

Assignment - 4

Assignment Date	November 10
Student Name	Rahul E
Student Roll Number	422619106013
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=abuthahir --email=ssnehasri178@gmail.com WARNING: login credentials saved in  
/home/abuthahir/.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
0168d115f220	3 days ago		1.954 GB	docker tag bb38976d03cf abuthahir
/verse_gapminder:firsttry docker push abuthahir				
/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 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
```

cd9an2u3_cd9av060qau0008hbjs0

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 PWD team.
#####
[models] (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
441194d000c8: Pull complete
Digest: sha256:fe371ff3a69549269b24073a5ab1244dd4c0b834cbadf244070572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
[models] (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
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24decb9ff7c1931923fc0d
[models] (local) root@192.168.0.13 ~
```

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

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: 887B
[1/6] FROM docker.io/library/python:3.6@sha256:f8652a7f88c25f6d22354d57d892591867aa4e26a7faa6d1d9f9388af6fc
-> resolve docker.io/library/python:3.6@sha256:f8652a7f88c25f6d22354d57d892591867aa4e26a7faa6d1d9f9388af6fc
-> sha256:f8652a7f88c25f6d22354d57d892591867aa4e26a7faa6d1d9f9388af6fc 1.86GB / 1.86GB
-> sha256:8097a498788e879d75a31872159c2de510f82214c0448e932393b376d3b8ed 2.22kB / 2.22kB
-> sha256:54268a3887c5a3ed34c621fc88ebbc848a27634c802088ff71c7f44b18e 9.27kB / 9.27kB
-> sha256:8e29546b541cdd389281d21a73a9d1d678865c1b95b74f32b88e9eb77ade1e3 54.92MB / 54.92MB
-> sha256:98828c73b52492097d5c87a54f08f3e921995a296c714b53a32ae67d19231fcd 5.15MB / 5.15MB
-> sha256:cb5b7ae361722f978eca51f35823ed21baa85d83d5d95cda95ab53d749cd56 10.87MB / 10.87MB
-> sha256:6494e4811622b31c027cc322ca63937fd805f569a930ef15c01ade718793 54.57MB / 54.57MB
-> sha256:6f9f74896dfae93fe8172f594faa85e0bae8a481a07ef09112efc7e4d3c78f7 196.51MB / 196.51MB
-> sha256:5e3b1213efc56598e78bd0602983945c164de2a37285e0ba3dada823134dc743 6.29MB / 6.29MB
-> extracting sha256:8e29546b541cdd389281d21a73a9d1d678865c1b95b74f32b88e9eb77ade1e3
-> sha256:9fd0fd36334f2eeefad7e241bf5e7459c40ed105c5478676f41c1244bd96752 14.21MB / 14.21MB
-> extracting sha256:9fd0fd36334f2eeefad7e241bf5e7459c40ed105c5478676f41c1244bd96752
-> extracting sha256:cb5b7ae361722f978eca51f35823ed21baa85d83d5d95cda95ab53d749cd56
-> sha256:484f02044bac0432ca522cb9f254b1c91fcea080bfeef0be0b243b2f31bab7 235B / 235B
-> sha256:c4f42be2be53b900ebffc048c1df13de538434ccc5f5d054a56848a6169a3af 2.21MB / 2.21MB
-> extracting sha256:6494e4811622b31c027cc322ca63937fd805f569a930ef15c01ade718793
-> extracting sha256:6f9f74896dfae93fe8172f594faa85e0bae8a481a07ef09112efc7e4d3c78f7
-> extracting sha256:5e3b1213efc56598e78bd0602983945c164de2a37285e0ba3dada823134dc743
-> extracting sha256:9fd0fd36334f2eeefad7e241bf5e7459c40ed105c5478676f41c1244bd96752
-> extracting sha256:484f02044bac0432ca522cb9f254b1c91fcea080bfeef0be0b243b2f31bab7
-> extracting sha256:c4f42be2be53b900ebffc048c1df13de538434ccc5f5d054a56848a6169a3af
[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 flw_db
exporting to image
-> exporting layers
-> writing image sha256:1756719486df003fad5dee105c5221513f2f2d1b49a8d242b22a28af8379f19
-> naming to docker.io/library/job-portal-main

```

• 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

Containers

Images

Volumes

Dev Environments

Extensions

Add Extensions

Images on disk

Last refresh: about 1 hour ago 1 images 0 Bytes total size Refresh to see disk usage Clean up

Images Give feedback

LOCAL REMOTE REPOSITORIES

Search

In use only

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

RAM 2.53GB CPU 1.56% Connected to Hub v4.13.0

QUESTION-3:

1. 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 and an 'ADD JOB' button. The 'Rolling Deploy' section contains a 'Deploy configuration' table with the following fields:

Field	Value
Deployer type	Cloud Foundry
IBM Cloud region	US South - https://api.ng.bluemix.net
Organization	bluemix_devops@ibm.com
Space	demo
Application name	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 }
```

QUESTION-4:

1. 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 abuthahir-add  
<your_abuthahir>ibmcloudresource service-instance-create example-postgresql databases-  
forpostgresql 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>
```

Hello World!

Thanks for creating an [IBM Cloud Databases for PostgreSQL](#) database.

Add a word to the database

The word is defined as

Database output

```
The word bye is defined as a goodbye
The word bye is defined as a farewell
The word hello is defined as a greeting
The word hello is defined as a greeting
The word hello bob is defined as a greeting
The word hello bob is defined as a greeting
```