

**Assignment - 4**  
**Docker and Kubernetes**

Date	5 November 2022
Name	BalaPriya.R
Team Id	PNT2022TMID50246
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](#) [see gitlog](#)

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-with`

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:
e41194d080c8: Pull complete
Digest: sha256:fef718f0ae3949269b24073a5ab1244dd4e0b834cbadf244870572130b1cb749
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
c590dd163101ee795bdcea0eb1ddd98f6fe549cb5f24dec9ff7c1931923fc0d
[node1] (local) root@192.168.0.13 ~
```

Not secure | ip172-18-0-4-cd9an2u3tccg00fgf6k0-9000.direct.labs.play-with-docker.com/

## UI For Docker


The UI for Docker container engine

[Learn more.](#)

Running Containers

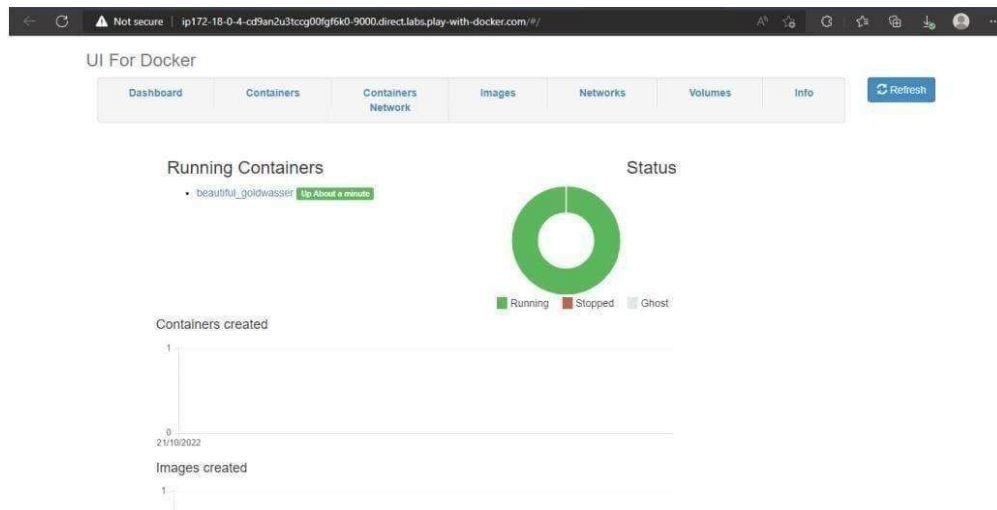
• [beautiful\\_goldwasser](#) [Tip About a minute](#)

Status



Dashboard Containers Containers Network Images Networks Volumes Info

[Refresh](#)

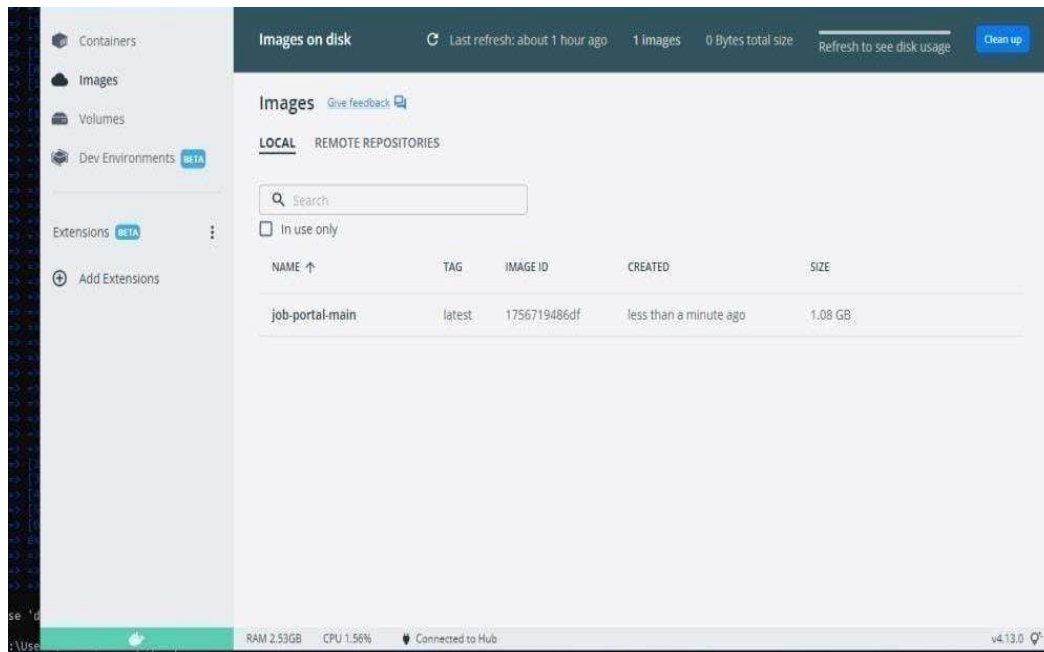


## Question-2:

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 dockerignore: 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:f8652a7ef88c15fd22154d547d892501967aad026a7faga0819df0f30daf6fc
=> resolve docker.io/library/python:3.6@sha256:f8652a7ef88c15fd22154d547d892501967aad026a7faga0819df0f30daf6fc
=> sha256:f8652a7ef88c15fd22154d547d892501967aad026a7faga0819df0f30daf6fc 1.86kB / 1.86kB
=> sha256:0007a4067e0c079d5fac31672359c2d6510f92214c0448e026393b376d3b000 2.22kB / 2.22kB
=> sha256:54266638007c5e3ad2ace21fc889abbcb486a2763ac0002000ff71f3f44b104 9.27kB / 9.27kB
=> sha256:0e29546d541cd309281d21a72a9d1db70665c1b95b74f32b009e0b77a6e1e3 54.92MB / 54.92MB
=> sha256:8b836c73b52b02b7d5c87a54f08f9e92199c296c714b53a22a607d19111fc8 5.15MB / 5.15MB
=> sha256:cb997ae0d7224078ec03f3833ed1ba05d015a95cd0b9a9d746cdd9e 10.81MB / 10.81MB
=> sha256:6404a401822b11c87ccac32c4803917f4d895f889a93ae35c01a0e718793 54.57MB / 54.57MB
