


DOCKER AND KUBERNETES

Student Name	R Hariharan
Student Roll Number	91761915016
Team ID	PNT2022TMID06095
Maximum Marks	2 Marks

Question 1:

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



uifd/ui-for-docker ☆

By [uifd](#) • Updated 6 years ago

A web interface for Docker, formerly known as DockerUI. Deprecated, use Portainer for new features.

[Other](#) [Image](#)

Pulls 10M+

Overview Tags

UI For Docker

This repo is deprecated. Development continues at: [portainer/portainer](#)

[chat on gitter](#)

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.

Docker Pull Command

```
docker pull uifd/ui-for-d
```

03:42:30

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.13
node1

cd9an2u3_cd9av060qau0008hbjs0

IP
192.168.0.13

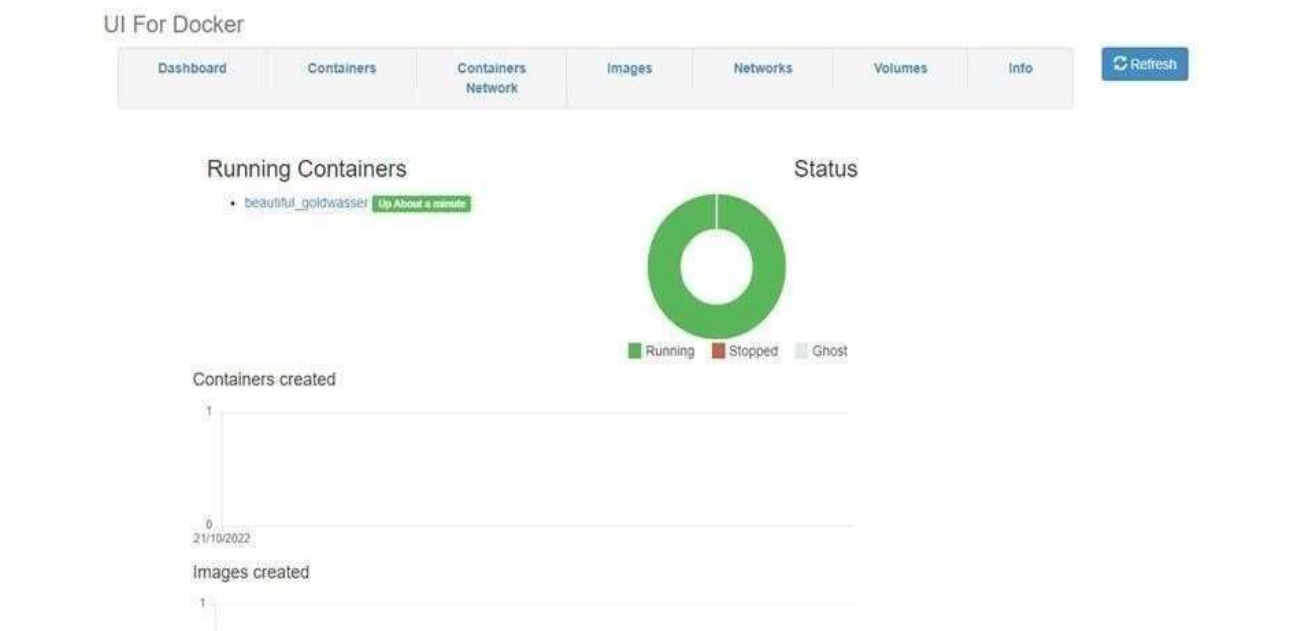
OPEN PORT

Memory CPU

SSH
ssh ip172-18-0-4-cd9an2u3tccg00fgf6k0@direct.labs.play-w

DELETE EDITOR

```
# 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
411194d080c8: 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
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dadb9ff7c1931923fcd
(node1) (local) root@192.168.0.13 ~
$
```

