CONTAINER ORCHESTRATION AND INFRASTRUCTURE AUTOMATION LAB

NAME- Shubhi Dixit

BATCH- 05

SAP ID- 500094571

Dockerfile

Step 1- Create the directory “images”

Step 2- Change the current directory to “images”

Step 3- Create 2 files named “hello.java” and “Dockerfile”

Command Prompt - docker p

```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>mkdir images

C:\Users\admin>cd images

C:\Users\admin\images>notepad Hello.java

C:\Users\admin\images>notepad Dockerfile
```

Hello.java

File Edit View

```
class Hello
{
     public static void main( String args[])
      {
         System.out.println("Hello World");
      }
}
```

Hello.java • Dockerfile.txt **Dockerfile**

File Edit View

```
FROM ubuntu:latest
RUN apt-get update && apt-get install -y openjdk-11-jdk
COPY Hello.java /app/
WORKDIR /app/
RUN javac Hello.java
CMD ["java","Hello"]
```

Step 4- Now using build command build an image through dockerfile and give any name to your image and create a container through that image.

```
C:\Users\admin\images>docker build -t mynewimage .
[+] Building 2.1s (10/10) FINISHED                                                    docker:default
 => [internal] load build definition from Dockerfile                                            0.0s
 => => transferring dockerfile: 196B                                                            0.0s
 => [internal] load .dockerignore                                                               0.0s
 => => transferring context: 2B                                                                 0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                0.0s
 => [internal] load build context                                                               0.0s
 => => transferring context: 164B                                                               0.0s
 => [1/5] FROM docker.io/library/ubuntu:latest                                                  0.0s
 => CACHED [2/5] RUN apt-get update && apt-get install -y openjdk-11-jdk                        0.0s
 => [3/5] COPY Hello.java /app/                                                                 0.0s
 => [4/5] WORKDIR /app/                                                                         0.1s
 => [5/5] RUN javac Hello.java                                                                  1.7s
 => exporting to image                                                                          0.1s
 => => exporting layers                                                                         0.1s
 => => writing image sha256:39c45d02e19da6f3b4f2c4408ac95a7fcd67ec1e6d5377af45f983361d09cb16    0.0s
 => => naming to docker.io/library/mynewimage                                                   0.0s

What's Next?
  View summary of image vulnerabilities and recommendations → docker scout quickview

C:\Users\admin\images>docker run -dt --name=shubhi mynewimage
6e94ef1a91161a72c74eb927bcadf55683f017df9090062fabd012d8e9ac40fe
```

Step 5- Use commit command to create an image of the entire setup(container + image) and then push this image to your account in Docker Hub by creating a private repository and give it a name.

```
C:\Users\admin\images>docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/

C:\Users\admin\images>docker commit shubhi shubhid/newrepo
sha256:d732f1af5df1d42d6f081e072fad39892b2eeb1137eab59fb10995d06e99acdf

C:\Users\admin\images>docker push shubhid/newrepo
Using default tag: latest
The push refers to repository [docker.io/shubhid/newrepo]
e402efb6ab11: Pushed
8c92fe12e338: Pushed
5f70bf18a086: Pushed
a6efb10f6dbb: Pushed
97ba0ca3f908: Pushed
dc0585a4b8b7: Mounted from library/ubuntu
Patch "https://registry-1.docker.io/v2/shubhid/newrepo/blobs/uploads/cf0a5f97-30cb-4e70-9f00-7ac7fc47f3f7?_state=52BIstcCipyLz4GRUisWzq0_ypdg5znsH1P
NuZLzh-17Ik5hbWUiOiJzaHViaGlkL25ld3JlcG8iLCJVVUlEIjoiY2YwYTVmOTctMzBjYi00ZTcwLTlmMDAtN2FjN2ZjNDdmM2Y3IiwiT2Zmc2V0IjowLCJTdGFydGVkQXQiOiIyMDIzLTA5LTE
zVDE1OjE1OjUzLjM1NzI4NDM3NFoifQ%3D%3D": net/http: TLS handshake timeout

C:\Users\admin\images>docker push shubhid/newrepo
Using default tag: latest
The push refers to repository [docker.io/shubhid/newrepo]
e402efb6ab11: Layer already exists
8c92fe12e338: Layer already exists
5f70bf18a086: Pushed
a6efb10f6dbb: Layer already exists
```

Step 6- Now pull the image from the created repository

```
Command Prompt

Using default tag: latest
The push refers to repository [docker.io/shubhid/newrepo]
e402efb6ab11: Layer already exists
8c92fe12e338: Layer already exists
5f70bf18a086: Pushed
a6efb10f6dbb: Layer already exists
97ba0ca3f908: Layer already exists
dc0585a4b8b7: Layer already exists
latest: digest: sha256:aaff9b66d2c203300e070aa3a0c0b8314211818affeae73659b37d0f4165c514 size: 1569

C:\Users\admin\images>docker pull shubhid/newrepo
Using default tag: latest
latest: Pulling from shubhid/newrepo
Digest: sha256:aaff9b66d2c203300e070aa3a0c0b8314211818affeae73659b37d0f4165c514
Status: Image is up to date for shubhid/newrepo:latest
docker.io/shubhid/newrepo:latest
```

Step 7- Now create another container in interactive mode(to see the output) from the pulled image and check whether the output is correct or not.

```
C:\Users\admin\images>docker run -dt --name=newcontainer shubhid/newrepo
fddcaddf6c82ede6f345be7a9d0b16d671623f0d65b5806adb64275277280eb9

C:\Users\admin\images>docker run -it --name=container2 shubhid/newrepo
Hello World

C:\Users\admin\images>
```

You can also check if the repository has been successfully created on your Docker Hub account.