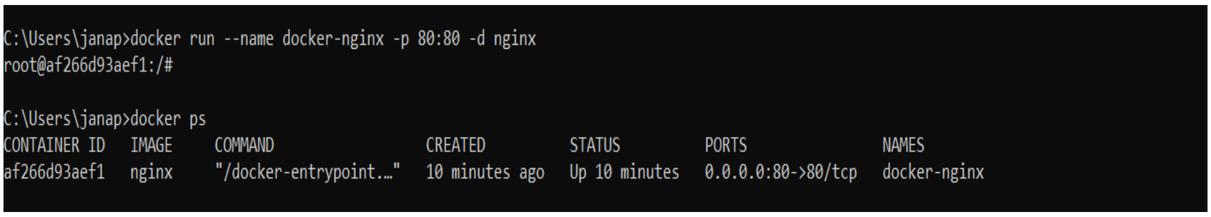


```
C:\Users\janap>docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
bb263680fed1: Pull complete
258f176fd226: Pull complete
a0bc35e70773: Pull complete
077b9569ff86: Pull complete
3082a16f3b61: Pull complete
7e9b29976cce: Pull complete
Digest: sha256:6650513efd1d27c1f8a5351cbd33edf85cc7e0d9d0fcb4ffb23d8fa89b601ba8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

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

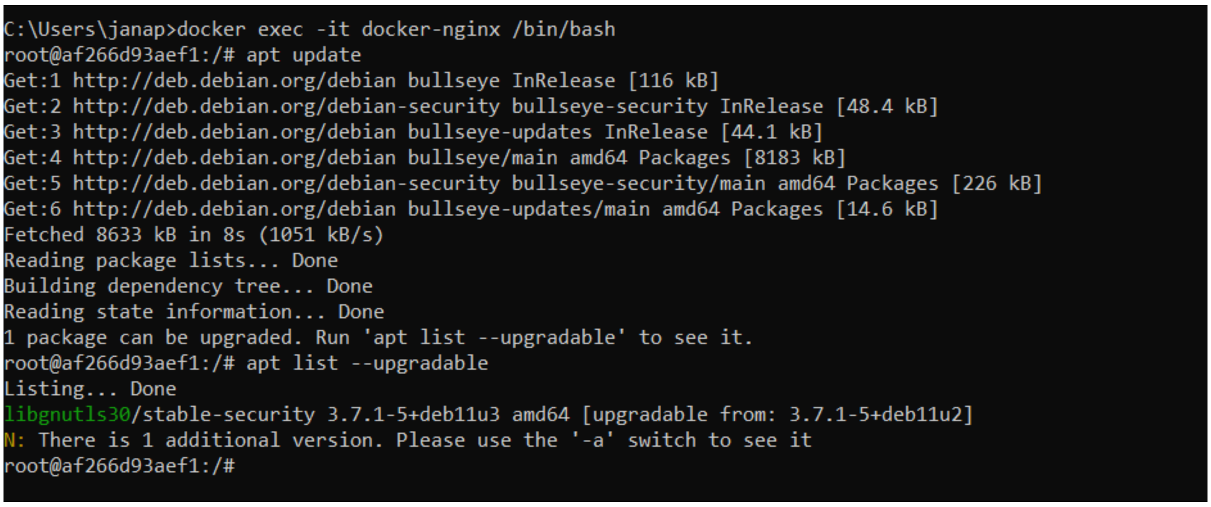
<u>Docker run:</u> This command creates a container from a given image and starts the container using a given command.

<u>Docker ps:</u> This command only shows running containers by default.



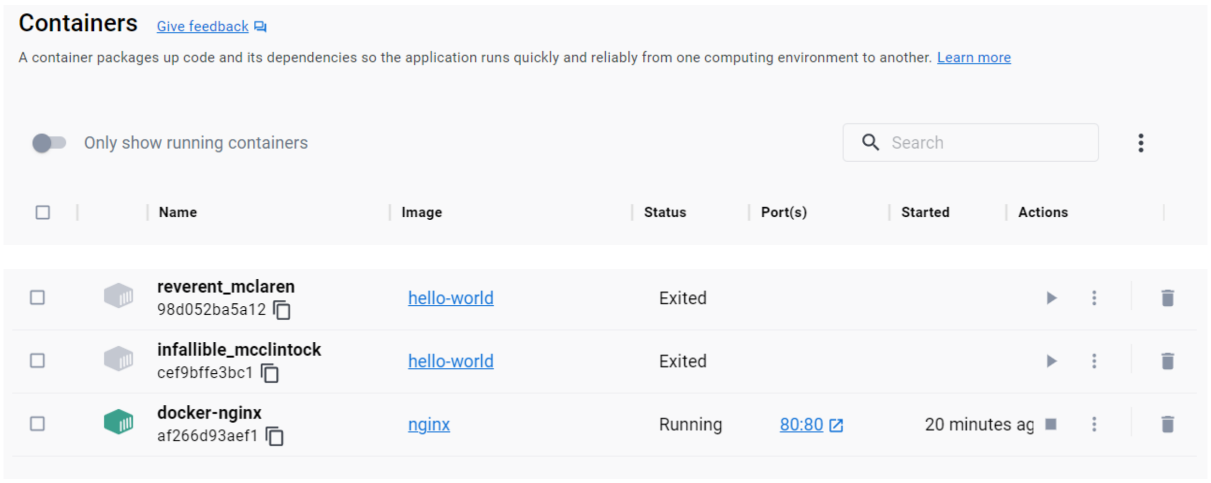
```
C:\Users\janap>docker run --name docker-nginx -p 80:80 -d nginx
root@af266d93aef1:/#