=> sha256:6f9f7409d0fa03fa8172f504fab48e0b4a8a9401a8fef9112efc7e4d3c76f7 195.51MB / 195.51MB
=> sha256:5e3b1213efc50598e78b06030e3045c164de7a37205e66a3dada023124dc743 6.29MB / 6.29MB
=> extracting sha256:0e29546d541cd309281d21a72a9d1db70665c1b95b74f32b009e0b77a6e1e3
=> sha256:0fdd9dc96336f2c0efad7e241bf5e7439c40ed195c5470076f41c1245006752 14.21MB / 14.21MB
=> extracting sha256:0b29c73052b0209705e87e54f09f3ed118950296c714b53a22a607d19111fc8
=> sha256:c05b2e361224f070eca53f35623ed21bae85d63d5095c95a95ab3d7406d456 2 2
=> sha256:404f0204bac0432cab22cbb0f75401c91fca6006f0ee0b0b243b2f31ba07 235B / 235B
=> sha256:c4f42e2ba53b090ebffcc04cd1f13de338434ccc5f5d0954a5084ba6159a3af 2.21MB / 2.21MB
=> extracting sha256:6404a401822b11c87ccac32c4803917f4d895f889a93ae35c01a0e718793
=> extracting sha256:6f9f7409d0fa03fa8172f504fab48e0b4a8a9401a8fef9112efc7e4d3c76f7
=> extracting sha256:0b30121bf5059878a0802031045c164de2c37289e8a6d0dada023124dc743
=> extracting sha256:91d0f6c5634f2e6fa07e2d1bf547459c9ed105c5278676f41c1244000702
=> extracting sha256:404f0204bac0432cab22cbb0f75401c91fca6006f0ee0b0b243b2f31ba07
=> extracting sha256:c4f42e2ba53b090ebffcc04cd1f13de338434ccc5f5d0954a5084ba6159a3af
[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 job_db
=> exporting image
=> exporting layers
=> writing image sha256:1756719488d002fad5dae395c5221b13f2ff2d1049a8d242b22a28f0370f19
=> 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

DEPLOY
DELETE

INPUT
JOBS
ENVIRONMENT PROPERTIES

Rolling De...
ADD JOB

Rolling Deploy
REMOVE

Deploy configuration

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 }

```

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>/icdpdg" # 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>/icdpdg .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-deploy
ment.yml
deployment.ap
ps/icdpostgres-
app      created
service/cloudpo
stgres-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
```