Assignment - 4
Docker and Kubernetes

Assignment Date	November 3
Student Name	Eshwaran
Student Roll Number	2116190701046
Maximum Marks	2 Marks

Question-1:

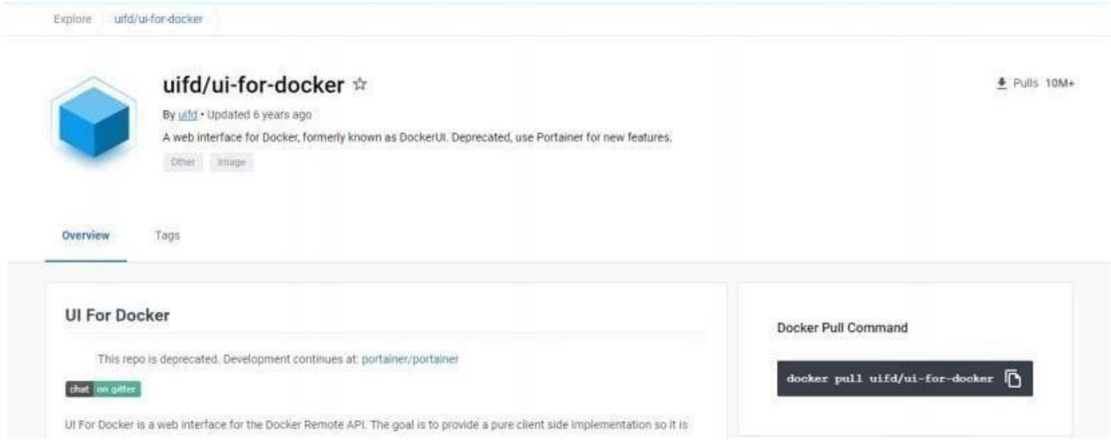
1. Pull an Image from docker hub and run it in docker playground.

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
```



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.

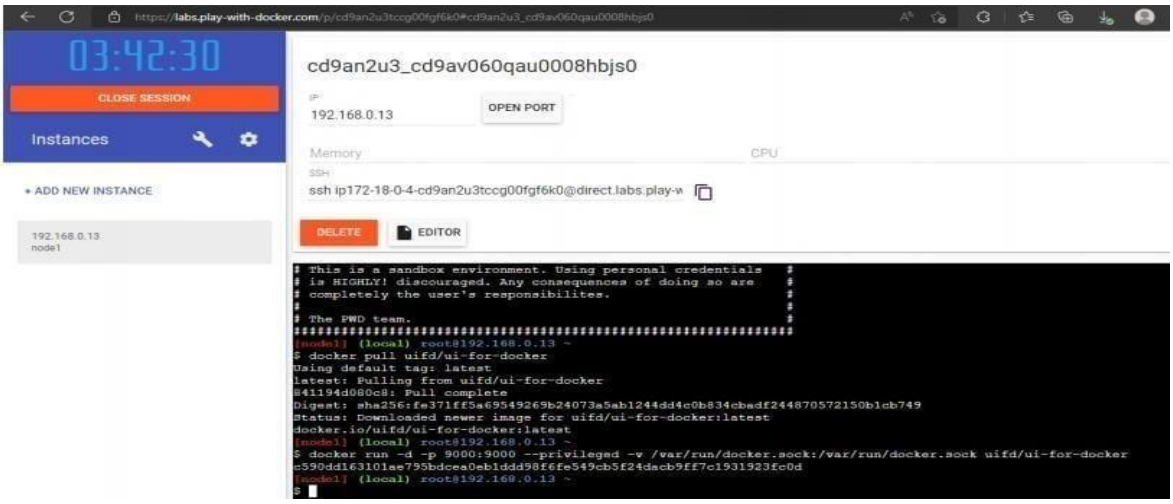
UI For Docker

This repo is deprecated. Development continues at: portainer/portainer

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
```

