

Assignment -4

CLOUD APPLICATION DEVELOPEMENT

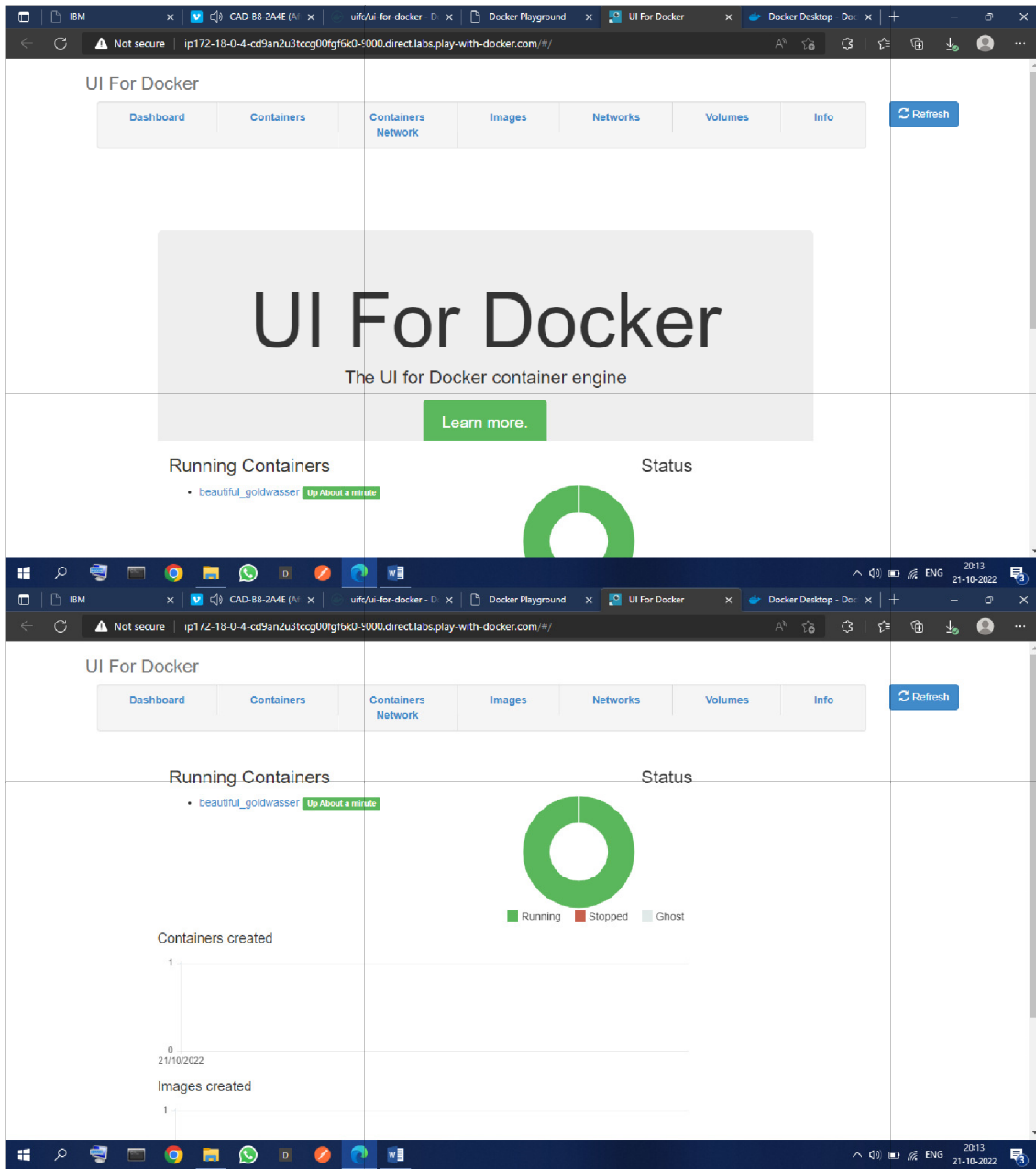
Assignment Date	19 September 2022
Student Name	M.SINDHU
Student Roll Number	812419104062
Maximum Marks	2 Marks

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

The image shows two screenshots. The top screenshot is a browser view of the Docker Hub repository for `uifd/ui-for-docker`. The repository is marked as deprecated, with a note that development continues at `portainer/portainer`. The description states: "UI For Docker is a web interface for the Docker Remote API. The goal is to provide a pure client side implementation so it is effortless to connect and manage docker." The Docker Pull Command is shown as `docker pull uifd/ui-for-docker`.

