

Assignment - 4

Docker and Kubernetes

Assignment Date	November 3
Student Name	Bharath Kishore R
Student Roll Number	73151913013
Maximum Marks	2 Marks

Question-1:

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

Solution:

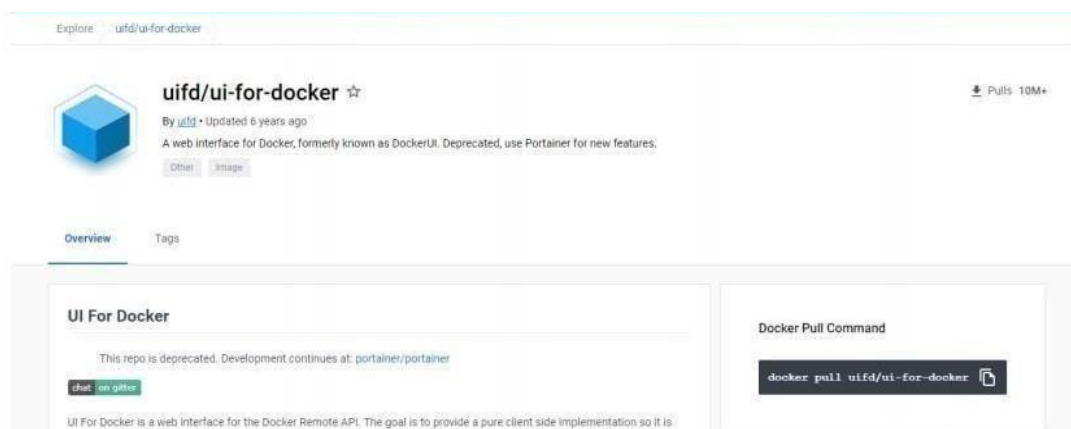
```
docker run --rm -p 8787:8787 rocker/verse
docker pull rocker/verse
docker login --username=nishanthc --email=ssnehasri178@gmail.com
WARNING: login credentials saved in
/home/nishanthc/.docker/config.jsonLogin Succeeded
```

```
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 nishanthc
/verse_gapminder:firsttry
docker push nishanthc
/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](#)

Pulls 10M+

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. #
#####
(node1) (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
41194d080c8: Pull complete
Digest: sha256:fe371ff3ae69549269b24073a5ab1244dd4ec0b834cbadf244870572150b1cb749
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
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dcb9ff7c1931923fc0d
(node1) (local) 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

21/10/2022

Images created

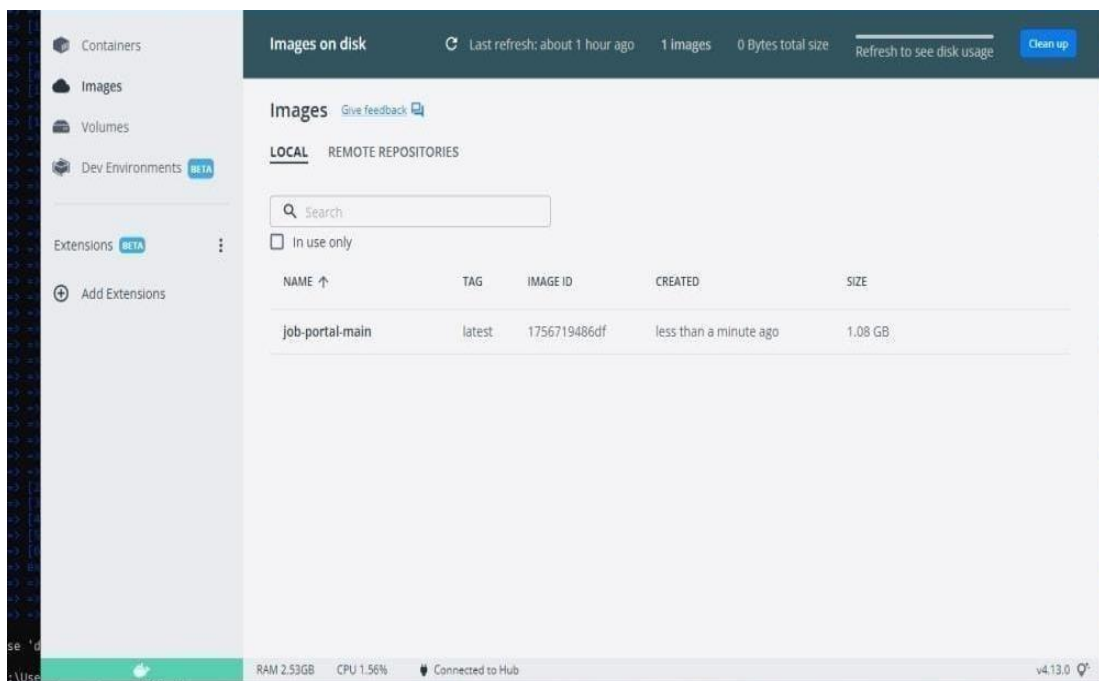
1

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
[5/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c35f0d22354d547d892591067aa4826a7fa9a810df0f308af0fc
--> resolve docker.io/library/python:3.6@sha256:f8652afaf88c35f0d22354d547d892591067aa4826a7fa9a810df0f308af0fc
--> sha256:f8652afaf88c35f0d22354d547d892591067aa4826a7fa9a810df0f308af0fc 1.86kB / 1.86kB
--> sha256:8007a4097a8ec879dF5ac31872359c2de510FB2214c0448e92630b376d3b060d 2.22kB / 2.22kB
--> sha256:54268638007c5e3ad2Ac6e21fc889abbc8486a27634c8892086ff71f3f44b104 9.27kB / 9.27kB
--> sha256:8e29546d541cdd389281d21a73e9d1db78665c1b95b74f32b000e0b77a6e1e3 54.92MB / 54.92MB
--> sha256:8b29c73b52b02b07d5c07a54f0f3e921995a296c714b53a32ae67d192331fcd 5.15MB / 5.15MB
--> sha256:c8b07ae381723f078ecac3f35823ed21ba85d61d5a95cd5a95ab53d746cd056 10.87MB / 10.87MB
--> sha256:c4944a811622b310027cacc32c445937f600cf569a930ef35c81aad0718793 54.57MB / 54.57MB
--> sha256:6f9f74886df493fe0172f504fab85e0b4e8a8481d0fef0112af764d3c70f2 106.51MB / 106.51MB
