

## Assignment -4

### Docker and Kubernetes

Assignment Date	17 NOV 2022
Student Name	JEFNA ASHA J
Student Roll Number	953419104026
Maximum Marks	2 Marks

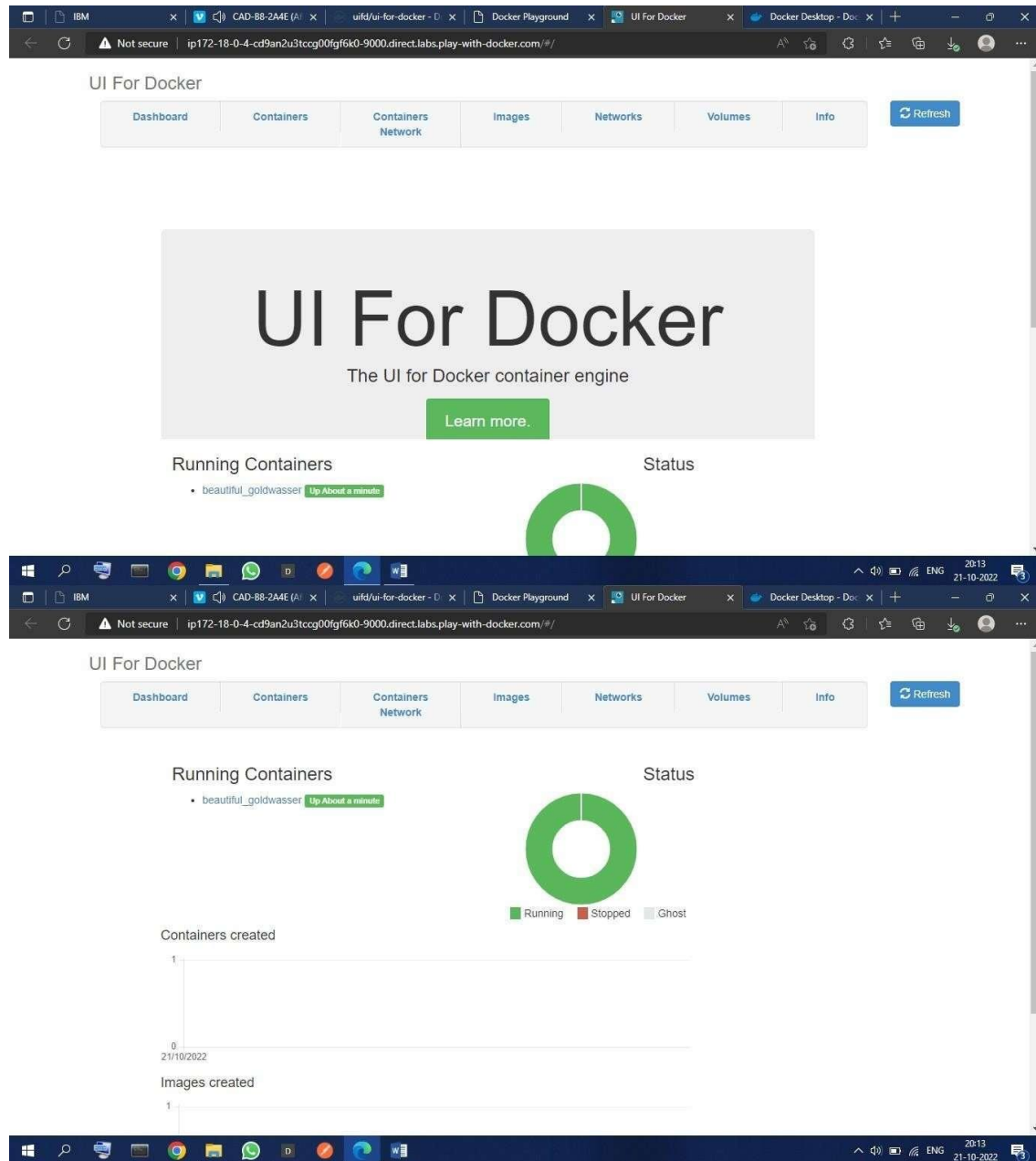
#### 1. Pull an image from Docker hub and run it in Docker Playground

The screenshot displays the Docker Hub page for the `uifd/ui-for-docker` repository. The page indicates that the repository is deprecated and suggests using Portainer instead. The Docker Pull Command is shown as `docker pull uifd/ui-for-docker`.

