

## Assignment - 4

### Docker and Kubernetes

Assignment Date	November 3
Student Name	Nithish J
Student Roll Number	310819106057
Maximum Marks	2 Marks

#### Question-1:

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

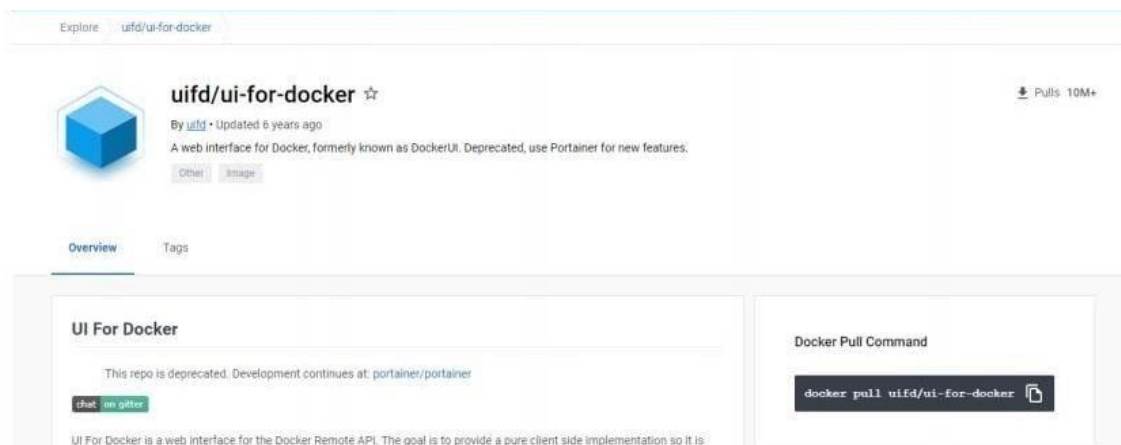
Solution:

```
docker run --rm -p 8787:8787 rocker/verse
docker pull rocker/verse
docker login --username=nithishj --email=nithishjaganathan@gmail.com
WARNING: login credentials saved in /home/nithishj/.docker/config.json
Login Succeeded
```

```
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
verse_gapminder_gsl latest  023ab91c6291  3 minutes ago  1.975 GB
verse_gapminder      latest  bb38976d03cf  13 minutes ago 1.955 GB
rocker/verse         latest  0168d115f220  3 days ago    1.954 GB
docker tag bb38976d03cf nithishj
/verse_gapminder:firsttry
docker push nithishj
/verse_gapminder
```

Saving and loading images

```
docker save
verse_gapminder
docker save verse_gapminder > verse_gapminder.tar
docker load --input verse_gapminder.tar
docker load --input verse_gapminder.tar
```



Explore uifd/ui-for-docker

**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 github

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

**Docker Pull Command**

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

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.
#####
root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
41194d9008: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
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
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dadb9ff7c1931923fc0d
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