--> sha256:5e3b1213efc56598e78bd002983045c164de2a37205e06a62dadab23124dc743 6.29MB / 6.29MB
--> extracting sha256:8e29546d541cdd389281d21a73e9d1db78665c1b95b74f32b000e0b77a6e1e3
--> sha256:9fd9dc56334f2eefad7e241bf5e7459c40ed105c5478676f41c1244bd06752 14.21MB / 14.21MB
--> extracting sha256:8b29c73b52b02b07d5c07a54f0f3e921995a296c714b53a32ae67d192331fcd
--> extracting sha256:c8b07ae381723f078ecac3f35823ed21ba85d61d5a95cd5a95ab53d746cd056
--> sha256:404f02044bac0432ca322c0b0f254b1c91fca6886bfeef0be0b243b2f31bab7 235B / 235B
--> sha256:c4f42be2bc57e000ebffca8bc1df13de53824ccc5f5d954a5084bae109a3af 2.21MB / 2.21MB
--> extracting sha256:64944a811622b310027cacc32c445937f600cf569a930ef35c81aad0718793
--> extracting sha256:6f9f74886df493fe0172f504fab85e0b4e8a8481d0fef0112af764d3c70f2
--> extracting sha256:5e3b1213efc56598e78bd002983045c164de2a37205e06a62dadab23124dc743
--> extracting sha256:9fd9dc56334f2eefad7e241bf5e7459c40ed105c5478676f41c1244bd06752
--> extracting sha256:404f02044bac0432ca322c0b0f254b1c91fca6886bfeef0be0b243b2f31bab7
--> extracting sha256:c4f42be2bc57e000ebffca8bc1df13de53824ccc5f5d954a5084bae109a3af
[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 shw_db
exporting to image
--> exporting layers
--> writing image sha256:1756719486df002fad5dae305c5221513f2ff2d1b49a8d242b22a28ef0379f19
--> naming to docker.io/library/job-portal-main
se 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```



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- $\{\text{random}\}$
- name: simple-website- $\{\text{random}\}$
- memory: 64M
- stack: cflinuxfs2

The screenshot shows the 'DEPLOY' tab in the IBM Cloud console. It includes a 'DELETE' button and tabs for 'INPUT', 'JOBS', and 'ENVIRONMENT PROPERTIES'. Below these are icons for 'Rolling De...' and 'ADD JOB'. The 'Rolling Deploy' section contains a 'REMOVE' button and a 'Deploy configuration' table with the following details:

Deploy configuration
Deployer type (1)
Cloud Foundry
IBM Cloud region (1)
US South - https://api.ng.bluemix.net
Organization (1)
bluemix_devops@ibm.com
Space (1)
demo
Application name (1)
simple-website-ae7f5ff6

```
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 nishanthc-add  
<your_nishanthc>ibmcloudresource 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>
```