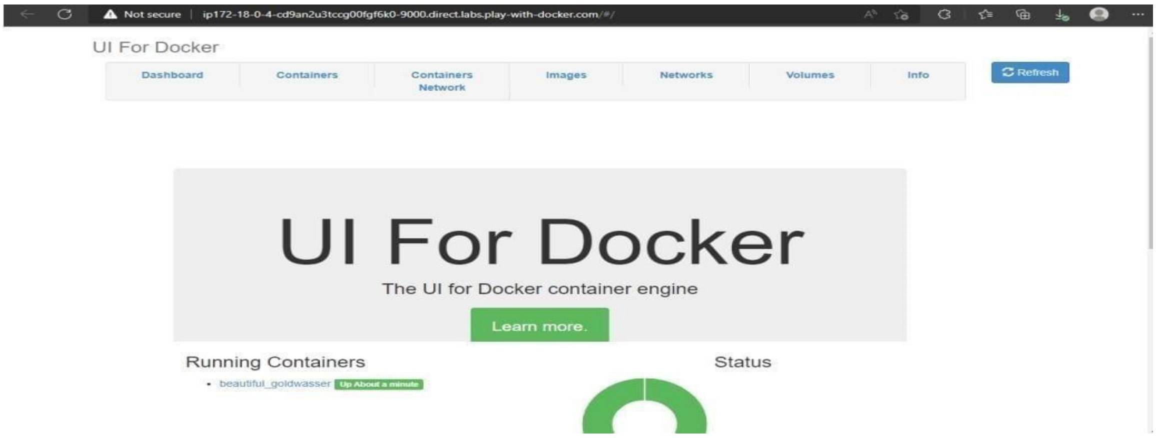


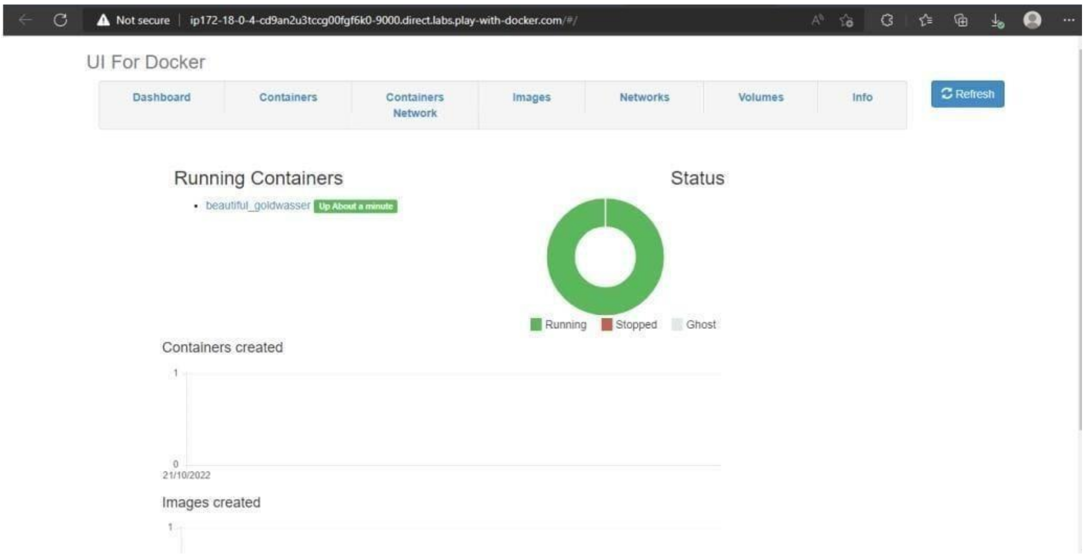
cd9an2u3_cd9av060qau0008hbjs0

IP: 192.168.0.13

SSH: ssh ip172-18-0-4-cd9an2u3tccg00fgf6k0@direct.labs.play-w

```
# This is a sandbox environment. Using personal credentials   #
# is HIGHLY! discouraged. Any consequences of doing so are    #
# completely the user's responsibilites.                      #
#                                                             #
# 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
841194d080c8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749
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
c590dd163101ae795bdcea0eb1ddd98f6fe549cb5f24dacb9ff7c1931923fc0d
[node1] (local) root@192.168.0.13 ~
$
```

