

Assignment - 4

Docker and Kubernetes

Assignment Date	November 3
Student Name	Ananthi S
Student Roll Number	2116190701017
Maximum Marks	2 Marks

Question-1:

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

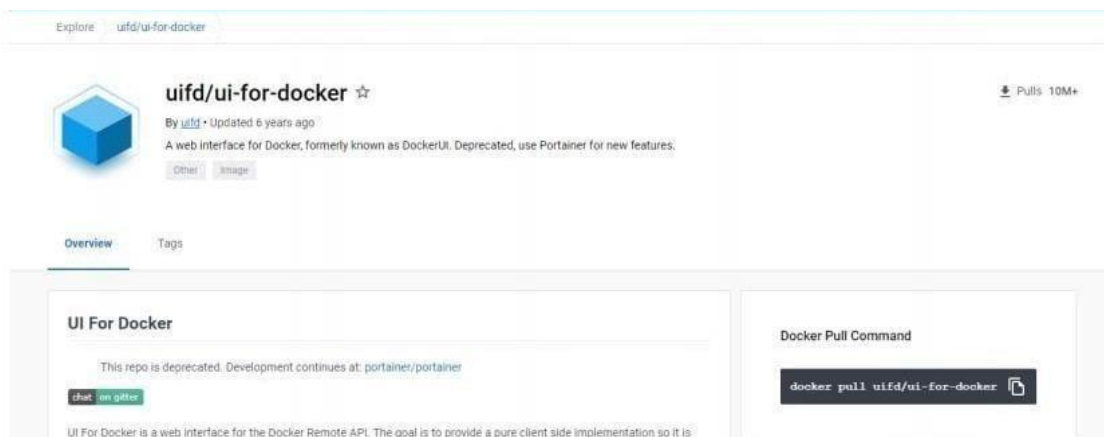
Solution:

```
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 ☆ Pulls 10M+

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

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 FWD team. #
#####
[ui-for-docker] (local) root@192.168.0.13 ~
$ docker pull ui-for-docker
Using default tag: latest
latest: Pulling from ui-for-docker
41194d080e8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244d4e0b834cbadf244870572150b1cb749
Status: Downloaded newer image for ui-for-docker:latest
docker.io/ui-for-docker:latest
[ui-for-docker] (local) root@192.168.0.13 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock ui-for-docker
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dadb9ff7c1931923fc0d
[ui-for-docker] (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

0

21/10/2022

Images created

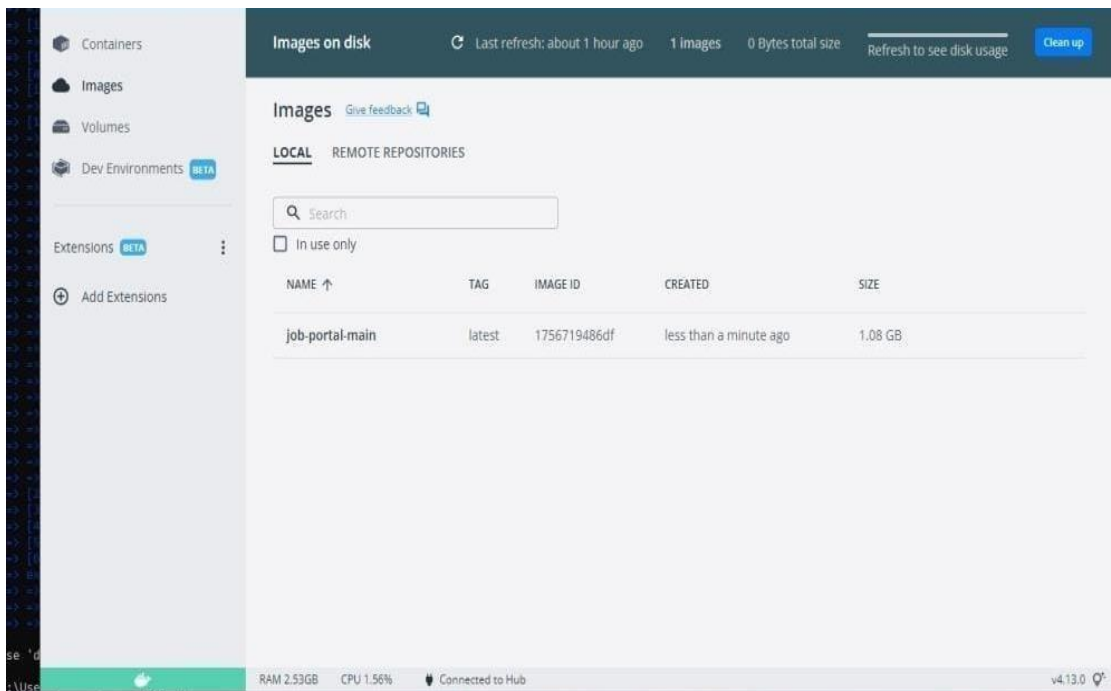
1

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:f8652afef88c25f0d22354d547d802591067aa4026a7fa0a019df9f300af0fc
=> resolve docker.io/library/python:3.6@sha256:f8652afef88c25f0d22354d547d802591067aa4026a7fa0a019df9f300af0fc
=> sha256:f8652afef88c25f0d22354d547d802591067aa4026a7fa0a019df9f300af0fc 1.86kB / 1.86kB
=> sha256:0097a4907a8ec879d75ac31872359c2de510f82214c04a8e926393b376d3b00d 2.22kB / 2.22kB
=> sha256:5426003807c5e3ad24c6e21fc889ab0c6486a27634c002000ff71f3f44d104 0.27kB / 0.27kB
=> sha256:0e29946e541c0dd09281d21a77ae01db7865c1095074f32e00e06774ee1e9 54.92MB / 54.92MB
=> sha256:0920c7302090b07d0e07a04f00fa21a95a206c714b53a22ee0709231fcd 5.15MB / 5.15MB
=> sha256:cb0b7ae361722f0700ca3f35823ed21ba085d61d509c05a05ab53d740c0d56 10.87MB / 10.87MB
