

Assignment - 4

Docker and Kubernetes

Assignment Date	24 October 2022
Student Name	Punitha Vennila N
Student Roll Number	923819106042
Maximum Marks	2 Marks

Question-1:

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

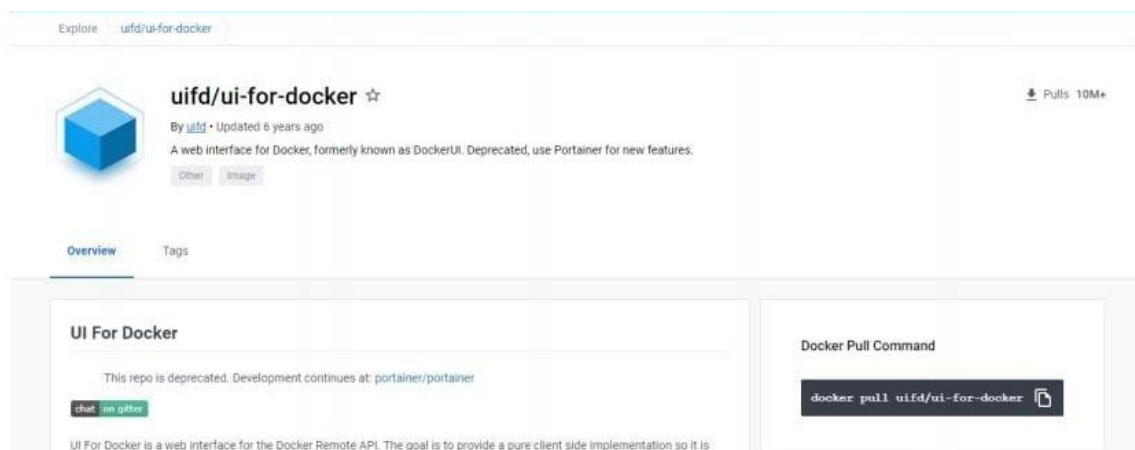
Solution:

```
docker run --rm -p 8787:8787 rocker/verse
docker pull rocker/verse
docker login --username=punithavennilan--email=punithavennila025@gmail.com
WARNING: login credentials saved in /home/punithavennilan/.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 punithavennilan
/verse_gapminder:firsttry
docker push
punithavennilan/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

uifd on docker

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. #
#####
root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
441194d980c8: 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
c590dd163101ae795bdcea0eb1dd98f6fe549cb5f24dcb9ff7c1931923fc0d
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

Containers created

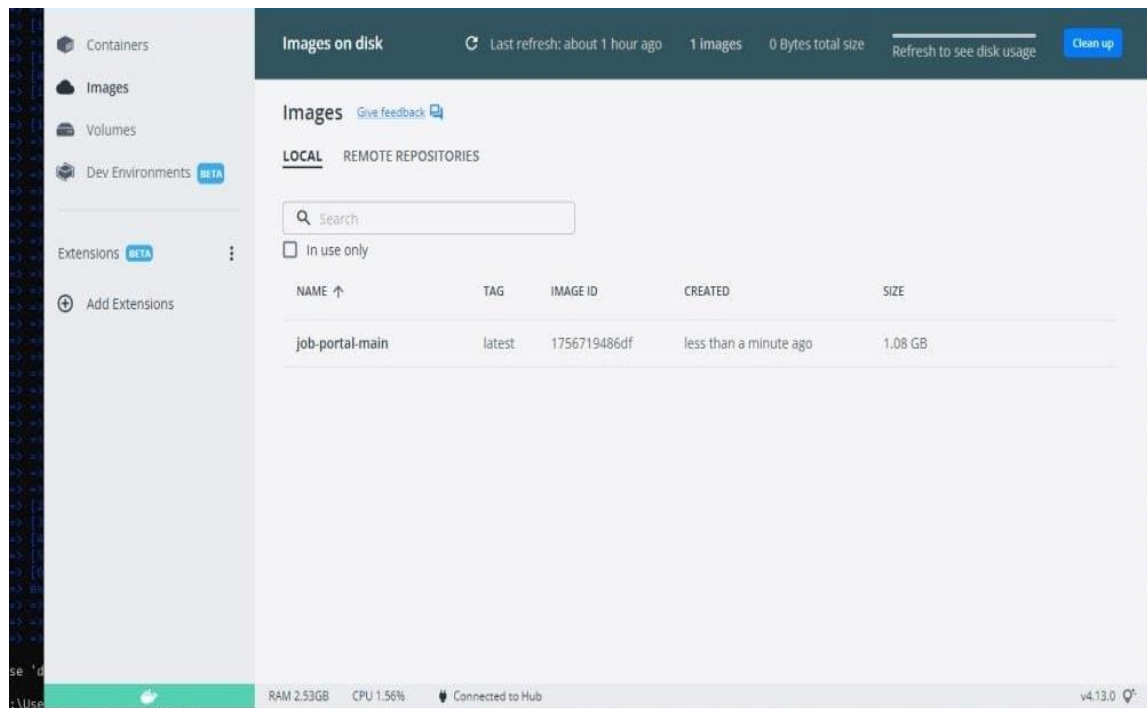
Images created

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: 607B
[1/6] FROM docker.io/library/python:3.6@sha256:f8052eaf88c25f6d22354d547d892591067aa4026a7fa9a810df0f300af6fc
=> resolve docker.io/library/python:3.6@sha256:f8052eaf88c25f6d22354d547d892591067aa4026a7fa9a810df0f300af6fc
=> sha256:f8052eaf88c25f6d22354d547d892591067aa4026a7fa9a810df0f300af6fc 1.86kB / 1.86kB
=> sha256:0097a496788c079df5ac31872359c20e519f82214c0448c928393b37623b0800 2.21kB / 2.21kB
=> sha256:5420863d07c53ed1a6e21fc889abbc8486a27d34c009208ff791f3f44b104 0.27kB / 0.27kB
=> sha256:0e29566541c0b309281d21a73a0d1b7865c1b5c74f32d009e0b77a6e1a3 54.92MB / 54.92MB
=> sha256:90828c73b52b92b97d5c07a54fb0f3e921995a204c714b51a32a67d19231fcd 5.15MB / 5.15MB
=> sha256:c5b7ae361722f070eca53f35823ed21baa85d61d5d95cd9a5ab53d748cd56 10.87MB / 10.87MB
