Assignment - 4 Docker and Kubernetes

Assignment Date	November 3
Student Name	Pathuri supriya
TEAM ID	PNT2022TMID37166
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 rocker/verse   latest
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

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: 6B7B
[1/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6810df9f300af6fc
 => resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6810df9f300af6fc
 => sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6810df9f300af6fc 1.86kB / 1.86kB
 => sha256:d097a4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d3b60d 2.22kB / 2.22kB
 => sha256:54280630d07c5e3ad24c6e21fc889abbc8486a27634c089208eff71f3f44b1b4 9.27kB / 9.27kB
 => sha256:0e29546d541cdbd309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3 54.92MB / 54.92MB
 => sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd 5.15MB / 5.15MB
 => sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56 10.87MB / 10.87MB
 => sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793 54.57MB / 54.57MB
 => sha256:6f9f74896dfa93fe0172f594faba85e0b4e0a0481a0fefd9112efc7e4d3c78f7 196.51MB / 196.51MB
 => sha256:5e3b1213efc56598e78bd602983045c164da2a37205e06a62dada023124dc743 6.29MB / 6.29MB
 => extracting sha256:0e29546d541cdbd309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3
 => sha256:9fdd0fdc56334f2e0efad7e241bf5e7459c40ed105c5478676f41c1244bd06752 14.21MB / 14.21MB
 => extracting sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd
 => extracting sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56
 => sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7 235B / 235B
 => sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a56848a6180a3a3f 2.21MB / 2.21MB
 => extracting sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793
 => extracting sha256:6f9f74896dfa93fe0172f594faba85e0b4e0a0481a0fefd9112efc7e4d3c78f7
 => extracting sha256:5e3b1213efc56598e78bd602983045c164da2a37205e06a62dada023124dc743
 => extracting sha256:9fdd0fdc56334f2e0efad7e241bf5e7459c40ed105c5478676f41c1244bd06752
 => extracting sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7
 => extracting sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a56848a6180a3a3f
[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 sha256:1756719486df002fad5dae305c5221513f2ff2d1b49a8d242b22a28af0379f19
 => 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
```



```json
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/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>

Hello World!

Thanks for creating an IBM Cloud Databases for PostgreSQL database.

Add a word to the database

The word [hello] is defined as [a greeting] [Add]

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
```