=> sha256:6494a4811622b31c027ccac322ca63937fd005f509a9306f15c01aade718793 54.57MB / 54.57MB
=> sha256:6f9f74800d7a93fe0172f594fab05e0b4e0a041a0fe09112efc7e4d3c70f7 106.51MB / 106.51MB
=> sha256:5e3b1213efc56509e78bd002983945c164de2a37205e06a62dada023124dc743 6.20MB / 6.20MB
=> extracting sha256:0e29946e541c0dd09281d21a77ae01db7865c1095074f32e00e06774ee1e9
=> sha256:9fddfd56334f2e0efad7e241bf5e7459c40ed105c5478676f41c1244bd00752 14.21MB / 14.21MB
=> extracting sha256:90d29c73b5200209705c07a54f0bf3e921095a296c714053a32ae07d19231fcd
=> extracting sha256:c08297ae03722f0700ca3f35823ed21ba085d61d509c05a05ab53d740c0d56
=> sha256:408f02044bac0432ca52cbb9f254b1c91fca0800bfeef0be0b243b3f31bab7 235B / 235B
=> sha256:c4f42be2be52b0900ebffc040c1d0f13de530434ccc5f5d0954a50848a0160a2a3f 2.21MB / 2.21MB
=> extracting sha256:6494a4811622b31c027ccac322ca63937fd005f509a9306f15c01aade718793
=> extracting sha256:6f9f74800d7a93fe0172f594fab05e0b4e0a041a0fe09112efc7e4d3c70f7
=> extracting sha256:5e3b1213efc56509e78bd002983945c164de2a37205e06a62dada023124dc743
=> extracting sha256:9fddfd56334f2e0efad7e241bf5e7459c40ed105c5478676f41c1244bd00752
=> extracting sha256:408f02044bac0432ca52cbb9f254b1c91fca0800bfeef0be0b243b3f31bab7
=> extracting sha256:c4f42be2be52b0900ebffc040c1d0f13de530434ccc5f5d0954a50848a0160a2a3f
[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 sha256:1756719486df002fad5dae305c5221513f2f2d10a48e8d242b22a28af0379f19
=> 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- $\{random\}$
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. This section includes a 'Deploy configuration' area with several dropdown menus: '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). There is also an 'ADD JOB' button and a 'REMOVE' button for the current deployment.

```
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>
```