C:\Users\janap>docker ps
CONTAINER ID   IMAGE    COMMAND                  CREATED          STATUS          PORTS                NAMES
af266d93aef1   nginx    "/docker-entrypoint.…"   10 minutes ago   Up 10 minutes   0.0.0.0:80->80/tcp   docker-nginx
```

Step 4: Connect to Container Terminal.



```
C:\Users\janap>docker exec -it docker-nginx /bin/bash
root@af266d93aef1:/# 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 8633 kB in 8s (1051 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@af266d93aef1:/# apt list --upgradable
Listing... Done
libgnutls30/stable-security 3.7.1-5+deb11u3 amd64 [upgradable from: 3.7.1-5+deb11u2]
N: There is 1 additional version. Please use the '-a' switch to see it
root@af266d93aef1:/#
```

Docker Desktop:



Containers Give feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

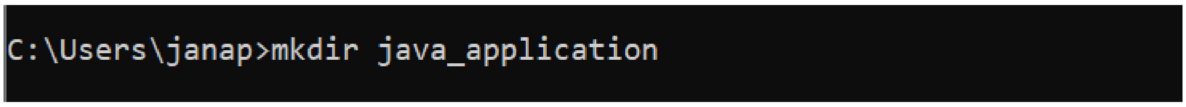
Only show running containers

Name	Image	Status	Port(s)	Started	Actions
reverent_mclaren 98d052ba5a12	hello-world	Exited			