=> sha256:6404ae011622b31c027ccac322ca463937fd005f569a930ef15c01aade718793 54.57MB / 54.57MB
=> sha256:6f9f74096dfa93fe0172f594fab85ebba0a041a0fefd9112efc7e4d3c78f7 196.51MB / 196.51MB
=> sha256:5e3b1213efc56590e78bd602983945c164de2a37285e06a62dada823134dc743 6.29MB / 6.29MB
=> extracting sha256:0e29566541c0b309281d21a73a0d1b7865c1b5c74f32d009e0b77a6e1a3 27
=> sha256:9fddfdcc334f2e0cfad7e241bf5e7459c40ed10c5478076f41c124bd09752 14.21MB / 14.21MB
=> extracting sha256:0097a496788c079df5ac31872359c20e519f82214c0448c928393b37623b0800 2
=> extracting sha256:c5b7ae361722f070eca53f35823ed21baa85d61d5d95cd9a5ab53d748cd56 4
=> sha256:404f8204dbac0432ca522cbb9f254b1c91fca000b7fee00e0b243b2f31bab7 235B / 235B
=> sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a50848a6169a3a3f 2.21MB / 2.21MB
=> extracting sha256:6404ae011622b31c027ccac322ca463937fd005f569a930ef15c01aade718793 27
=> sha256:6f9f74096dfa93fe0172f594fab85ebba0a041a0fefd9112efc7e4d3c78f7 131
=> extracting sha256:5e3b1213efc56590e78bd602983945c164de2a37285e06a62dada823134dc743 0
=> extracting sha256:9fddfdcc334f2e0cfad7e241bf5e7459c40ed10c5478076f41c124bd09752 11
=> sha256:404f8204dbac0432ca522cbb9f254b1c91fca000b7fee00e0b243b2f31bab7 1
=> extracting sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a50848a6169a3a3f 2
[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 sha256:1756719486df007fad5ae3d5c5221513f2ff2db49a0d242b22a28ef0379f19
=> 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' job. The job configuration includes: 'Deploy configuration' (Rolling Deploy), '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). A 'REMOVE' button is visible next to the job name.

```
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 punithavennilan-add  
<your_punithavennilan>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/clouddatabases-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>