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

1

0

21/10/2022

Images created

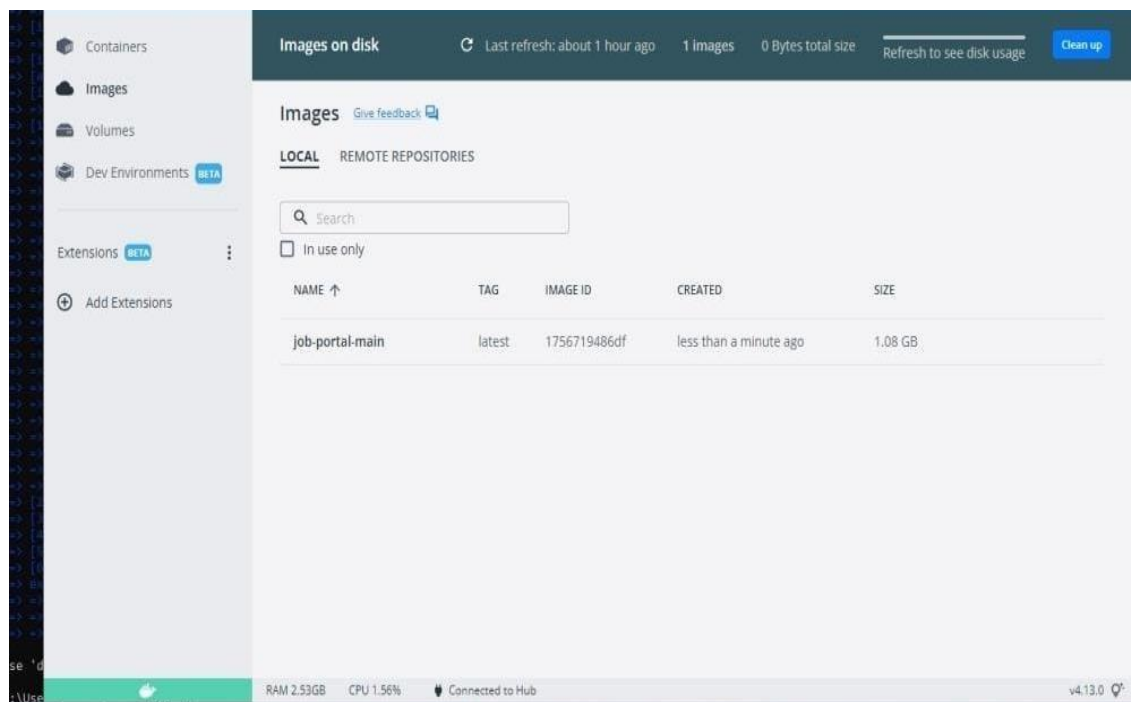
1

## Question-2:

2. Create a docker file for the jobportal application and deploy it in Docker desktop application.

## SOLUTION:

```
[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:f80521f8f88c25f0d22354d547d892591867aa4026a7fa9a6810df9f300af0fc
--> resolve docker.io/library/python:3.6@sha256:f80521f8f88c25f0d22354d547d892591867aa4026a7fa9a6810df9f300af0fc
--> sha256:f80521f8f88c25f0d22354d547d892591867aa4026a7fa9a6810df9f300af0fc 1.86kB / 1.86kB
--> sha256:409774097080c879d75ac31872359c2de519f82214c0448a926393b376d3b60d 2.22kB / 2.22kB
--> sha256:54268638097c5a3ad24c6e21fc889abbc8486a27634c8891080ff71f3f44b184 9.27kB / 9.27kB
--> sha256:6c29566d541c8d309281d21a73a9d1db786651b05b74f32b00e0b77ade1e3 54.92MB / 54.92MB
--> sha256:90829c73b521b97d5c07a54f0f3e921995a296c714b51a32ae67d10231fcd 5.15MB / 5.15MB
--> sha256:c5b7ae3b1722f070eca53f35823ed21ba085d61d5d95cd5a95ab53d746cdd56 10.87MB / 10.87MB
--> sha256:6494e4811622b31c027ccac322ca463927f0805f569a93e6f15c01aade7318793 54.57MB / 54.57MB
--> sha256:6f9f74896d7a93fe8172f594fab085eb4e8a0401a0f0d9112efc7e4d3c78f7 196.51MB / 196.51MB
--> sha256:5e3b1213efc0559b70b08030834c164de2a37205e06e61ada823124de743 6.29MB / 6.29MB
--> extracting sha256:6e29566d541c8d309281d21a73a9d1db786651b05b74f32b00e0b77ade1e3 27
--> sha256:9fd9fd636334f2e0efad7e261bf5e7459c48ed195c5478676f61c1044bd96752 14.21MB / 14.21MB
--> extracting sha256:90829c73b521b97d5c07a54f0f3e921995a296c714b51a32ae67d10231fcd 2
--> extracting sha256:cb5b7ae3b1722f070eca53f35823ed21ba085d61d5d95cd5a95ab53d746cdd56 4
--> sha256:404f02044bac0432ca522cbb9f254b1c91fca0806bfeef0be0b243b2f31bab7 235B / 235B
--> sha256:c4f42be2b53b900ebffc048c1df13de538434ccc5f5d954a56884a6109a3af 2.21MB / 2.21MB
--> extracting sha256:6494e4811622b31c027ccac322ca463927f0805f569a93e6f15c01aade7318793 27
--> extracting sha256:6f9f74896d7a93fe8172f594fab085eb4e8a0401a0f0d9112efc7e4d3c78f7 191
--> extracting sha256:5e3b1213efc0559b70b08030834c164de2a37205e06e61ada823124de743 5
--> extracting sha256:9fd9fd636334f2e0efad7e261bf5e7459c48ed195c5478676f61c1044bd96752 11
--> extracting sha256:404f02044bac0432ca522cbb9f254b1c91fca0806bfeef0be0b243b2f31bab7 6
--> extracting sha256:c4f42be2b53b900ebffc048c1df13de538434ccc5f5d954a56884a6109a3af 2
[2/6] WORKDIR /app
--> [2/6] WORKDIR /app 2
[3/6] ADD . /app
--> [3/6] ADD . /app 2
[4/6] COPY requirements.txt /app
--> [4/6] COPY requirements.txt /app 2
[5/6] RUN python3 -m pip install -r requirements.txt
--> [5/6] RUN python3 -m pip install -r requirements.txt 372
[6/6] RUN python3 -m pip install job
--> [6/6] RUN python3 -m pip install job 9
exporting to image
--> exporting image 2
--> exporting layers 2
--> writing image sha256:1756719486df002fad5dae305c5221513f2ff3d1b49a8d242b22a28e78379f19 6
--> naming to docker.io/library/job-portal-main 6
se 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them 6
```