The bottom screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:42:30, a "CLOSE SESSION" button, and a list of instances. The main area displays the instance details for `cd9an2u3_cd9av060qau0008hbjs0`, including its IP address (192.168.0.13) and an SSH command. Below this, a terminal window shows the execution of the Docker commands:

```
# 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:fe371f3e695492c9b24073a5b1244dd4e0b834cbad5244870572150b1cb749
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/u
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dacb9ff7c1931923fc0d
(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
-> [internal] load .dockerignore
-> [internal] load 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:f852afaf88c25f0d22354d547d802501067aa4036a7fa9a6819d9f30aaf6fc
-> resolve docker.io/library/python:3.6@sha256:f852afaf88c25f0d22354d547d802501067aa4036a7fa9a6819d9f30aaf6fc
-> sha256:f852afaf88c25f0d22354d547d802501067aa4036a7fa9a6819d9f30aaf6fc 1.80kB / 1.80kB
-> sha256:d097ad007a0e079df5ac31872356c2d510f0221ac0448e926303b376d3b0ed 2.22kB / 2.22kB
-> sha256:54208638d07c5e3ad24c6e21fc889abcb8486a27634c892086ff71f3f44b184 0.27kB / 0.27kB
-> sha256:0e29546d541cdd309201d21a73a9d1db78605c1b95b71f32b009e0b77a6e1e3 54.92MB / 54.92MB
-> sha256:9b829c73b52b02b97d5c07a54fb0f3e921995a296c71a53a32ae67d19231fcd 5.15MB / 5.15MB
-> sha256:cb5b7ae361722f070eca53f35823ed21baa85d01d5d95cd5a95ab5d748cd50 10.87MB / 10.87MB
-> sha256:049ae401602031c022c0a22c403937f0005f509a9e0f15c01a0e673073 54.57MB / 54.57MB
-> sha256:09f74800df93f0e172f504fab8e8a8a81a0f0f0d112efc7e4d3c787f 196.51MB / 196.51MB
-> sha256:5e3b1213efc58598e78b0602983945c164de2a37205e05a62da823174d0c743 6.29MB / 6.29MB
-> extracting sha256:0e29546d541cdd309201d21a73a9d1db78605c1b95b71f32b009e0b77a6e1e3
-> sha256:9fddfd5633af2e0efad7e241bf5e7459c48ed185e5473676f41c1244bd96752 14.21MB / 14.21MB
-> extracting sha256:9b829c73b52b02b97d5c07a54fb0f3e921995a296c71a53a32ae67d19231fcd 2.35
-> extracting sha256:cb5b7ae361722f070eca53f35823ed21baa85d01d5d95cd5a95ab5d748cd50 4.08
-> sha256:09f74800df93f0e172f504fab8e8a8a81a0f0f0d112efc7e4d3c787f 225B / 225B
-> sha256:5e3b1213efc58598e78b0602983945c164de2a37205e05a62da823174d0c743 2.21MB / 2.21MB
-> extracting sha256:6404a481162b31c027ccac322ca463037fd805f560a93e6f15c01aade718703 27.35
-> extracting sha256:6f9f74896df93f0e172f504fab8e8a8a81a0f0f0d112efc7e4d3c787f 131.45
-> extracting sha256:5e3b1213efc58598e78b0602983945c164de2a37205e05a62da823174d0c743 0.25
-> extracting sha256:9fddfd5633af2e0efad7e241bf5e7459c48ed185e5473676f41c1244bd96752 11.25
-> extracting sha256:404f010404ac0421c521c0b97254b1cd1fca6806b9ef0b0e0243b2f31ba07 104.25
-> extracting sha256:ic4f42be2b53b000ebff040c1df13de53843ccc5f5d954a56848a6169a3af 2.25
-> [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 lxml
-> exporting to image
-> exporting layers
-> writing image sha256:1756719486df002fadsdae305c5221513f2ff2d1b49a8d42b22a28af0379f19
-> 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>

```