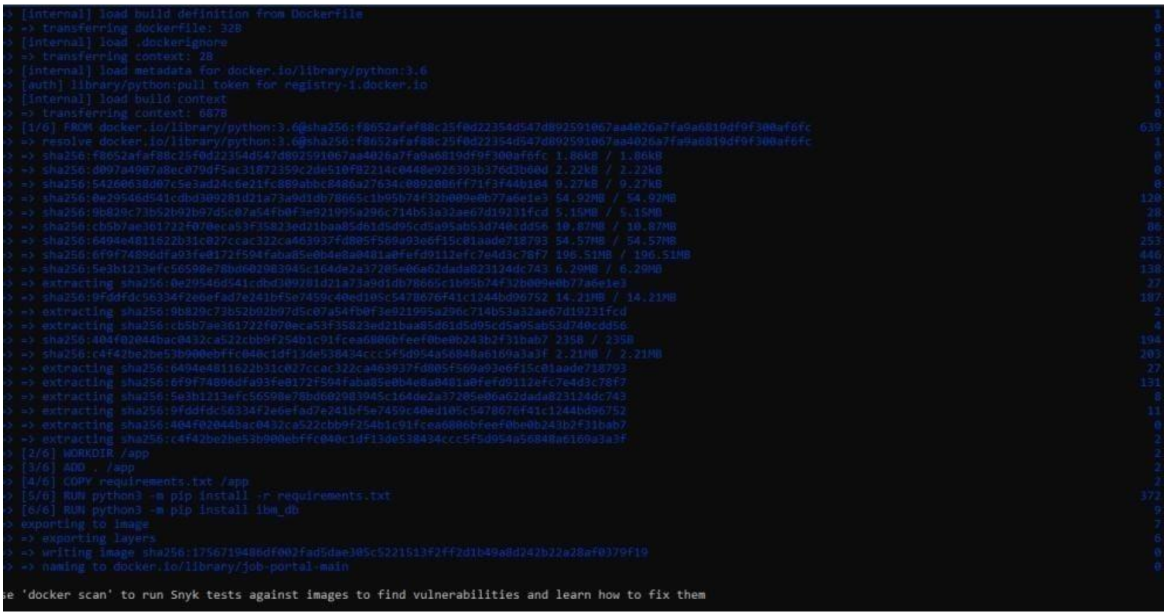




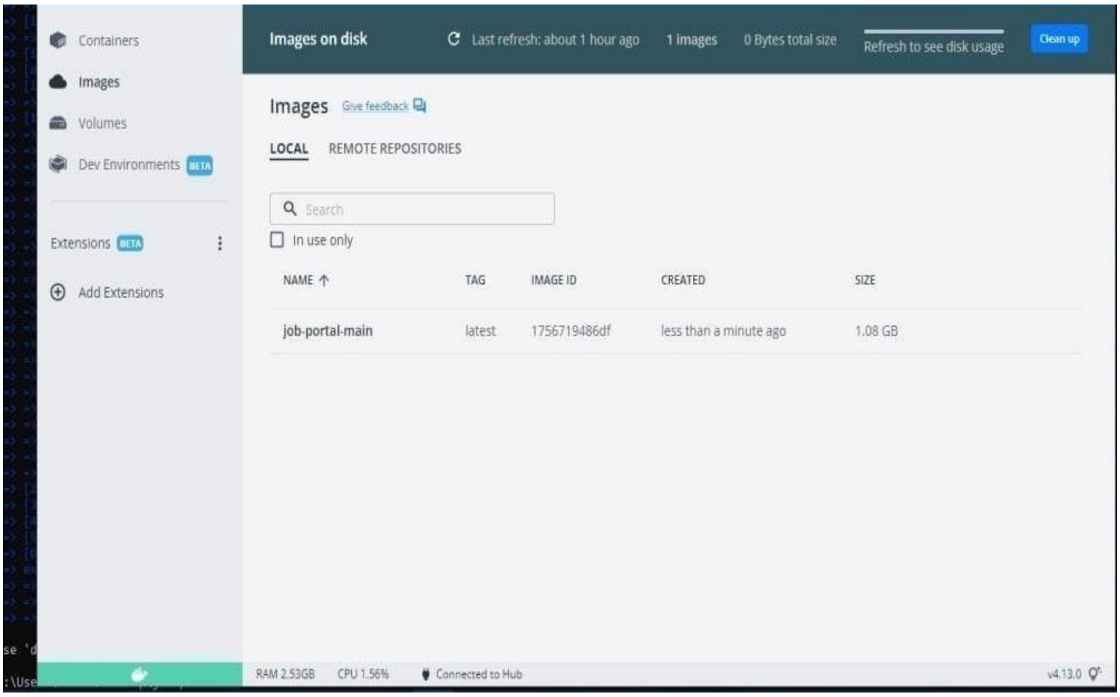
Question-2:

