

Assignment -4

Docker and Kubernetes

Assignment Date	21 October 2022
Student Name	Team leader
Maximum Marks	2 Marks

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

The screenshot displays the Docker Hub page for the `uifd/ui-for-docker` image. The page indicates that the repository is deprecated and development continues at `portainer/portainer`. A Docker Pull Command box shows the command: `docker pull uifd/ui-for-docker`.

Below the Docker Hub page, the Docker Playground interface is shown. The instance is named `cd9an2u3_cd9av060qau0008hbjs0` with IP `192.168.0.13`. The terminal output shows the following commands and results:

```
# 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.
#####
[node1] (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
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
e590dd163101ae795bdcea0eb1dd98f6fe549cb5f24dcb9ff7c1931923fc0d
[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



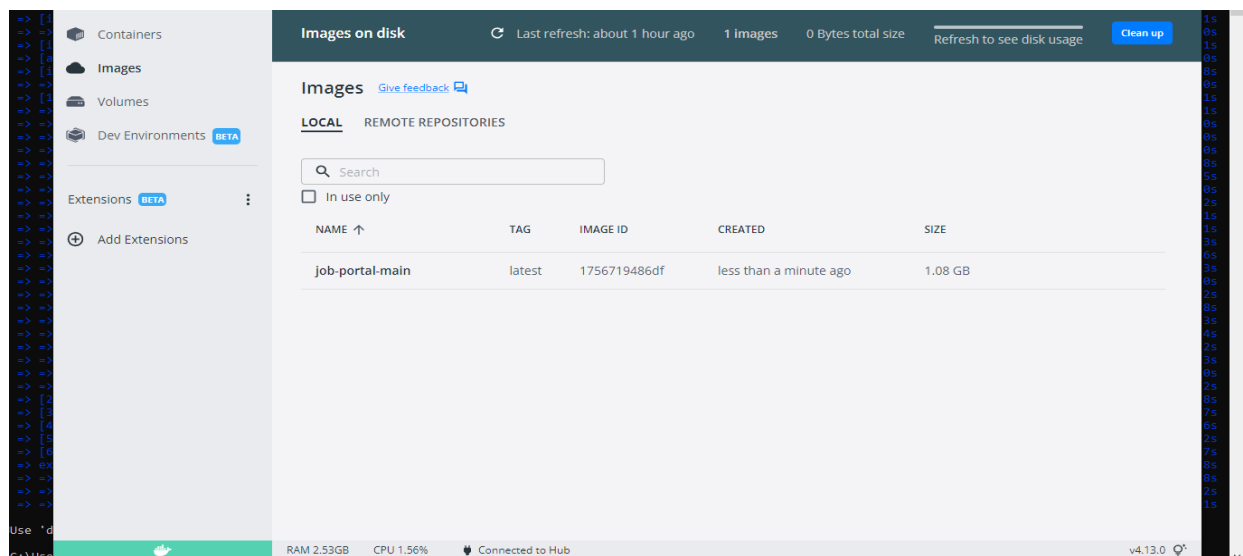
Images created



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:d897a4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d3b6bd 2.22kB / 2.22kB
-> => sha256:54260639d07c5e3ad246621fc0809b8b486a27634c0892080ff7143f44b104 9.27kB / 9.27kB
-> => sha256:0e29546d541c0b30281d1a73a9d1db78665c1b95b74f32b089e0b77a6e1e3 54.92MB / 54.92MB
-> => sha256:9b029c73b52b92b7d5c07a54fb0f3a921995a296c714b53a32ae67d19231fcd 5.15MB / 5.15MB
-> => sha256:cb507ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56 10.87MB / 10.87MB
-> => sha256:6404e4811622b31c027ccac322ca63937fd805f569a93e6f15c01aade718793 54.57MB / 54.57MB
-> => sha256:6f9f74896dfa93fe0172f594faba85e0b4e8a0481a0fef9d112efc7e4d3c78f7 196.51MB / 196.51MB
-> => sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743 6.29MB / 6.29MB
-> => extracting sha256:0e29546d541c0b30281d1a73a9d1db78665c1b95b74f32b089e0b77a6e1e3 27.3s
-> => sha256:9fddfd56334f2e6efad7e241bf5e7459c40ed185c5478676f41c1244bd96752 14.21MB / 14.21MB
-> => extracting sha256:9b029c73b52b92b7d5c07a54fb0f3a921995a296c714b53a32ae67d19231fcd 2.3s
-> => extracting sha256:cb507ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56 4.0s
-> => sha256:404f0204b4ac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7 235B / 235B
-> => sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a5604a6169a3a3f 2.21MB / 2.21MB
-> => extracting sha256:6404e4811622b31c027ccac322ca63937fd805f569a93e6f15c01aade718793 27.3s
-> => extracting sha256:6f9f74896dfa93fe0172f594faba85e0b4e8a0481a0fef9d112efc7e4d3c78f7 131.4s
-> => extracting sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743 8.2s
-> => extracting sha256:9fddfd56334f2e6efad7e241bf5e7459c40ed185c5478676f41c1244bd96752 11.3s
-> => extracting sha256:404f0204b4ac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7 0.0s
-> => extracting sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a5604a6169a3a3f 2.2s
-> [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
-> exporting image
-> writing image sha256:1756710486df002fad5dae305c5221513f2ff2d1b49a8d242b22a28af0379f19
-> naming to docker.io/library/job-portal-main
-> 0.1s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
C:\Users\NVC\PC\Desktop\job-portal-main>
```



3.Create a IBM container registry and deploy helloworld app