### QUESTION-3:

3. Create a IBM container registry and deploy helloworld app or jobportalapp.

Solution:

```
<html>
<body>
  Hello, IBM Cloud World!
</body>
</html>---
```

applications:

```
- buildpack: https://github.com/cloudfoundry/staticfile-buildpack.git
host: simple-website-`${random}`
name: simple-website-`${random}`
memory: 64M
stack: cflinuxfs2
```

The screenshot shows the IBM Cloud Deploy console. At the top, there's a 'DEPLOY' header with a 'DELETE' button. Below it are tabs for 'INPUT', 'JOBS', and 'ENVIRONMENT PROPERTIES'. The 'JOBS' tab is active, showing a 'Rolling Deploy' section with a 'ROLLING DEPLOY' button. Below this, there's a 'Deploy configuration' section with several fields: 'Deployer type' (Cloud Foundry), 'IBM Cloud region' (US South - https://api.ng.bluemix.net), 'Organization' (bluemix\_devops@ibm.com), 'Space' (demo), and 'Application name' (simple-website-ae7f5ff6). Each field has a dropdown arrow and an information icon.

```
1  {
2    "ServiceId": "com.ibm.cloudoe.orion.client.deploy",
3    "Params": {
4      "Target": {
5        "Url": "https://api.ng.bluemix.net",
6        "Org": "bluemix_devops@ibm.com",
7        "Space": "demo"
8      },
9      "Name": "simple-website-ae7f5ff6",
10     "Instrumentation": {}
11   },
12   "Path": "manifest.yml",
13   "Type": "Cloud Foundry"
14 }
```

Hello, IBM Cloud World!

#### QUESTION-4:

4. Create a Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the same app to run in nodeport.

Solution:

```
ibmcloud target -g <resource_group_name>ibmcloud cr nithishj-add <your_nithishj>ibmcloud  
resource service-instance-create example-postgresql databases-for-postgresql standard us-  
southibmcloud ks cluster-service-bind mycluster default example-postgresqlgit clone -b node  
git@github.com:IBM-Cloud/cloudatabases-helloworld-kubernetes-examples.gitspec:
```

```
replicas: 3name: cloudpostgres-nodejs-app
```

```
image: "registry.<region>.bluemix.net/<namespace>/icdpg" # Edit me
```

```
imagePullPolicy: Alwaysibmcloud cr regionYou are targeting region 'us-south', the registry is  
'registry.ng.bluemix.net'.ibmcloud cr build -t registry.ng.bluemix.net/<namespace>/icdpg .ibmcloud  
cr images
```

env:

```
- name: BINDING
```

```
valueFrom:
```

```
secretKeyRef:
```

```
name: <postgres-secret-name> # Edit me
```

```
key: binding
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
name: cloudpostgres-service
```

```
labels:
```

```
run: clouddb-demo
```

```
spec:
```

```
type: NodePort
```

```
selector:
```

```
run: clouddb-demo
```

```
ports:
```

```
- protocol: TCP
```

```
port: 8080
```

```
nodePort: 30081
```

```
kubectl apply -f clouddb-deployment.yml
```

```
deployment.apps/icdpostgres-app created
```

```
service/cloudpostgres-service created
```

```
kubectl get pods -o wideibmcloud ks workers <your_cluster_name>
```