2. Create a docker file for the jobportal application and deploy it in Docker desktop application.

SOLUTION:



```
=> [internal] load build definition from Dockerfile                                                                   1
=> => transferring dockerfile: 32B                                                                                    0
=> [internal] load .dockerignore                                                                                      1
=> => transferring context: 2B                                                                                        0
=> [internal] load metadata for docker.io/library/python:3.6                                                          9
=> [auth] library/python:pull token for registry-1.docker.io                                                          0
=> [internal] load build context                                                                                      1
=> => transferring context: 687B                                                                                      0
=> [1/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc  639
=> => resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc    1
=> => sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc 1.86kB / 1.86kB                          0
=> => sha256:d097a4907a8ec079df5ac31872359c2de510f82214c0448e926393b376d3b60d 2.22kB / 2.22kB                          0
=> => sha256:54260638d07c5e3ad24c6e21fc889abbc8486a27634c0892086ff71f3f44b104 9.27kB / 9.27kB                          8
=> => sha256:0e29546d541cdbd309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3 54.92MB / 54.92MB                      120
=> => sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd 5.15MB / 5.15MB                         28
=> => sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56 10.87MB / 10.87MB                       80
=> => sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793 54.57MB / 54.57MB                      253
=> => sha256:6f9f74896dfa93fe0172f594faba85e0b4e8a0481a0fefd9112efc7e4d3c78f7 196.51MB / 196.51MB                    446
=> => sha256:5e3b1213efc56598e78bd602983945c164de2a37285e06a62dada823124dc743 6.29MB / 6.29MB                        138
=> => extracting sha256:0e29546d541cdbd309281d21a73a9d1db78665c1b95b74f32b009e0b77a6e1e3                               27
=> => sha256:9fddfdc56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752 14.21MB / 14.21MB                      187
=> => extracting sha256:9b829c73b52b92b97d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd                                2
=> => extracting sha256:cb5b7ae361722f070eca53f35823ed21baa85d61d5d95cd5a95ab53d740cdd56                                4
=> => sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7 235B / 235B                            194
=> => sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a56848a6169a3a3f 2.21MB / 2.21MB                        203
=> => extracting sha256:6494e4811622b31c027ccac322ca463937fd805f569a93e6f15c01aade718793                               27
=> => extracting sha256:6f9f74896dfa93fe0172f594faba85e0b4e8a0481a0fefd9112efc7e4d3c78f7                              131
=> => extracting sha256:5e3b1213efc56598e78bd602983945c164de2a37285e06a62dada823124dc743                                8
=> => extracting sha256:9fddfdc56334f2e6efad7e241bf5e7459c40ed105c5478676f41c1244bd96752                               11
=> => extracting sha256:404f02044bac0432ca522cbb9f254b1c91fcea6806bfeef0be0b243b2f31bab7                                0
=> => extracting sha256:c4f42be2be53b900ebffc040c1df13de538434ccc5f5d954a56848a6169a3a3f                                2
=> [2/6] WORKDIR /app                                                                                                 2
=> [3/6] ADD . /app                                                                                                   2
=> [4/6] COPY requirements.txt /app                                                                                   2
=> [5/6] RUN python3 -m pip install -r requirements.txt                                                             372
=> [6/6] RUN python3 -m pip install ibm_db                                                                            9
=> exporting to image                                                                                                 7
=> => exporting layers                                                                                                6
=> => writing image sha256:1756719486df002fad5dae305c5221513f2ff2d1b49a8d242b22a28af0379f19                           0
=> => naming to docker.io/library/job-portal-main                                                                     0

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



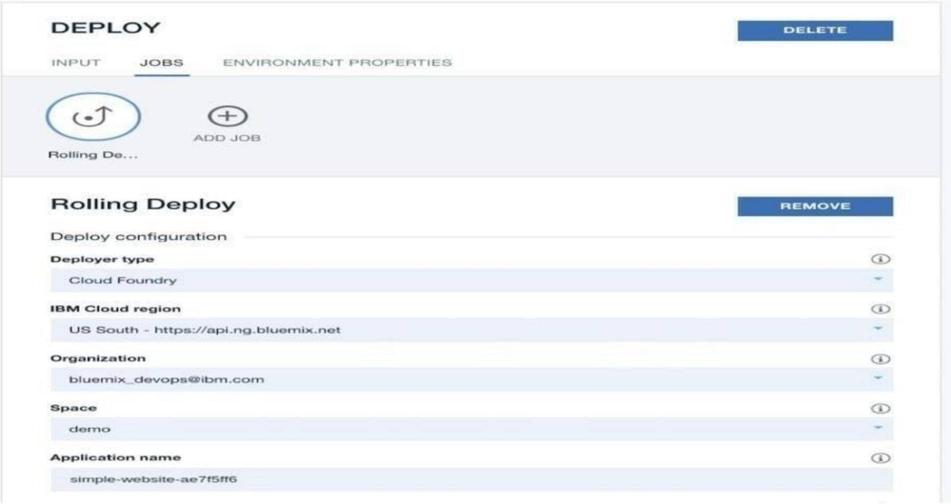
NAME ↑	TAG	IMAGE ID	CREATED	SIZE
job-portal-main	latest	1756719486df	less than a minute ago	1.08 GB

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
```