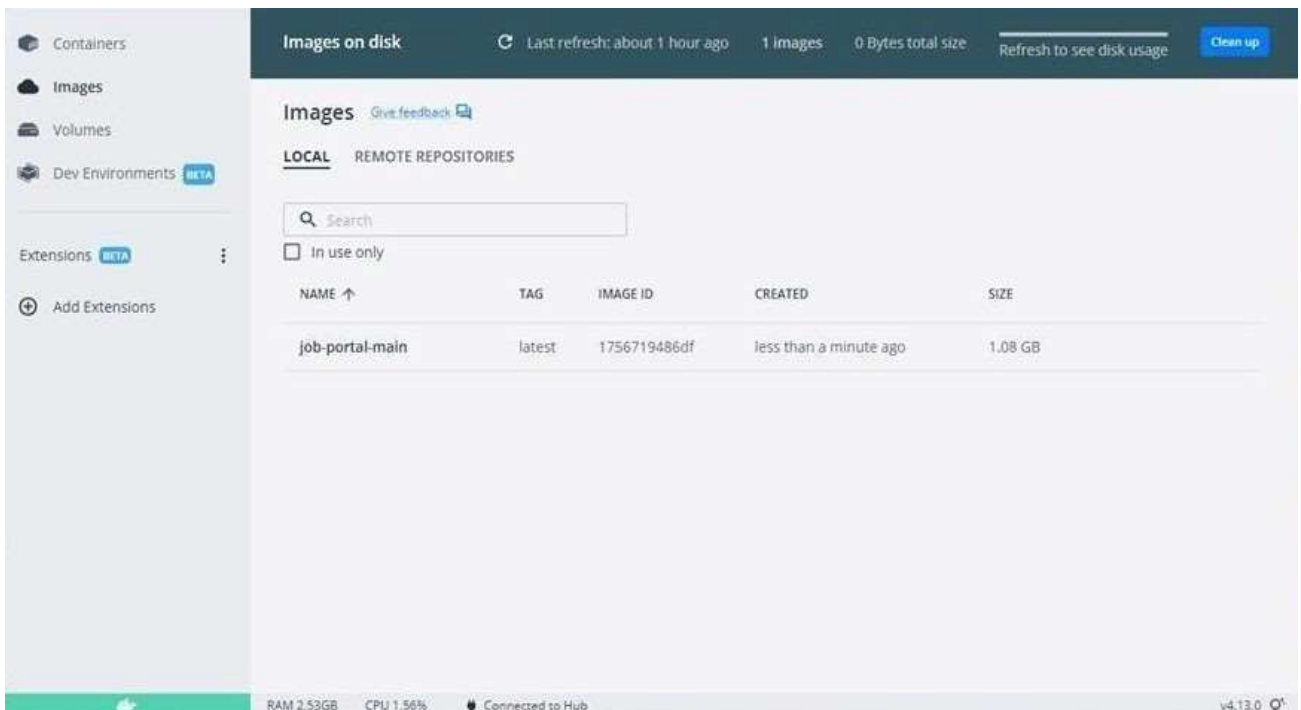


Question 2:

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

```
[internal] load build definition from Dockerfile
--> transferring dockerfile: 12B
[internal] load .dockerignore
--> transferring context: 3B
[internal] load metadata for docker.io/library/python:3.9
[auth] library/python:pull token for registry-1.docker.io
[internal] load build context
--> transferring context: 887B
[1/6] FROM docker.io/library/python:3.9@sha256:f8522afaf86c35f0d22354d547d892591067aa4826a7faad819d9f380afefc
--> resolve docker.io/library/python:3.9@sha256:f8522afaf86c35f0d22354d547d892591067aa4826a7faad819d9f380afefc
--> sha256:f8522afaf86c35f0d22354d547d892591067aa4826a7faad819d9f380afefc 1.80kB / 1.80kB
--> sha256:8997a907a8e07c0f5ac31072350c2d05189f92234c0448a9080b0376d3106d 2.22kB / 2.22kB
--> sha256:5420803087c5e3ed34c0a219c80a8b1c802a27654c8092086ff71f3f44c104 9.27kB / 9.27kB
--> sha256:8e20546d541cd0d109281a21a73a0d1d670685c1b93b74f31000000077ade1e1 54.92MB / 54.92MB
--> sha256:9082ac73b52092897d5c87a5af0bf3e921995a209c714b53a32aeb019231fcd 5.15MB / 5.15MB
--> sha256:c0b3aee361722f07eac59f3083ed21baa65d61d5d95cd5a95d53d74budd55 19.87MB / 19.87MB
--> sha256:8496a83116231c0d7cc322ca403937fd00c5f00a93a6f15c01aade718793 54.57MB / 54.57MB
--> sha256:c9f77400efaf93fe0172f594fab250b46a0481a0f6d9112efc7e4d3c78f7 106.51MB / 106.51MB
--> sha256:5a312133efc5598c70bd002983045c104a63a37209e06a621ada823124dc743 6.20MB / 6.20MB
--> extracting sha256:8e20546d541cd0d109281a21a73a0d1d670685c1b93b74f31000000077ade1e1 14.21MB / 14.21MB
--> sha256:9fd0fd5033872e6fa7a2410f5e7450c40ed105c5470676741c1244bd06752 14.21MB / 14.21MB
--> extracting sha256:9082ac73b52092897d5c87a5af0bf3e921995a209c714b53a32aeb019231fcd 2.33
--> extracting sha256:c0b3aee361722f07eac59f3083ed21baa65d61d5d95cd5a95d53d74budd55 4.01
--> sha256:403101044bac032a321c00f25401c91fcea0000f00b0c243d2f31bab7 235B / 235B
--> sha256:4462b2be530000ff1c04bc10f13de53043cc5f5d054a3004bae169a3a3f 2.21MB / 2.21MB
--> extracting sha256:403101044bac032a321c00f25401c91fcea0000f00b0c243d2f31bab7 27.37
--> extracting sha256:c0b3aee361722f07eac59f3083ed21baa65d61d5d95cd5a95d53d74budd55 8.25
--> extracting sha256:5a312133efc5598c70bd002983045c104a63a37209e06a621ada823124dc743 11.34
--> extracting sha256:9fd0fd5033872e6fa7a2410f5e7450c40ed105c5470676741c1244bd06752 11.35
--> extracting sha256:8496a83116231c0d7cc322ca403937fd00c5f00a93a6f15c01aade718793 6.86
--> extracting sha256:c9f77400efaf93fe0172f594fab250b46a0481a0f6d9112efc7e4d3c78f7 2.22
--> [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 flask_db
--> exporting to image
--> exporting layers
--> writing image sha256:1756719486df002fa0d0a0305c5221513f2f32d1b4a80242022a20a90370f19
--> 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>
```



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.

