

### **DOCKER AND KUBERNETES**

Assignment Date	15 NOVEMBER 2022
Student Name	SUNANTH V
Student Roll Number	AC19UIT044
Maximum Marks	2 Marks

- 1. Pull an image from docker hub and run it in docker Playground and**
- 2. Create a docker file for the job portal application and deploy it in Docker desktop application**

The screenshot is divided into two main horizontal sections. The top section shows the Docker Hub repository page for `uifd/ui-for-docker`. The page includes the repository name, a star icon, and a note stating "This repo is deprecated. Development continues at: portainer/portainer". A "chat on gitter" button is visible. The "Overview" tab is selected, showing a description of the web interface for the Docker Remote API. A "Goals" section is partially visible. To the right, the "Tags" section is empty. Below the repository name, there is a "UI For Docker" section with a note about deprecation and a "chat on gitter" button. A "Goals" section is also present. To the right of the repository page, the "Docker Pull Command" is displayed as `docker pull uifd/ui-for-docker`. The bottom section shows the Docker Playground interface. The top bar displays the time as 03:42:30 and a "CLOSE SESSION" button. Below this, the "Instances" section shows a list of instances, including one named "node1" with IP address "192.168.0.13". The main area of the playground shows the details of the selected instance, including its IP address "192.168.0.13", memory usage, CPU usage, and an SSH command: `ssh ip172-18-0-4-cd9an2u3tccg00fgf6k0@direct.labs.play-with-docker.com`. Below the instance details, there is a terminal window showing the execution of the `docker pull uifd/ui-for-docker` command and the subsequent `docker run` command to start the container. The terminal output shows the container pulling the latest image and running successfully.

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

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.

Goals

Docker Pull Command

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

03:42:30

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.13  
node1

cd9an2u3\_cd9av060qau0008hbjso

IP: 192.168.0.13 OPEN PORT

Memory CPU

SSH

```
ssh ip172-18-0-4-cd9an2u3tccg00fgf6k0@direct.labs.play-with-docker.com
```

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 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
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
c590dd163101ae798bdca0eb1dd98f6fe549cb5f24dadb9ff7c1931923fcd0d
[node1] (local) root@192.168.0.13 ~
$
```

3.Create an IBM container registry and deploy Helloworld or job portal app.

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




UI For Docker

Dashboard Containers Containers Network Images Networks Volumes Info Refresh

Running Containers

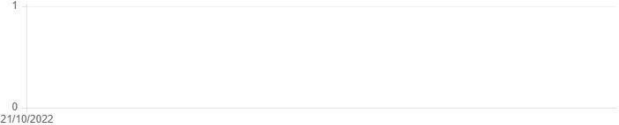
- beautiful\_goldwasser Up About a minute

Status




Running Stopped Ghost

Containers created



Images created

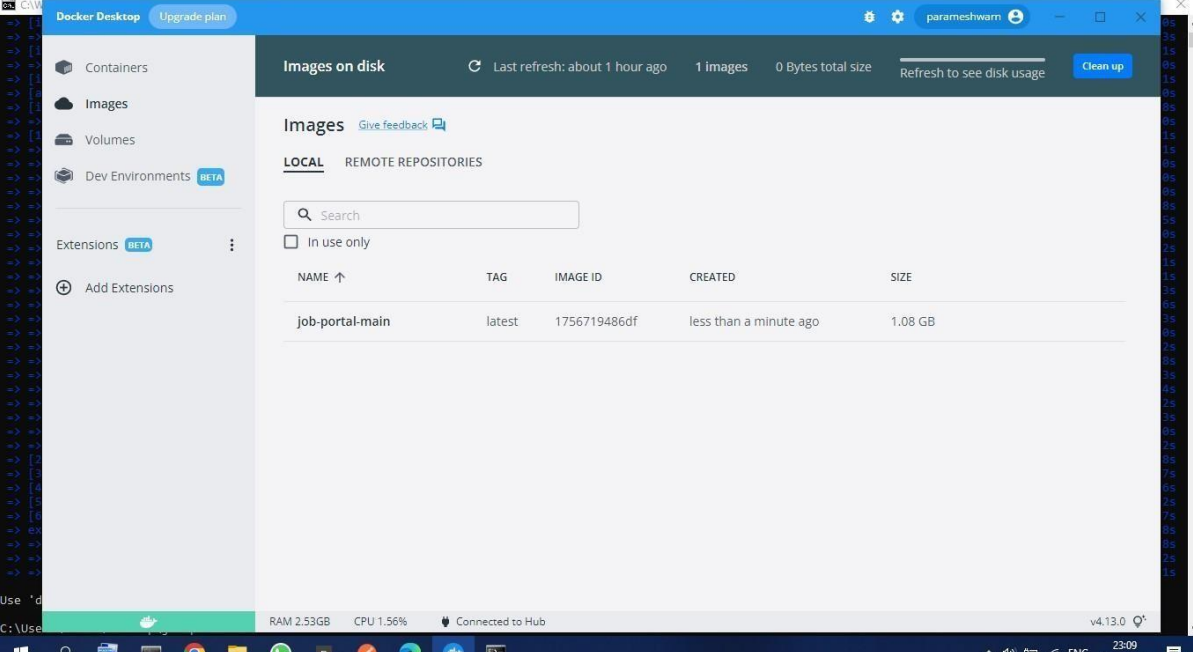


#### 4. Create a Kubernetes cluster in IBM cloud and deploy Helloworld image or job portal app image and also expose the same app to run in no deport.