Rolling Deploy

Deploy configuration

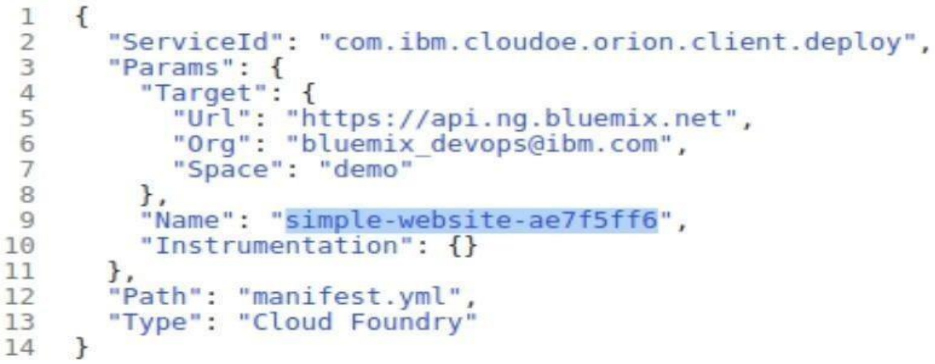
Deployer type: Cloud Foundry

IBM Cloud region: US South - https://api.ng.bluemix.net

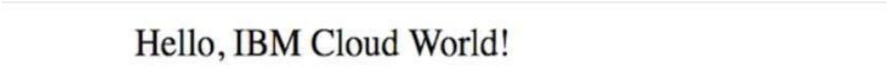
Organization: bluemix_devops@ibm.com

Space: demo

Application name: simple-website-ae7f5ff6



```
{
  "ServiceId": "com.ibm.cloudoe.orion.client.deploy",
  "Params": {
    "Target": {
      "Url": "https://api.ng.bluemix.net",
      "Org": "bluemix_devops@ibm.com",
      "Space": "demo"
    },
    "Name": "simple-website-ae7f5ff6",
    "Instrumentation": {}
  },
  "Path": "manifest.yml",
  "Type": "Cloud Foundry"
}
```



Hello, IBM Cloud World!

