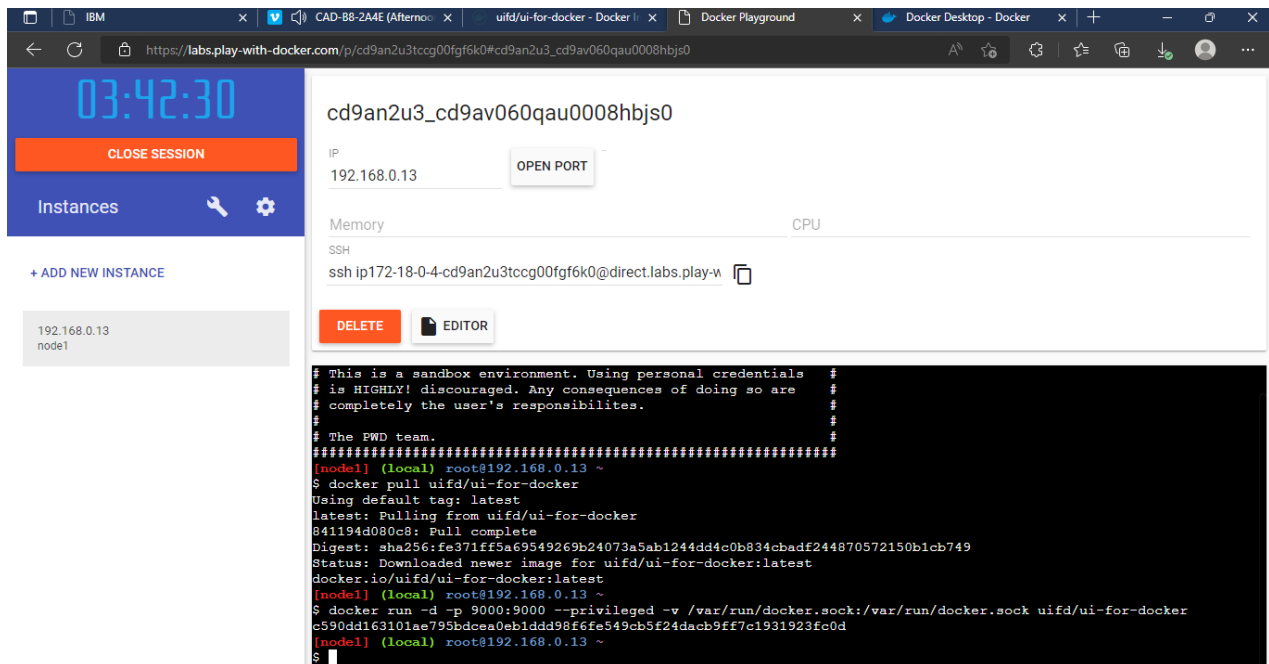


ASSIGNMENT-4

Assignment Date	01 November 2022
Student Name	R ROOBKA
Student Register Number	613019104063
Maximum Marks	2 Marks

Question 1:

Pull an Image from docker hub and run it in docker playground.



```
# This is a sandbox environment. Using personal credentials #
# is HIGHLY! discouraged. Any consequences of doing so are #
# completely the user's responsibilities. #
#
# The FWD team. #
#####
[rodel] (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
841194d080c8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[rodel] (local) root@192.168.0.13 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dcb9ff7c1931923fc0d
[rodel] (local) root@192.168.0.13 ~
$
```

```
PS C:\Windows\system32> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:e18f0a777aefabe047a671ab3ec3eed05414477c951ab1a6f352a06974245fe7
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
PS C:\Windows\system32>
```

UI For Docker

Dashboard Containers Containers Network Images Networks Volumes Info Refresh

UI For Docker


The UI for Docker container engine

Learn more.

Running Containers

- beautiful_goldwasser Up About a minute

Status




UI For Docker

Dashboard Containers Containers Network Images Networks Volumes Info Refresh

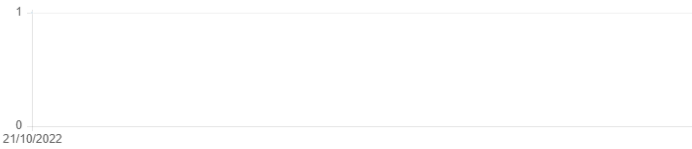
Running Containers

- beautiful_goldwasser Up About a minute


Status



Containers created



Images created



Question 2:

Create a docker file for the job portal application and deploy it in Docker Desktop Application

```
C:\Windows\System32\cmd.exe
>> [internal] load build definition from Dockerfile
>> transferring dockerfile: 32B
>> [internal] load .dockerignore
>> transferring context: 2B
>> [internal] load metadata for docker.io/library/python:3.6
>> [auth] library/python:pull token for registry-1.docker.io
>> [internal] load build context
>> transferring context: 687B
>> [1/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
>> resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
>> sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc 1.86kB / 1.86kB
>> sha256:d097a4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d3b60d 2.22kB / 2.22kB
>> sha256:54260638d07c5e3ad24c6e21fc889abb8486a27634c0892086ff71f3f44b104 9.27kB / 9.27kB
>> sha256:0e29546d541cbbd309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3 54.92MB / 54.92MB
>> sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd 5.15MB / 5.15MB
>> sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56 10.87MB / 10.87MB
>> sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793 54.57MB / 54.57MB
>> sha256:6f9f74896dfa93fe0172f594faba85e0b4e8a0481a0fef9d9112efc7e4d3c78f7 196.51MB / 196.51MB
>> sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743 6.29MB / 6.29MB
>> extracting sha256:0e29546d541cbbd309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3
>> sha256:9fddfd56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 14.21MB / 14.21MB
>> extracting sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd
>> extracting sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56
>> sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7 2358 / 2358
>> sha256:c4f42be2be53b900ebffcc048c1df13de538434ccc5f5d954a56848a6169a3a3f 2.21MB / 2.21MB
>> extracting sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793
>> extracting sha256:6f9f74896dfa93fe0172f594faba85e0b4e8a0481a0fef9d9112efc7e4d3c78f7
>> extracting sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743
>> extracting sha256:9fddfd56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752
>> extracting sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7
>> extracting sha256:c4f42be2be53b900ebffcc048c1df13de538434ccc5f5d954a56848a6169a3a3f
>> [2/6] WORKDIR /app
>> [3/6] ADD . /app
>> [4/6] COPY requirements.txt /app
>> [5/6] RUN python3 -m pip install -r requirements.txt
>> [6/6] RUN python3 -m pip install lbm_db
>> exporting to image
>> exporting layers
>> writing image sha256:4756719486df002fad5dae305c5221513f2ff2d1b49a8d242b22a28af0379f19
>> naming to docker.io/library/job-portal-main

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

C:\Users\VK-PC\Desktop\job-portal-main>
```

FROM helloworld:latest

WORKDIR ~/Desktop/

ADD . helloworld/

WORKDIR ~/Desktop/htmlfile

RUN pip install -r requirements

RUN chmod +x app.sh

CMD ["/bin/sh", "app.sh"]

Question 3:

Create an IBM container registry and deploy helloworld app or Job portal app.

```
PS C:\Users\HP> docker tag hello-world icr.io/0034ns/helloworld
PS C:\Users\HP> docker push icr.io/0034ns/helloworld
Using default tag: latest
The push refers to repository [icr.io/0034ns/helloworld]
e07ee1baac5f: Pushed
latest: digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525
```

Question 4:

Create a Kubernetes cluster in IBM cloud and deploy helloworld image or job portal image and also expose the same app to run in node port.