Below the Docker Hub page, the Docker Playground interface is visible. It shows a session titled `cd9an2u3_cd9av060qau0008hbjs0` with an IP address of `192.168.0.13`. The interface includes buttons for `OPEN PORT`, `DELETE`, and `EDITOR`.

The terminal output within the Docker Playground shows the following commands and results:

```
[node1] (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
341194d080e9: 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
a590dd163101ae793bdece0eb1dd898fefe549cb5f24dab99ff7c1931923fcd
[node1] (local) root@192.168.0.13 ~
```



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:3.8
[auth] library/python:pull token for registry-1.docker.io
-> [Internal] load build context
-> transferring context: 687B
-> [1/6] FROM docker.io/library/python:3.8@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7f9a6019d9f300af6fc
-> resolve docker.io/library/python:3.8@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7f9a6019d9f300af6fc
-> sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7f9a6019d9f300af6fc 1.86kB / 1.86kB
-> sha256:d897a4907a8ec0790f5ac31872359c1de510f0214c0448ae926393b376d306ad 2.22kB / 2.22kB
-> sha256:5426063d007c5e3ad24ce21fc889abbc0486a27634c0802006ff713f3440104 9.27kB / 9.27kB
-> sha256:0e29546d541cddb309281d21a73a9d1db70665c1b95074f320009e0b77a6e1e3 54.02MB / 54.02MB
-> sha256:9b629c73b52b92b97d5c87a54fb0f3e921995a296c714053a32ae67d19231fcd 5.15MB / 5.15MB
-> sha256:cb57ae3e1722407beca53f34823ed21baa85a61d5d95cd8e5a63a746c0d56 19.87MB / 19.87MB
-> sha256:6494e4011622b31c027ccac322ca463937fd0805f50a93e6f15c01ade718793 54.57MB / 54.57MB
-> sha256:6f9f7489d6f93fe0172f594fab85e0b4e8a041a0fef0112efc7e4d3c78f7 196.51MB / 196.51MB
-> sha256:5e3b21213efc50508e78b0d082983945c164de2a37205e06a62dada823124dc743 6.20MB / 6.20MB
-> extracting sha256:0e29546d541cddb309281d21a73a9d1db70665c1b95074f320009e0b77a6e1e3
-> sha256:9fd0f4dc5e334f2e6efad7e341bf5e7459c40ed105c5478070f41c1244bd96752 14.21MB / 14.21MB
-> extracting sha256:0e29546d541cddb309281d21a73a9d1db70665c1b95074f320009e0b77a6e1e3
-> extracting sha256:cb57ae3e1722407beca53f34823ed21baa85a61d5d95cd8e5a63a746c0d56 4.40B
-> sha256:404f02044bac0432ca522cbb9f254b1c91fcea680bfeef0be0b243b2f31bab7 235B / 235B
-> sha256:c4f42be2be53b0e0ebff040c1df13de538434ccc5f5d954a56848a6169a3a3f 2.21MB / 2.21MB
-> extracting sha256:6494e4011622b31c027ccac322ca463937fd0805f50a93e6f15c01ade718793
-> extracting sha256:6f9f7489d6f93fe0172f594fab85e0b4e8a041a0fef0112efc7e4d3c78f7
-> extracting sha256:5e3b21213efc50508e78b0d082983945c164de2a37205e06a62dada823124dc743
-> extracting sha256:9fd0f4dc5e334f2e6efad7e341bf5e7459c40ed105c5478070f41c1244bd96752
-> extracting sha256:404f02044bac0432ca522cbb9f254b1c91fcea680bfeef0be0b243b2f31bab7
-> extracting sha256:c4f42be2be53b0e0ebff040c1df13de538434ccc5f5d954a56848a6169a3a3f
-> [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 img_db
-> exporting to image
-> exporting layers
-> writing image sha256:1756719486df002fad5dae385c5221513f2ff2d1b49a8d242b22a28af0379f19
-> 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>
```