```
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:f8652afa88c25f8d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
-> resolve docker.io/library/python:3.6@sha256:f8652afa88c25f8d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
-> sha256:f8652afa88c25f8d22354d547d892591067aa4026a7fa9a6819df9f300af6fc 1.86kB / 1.86kB
-> sha256:d097f4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d3b60d 2.22kB / 2.22kB
-> sha256:54260638d07c5e3ad24c6e21fc889abbcc8486a27634c0802086ff71f3f44b104 9.27kB / 9.27kB
-> sha256:0e29546d541c0b380261d21a73a941db7865c1b95b74f326a09e077a0e1e3 54.92MB / 54.92MB
-> sha256:0a029c73b52b92b97d5c07e5410b3a921995a206c714b53a32ae67d10211fcd 5.15MB / 5.15MB
-> sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d05cd5a95ab53d740cdd56 10.87MB / 10.87MB
-> sha256:6494e4811622b31c027ccc322ca463937fd805f569a93e6f15c01aade718793 54.57MB / 54.57MB
-> sha256:6f9f74896df9a3fe0172f504fab85e0b4e8a041a0efed9112efc7e4d3c78f7 196.51MB / 196.51MB
-> sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743 6.29MB / 6.29MB
-> extracting sha256:0e29546d541c0b380261d21a73a941db7865c1b95b74f326a09e077a0e1e3
-> sha256:9fddfd5c56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 14.21MB / 14.21MB
-> extracting sha256:90829c73b52b92b97d5c07e5410b3a921995a206c714b53a32ae67d10211fcd
-> extracting sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d05cd5a95ab53d740cdd56
-> sha256:404f02044bac0432ca522cbb9f254b1c91fca68006bfeef0be0b243b2f31bab7 2350 / 2350
-> sha256:c4f42be2be53b900ebffcc048c1df13de538434ccc5f5d954a56848a6109a3a3f 2.21MB / 2.21MB
-> extracting sha256:6494e4811622b31c027ccc322ca463937fd805f569a93e6f15c01aade718793
-> sha256:6f9f74896df9a3fe0172f504fab85e0b4e8a041a0efed9112efc7e4d3c78f7
-> extracting sha256:5e3b1213efc56598e78bd602983945c164de2a37205e06a62dada823124dc743
-> extracting sha256:9fddfd5c56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752
-> extracting sha256:404f02044bac0432ca522cbb9f254b1c91fca68006bfeef0be0b243b2f31bab7
-> extracting sha256:c4f42be2be53b900ebffcc048c1df13de538434ccc5f5d954a56848a6109a3a3f
-> [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 to image
-> exporting layers
-> writing image sha256:1756719486df082fad5d9e305c5221513f2ff2d1b49a8d242b22a28af0379f19
-> 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>
```



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

**Create a IBM container registry and deploy hello word app**