

Assignment - 4 Docker and Kubernetes

Assignment Date	18th November 2022
Student Name	Rohith Jones S
Project ID	PNT2022TMID04044
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=rohithselvaraj -  
-email=rohithjones@gmail.com WARNING: login credentials saved in  
/home/rohithselvaraj/.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
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 ☆

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](#)

OverviewTags

UI For Docker

This repo is deprecated. Development continues at: [portainer/portainer](#)

[chat](#) [ask question](#)

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

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.

[rookel] (local) root@192.168.0.13 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
641194d880c8: Pull complete
Digest: sha256:fe371ff3a69549269b24073a5ab1244dd4c0b834cbwdf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[rookel] (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
c590dd16310ae795bdce0eb1dd498f6fe549cb5f24dcb9ff7c1931923fc0d
[rookel] (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: 687B
[1/6] FROM docker.io/library/python:3.6@sha256:f8b52afaf8bc25f0d22354d547d892591067aa4026a7fa9a0819df9f300af6fc
--> resolve docker.io/library/python:3.6@sha256:f8b52afaf8bc25f0d22354d547d892591067aa4026a7fa9a0819df9f300af6fc
--> sha256:f8b52afaf8bc25f0d22354d547d892591067aa4026a7fa9a0819df9f300af6fc 1.80GB / 1.80GB
--> sha256:d997a68070bc879d25ac31872159c2be51078214c6449028391b376d3b8d 2.21GB / 2.21GB
--> sha256:c42603808fca31db446e21fc809abc446a27624e09200ff7f1c3f6db184 9.27GB / 9.27GB
--> sha256:8e39c46041c0b109281021a73a081db78065-1b95174f32b0090b77ade1e3 54.92MB / 54.92MB
--> sha256:8882c7352402097d5c87a44b0f3e921095a306c714b53a32ae7d19231fcd 5.10MB / 5.10MB
--> sha256:cb5b7ae361727497eca33f35873ed21ba05d61d0b95cd5a95ab53d740cd56 19.87MB / 19.87MB
--> sha256:6494e4811622031c027cc322ca463937f089f508a9200f15c01ade718793 54.57MB / 54.57MB
--> sha256:6f9f74890df9a31e81727504faba85e0b4e0a0481a0fef09112efc7e4d3c78f2 196.51MB / 196.51MB
--> sha256:5e3b1213efc56590e78b0602003945c164de2a37205e00a02dada023124d743 6.20MB / 6.20MB
--> extracting sha256:0e295460541c0b109281021a73a081db78065-1b95174f32b0090b77ade1e3
--> sha256:9f0dfdc56334f2e0efad7e241bf5e7450c40ed105c5478070f41c1244bd06752 14.21MB / 14.21MB
--> extracting sha256:0e295460541c0b109281021a73a081db78065-1b95174f32b0090b77ade1e3
--> extracting sha256:cb5b7ae361727497eca33f35873ed21ba05d61d0b95cd5a95ab53d740cd56
--> sha256:404f02044bac0432ca522cb0f7354b1c91fcea0800bfee0b0b243b2f31bab7 235B / 235B
--> sha256:c4f42be2be53b900ebffcc040cd1f13de530434ccc5f3d054a56848a0160a3a3f 2.21MB / 2.21MB
--> extracting sha256:6494e4811622031c027cc322ca463937f089f508a9200f15c01ade718793
--> extracting sha256:6f9f74890df9a31e81727504faba85e0b4e0a0481a0fef09112efc7e4d3c78f2
--> extracting sha256:5e3b1213efc56590e78b0602003945c164de2a37205e00a02dada023124d743
--> extracting sha256:9f0dfdc56334f2e0efad7e241bf5e7450c40ed105c5478070f41c1244bd06752
--> extracting sha256:404f02044bac0432ca522cb0f7354b1c91fcea0800bfee0b0b243b2f31bab7
--> extracting sha256:c4f42be2be53b900ebffcc040cd1f13de530434ccc5f3d054a56848a0160a3a3f
[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 sha256:1756719486df087fad5dae305c5221513f2ff2db49a0d242b22a28af0379f10
--> naming to docker.io/library/job-portal-main

```

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

Containers

Images

Volumes

Dev Environments BETA

Extensions BETA

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 'DEPLOY' section of the IBM Cloud console. It has tabs for 'INPUT', 'JOBS', and 'ENVIRONMENT PROPERTIES'. Below the tabs are icons for 'Rolling De...' and 'ADD JOB'. The 'Rolling Deploy' section is expanded, showing a 'Deploy configuration' with a 'REMOVE' button. The configuration includes:

- 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
```