infallible_mcclintock cef9bffe3bc1	hello-world	Exited			
docker-nginx af266d93aef1	nginx	Running	80:80	20 minutes ag	

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

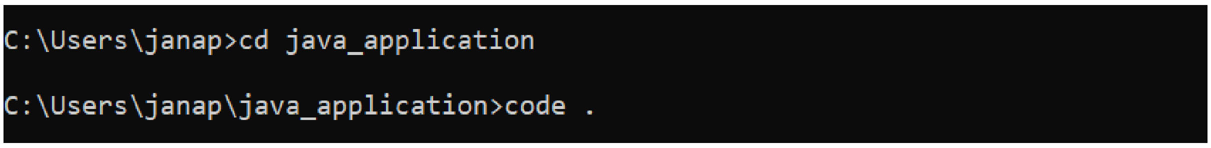
Creating the basic java application.

Step 1: Create a directory, which is used to store the files. In this case, we are also creating two files namely Hello.java and another Docker file.



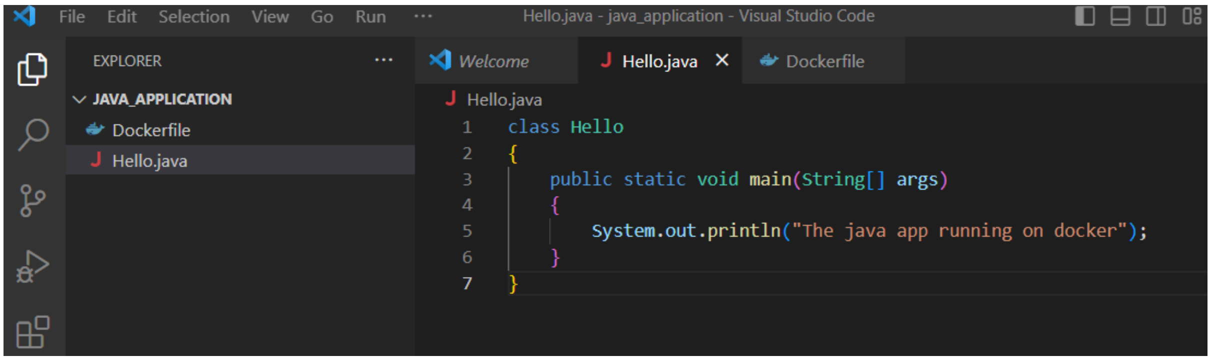
```
C:\Users\janap>mkdir java_application
```

Step 2: Go to the directory that you have created.



```
C:\Users\janap>cd java_application

C:\Users\janap\java_application>code .
```

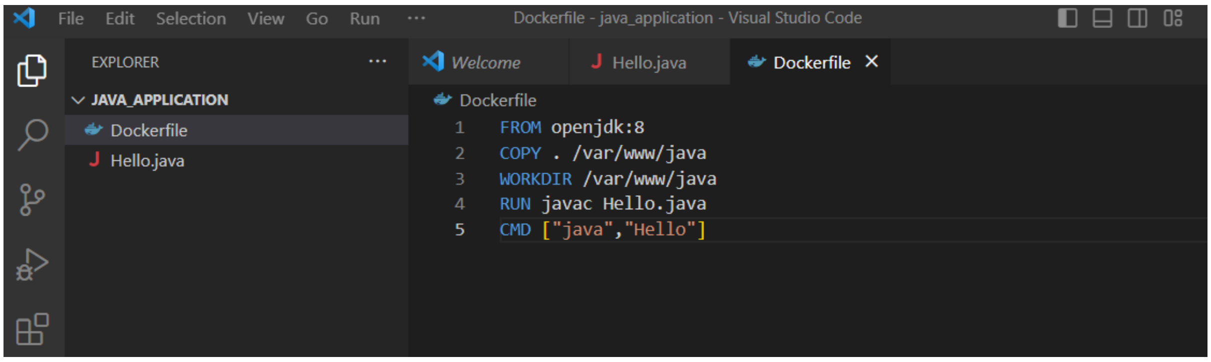
Step 3: Create a java file, and save it as Hello.java, in which we are executing the "The java app running on docker" message.



File Edit Selection View Go Run ···
Hello.java - java_application - Visual Studio Code
EXPLORER
JAVA_APPLICATION
Dockerfile
Hello.java
Welcome
Hello.java
Dockerfile