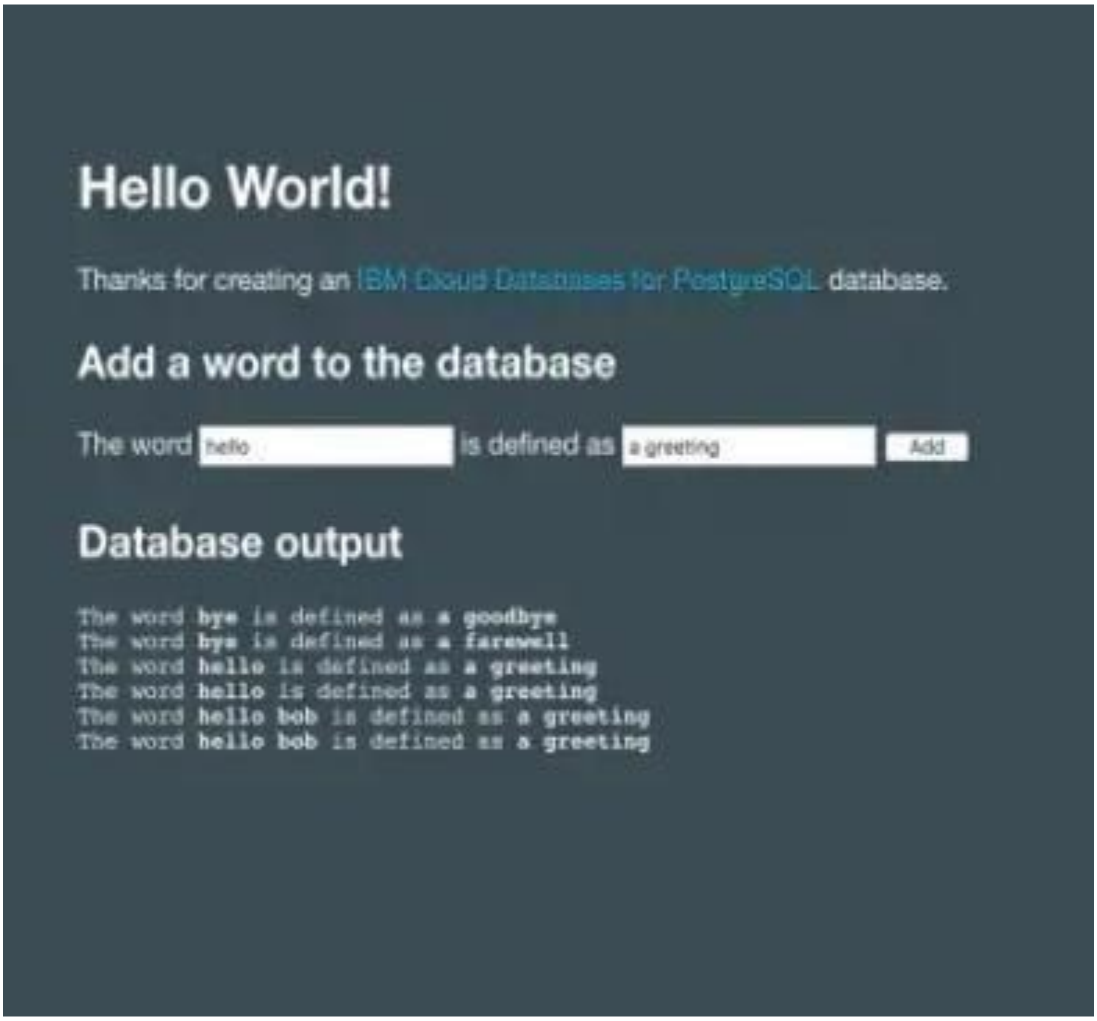
QUESTION-4:

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/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

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
```