

DOCKER AND KUBERNETES

Assignment Date	27 October 2022
Student Name	V SWARNADEVI
Student Roll Number	611819106054
Maximum Marks	2 Marks

1. Pull an image from docker hub and run it in docker Playground

The screenshot shows a web browser with multiple tabs. The active tab is Docker Hub, displaying the repository page for `uifd/ui-for-docker`. The page indicates the repository is deprecated and suggests using Portainer. Below the repository details, there is a section for the Docker Pull Command, which is `docker pull uifd/ui-for-docker`.

The second part of the screenshot shows the Docker Playground interface. It displays a session titled `cd9an2u3_cd9av060qau0008hbjs0` with an IP address of `192.168.0.13`. The interface includes a terminal window where the following commands are executed:

```
# This is a sandbox environment. Using personal credentials
# is HIGHLY discouraged. Any consequences of doing so are
# completely the user's responsibilities.
#
# The PWD team.
#####
[node1] (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
[node1] (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
c590dd163101ae795bdcea0eb1dd498f6fe549cb5f24dab9ff7c1931923fc0d
[node1] (local) root@192.168.0.13 ~
$
```

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



Containers created



Images created



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:d8974a0978ec079df5ac31872359c2de510f02214c04f8e26393b376d3b60d 2.22kB / 2.22kB

→ sha256:5420083807c5e3ad46621fc809abb486a27634c0892086ff71f3444b104 9.27kB / 9.27kB

→ sha256:0e29546d541cddb309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3 54.92MB / 54.92MB

→ sha256:9b829c73b52b92b07d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd 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 sha256:0e29546d541cddb309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3

→ extracting sha256:9b829c73b52b92b07d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd 14.21MB / 14.21MB

→ extracting sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56 2.36 / 2.36

→ sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfee0be0b243b2f31bab7 235B / 235B

→ sha256:c4f42be2be53b90ebfffc040c1df13de538434ccc5f5d954a5684a6169a3a3f 2.21MB / 2.21MB

→ extracting sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793

→ extracting sha256:6f9f74896dfa93fe0172f594faba85e0b4e8a0481a0fef9d9112efc7e4d3c78f7

→ extracting sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743

→ extracting sha256:9b829c73b52b92b07d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd

→ extracting sha256:0e29546d541cddb309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3

→ extracting sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfee0be0b243b2f31bab7

→ extracting sha256:c4f42be2be53b90ebfffc040c1df13de538434ccc5f5d954a5684a6169a3a3f

→ [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 ibm_db

→ exporting layers

→ writing image sha256:1756719486df002fad5dae305c5221513f2ff2d1b49a8d242b22a28af0379f19

→ 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>

Docker Desktop Upgrade plan

Containers

Images

Volumes

Dev Environments BETA

Extensions BETA

Add Extensions

Images on disk

Last refresh: about 1 hour ago 1 Images 0 Bytes total size Refresh to see disk usage Clean up

Images Give feedback

LOCAL REMOTE REPOSITORIES

☐ In use only

NAME	TAG	IMAGE ID	CREATED	SIZE
job-portal-main	latest	1756719486df	less than a minute ago	1.08 GB

RAM 2.53GB CPU 1.56% Connected to Hub v4.13.0

3. Create a IBM container registry and deploy helloworld app