```java
class Hello
{
    public static void main(String[] args)
    {
        System.out.println("The java app running on docker");
    }
}
```

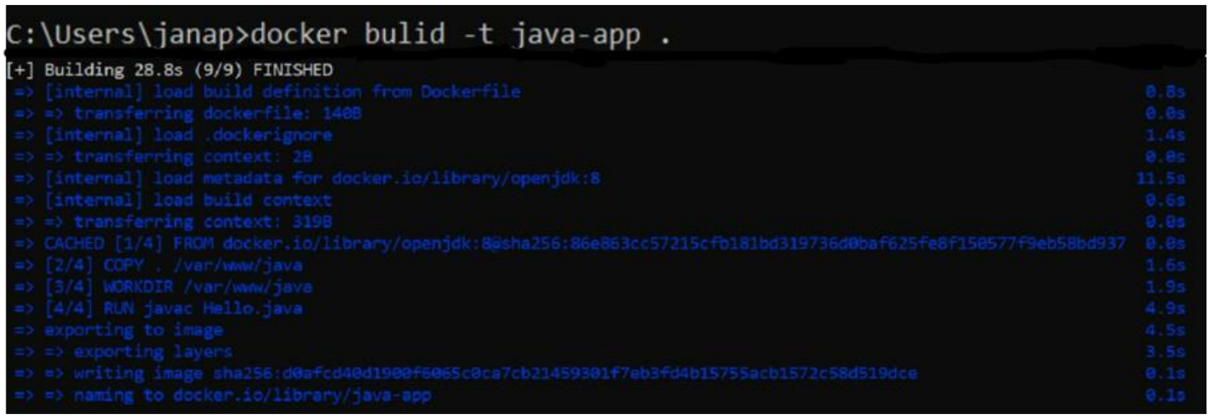
Step 4: Create a docker file.



File Edit Selection View Go Run ···
Dockerfile - java_application - Visual Studio Code
EXPLORER
JAVA_APPLICATION
Dockerfile
Hello.java
Welcome
Hello.java
Dockerfile

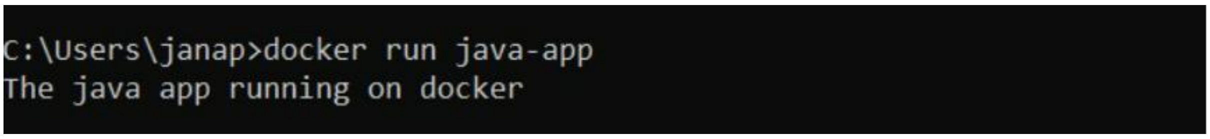
```
FROM openjdk:8
COPY . /var/www/java
WORKDIR /var/www/java
RUN javac Hello.java
CMD ["java","Hello"]
```

Step 5: 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.



```
C:\Users\janap>docker bulid -t java-app .
[+] Building 28.8s (9/9) FINISHED
 => [internal] load build definition from Dockerfile                                                        0.8s
 => => transferring dockerfile: 140B                                                                        0.0s
 => [internal] load .dockerignore                                                                           1.4s
 => => transferring context: 2B                                                                             0.0s
 => [internal] load metadata for docker.io/library/openjdk:8                                               11.5s
 => [internal] load build context                                                                           0.6s
 => => transferring context: 319B                                                                           0.0s
 => CACHED [1/4] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937  0.0s
 => [2/4] COPY . /var/www/java                                                                              1.6s
 => [3/4] WORKDIR /var/www/java                                                                             1.9s
 => [4/4] RUN javac Hello.java                                                                              4.9s
 => exporting to image                                                                                      4.5s
 => => exporting layers                                                                                     3.5s
 => => writing image sha256:d0afcd40d1900f6065c0ca7cb21459301f7eb3fd4b15755acb1572c58d519dce                0.1s
 => => naming to docker.io/library/java-app                                                                 0.1s
```

Step 6: After successfully building the image, now we can run docker by using the run command.



```
C:\Users\janap>docker run java-app
The java app running on docker
```