ASSIGNMENT - 4

ROLL NUMBER	2019503020
NAME	KARTHIKEYAN C
TEAM ID	PNT2022TMID35667

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

```
$ docker --version
Docker version 20.10.17, build 100c701
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
[node1] (local) root@192.168.0.18 ~
$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fccceeec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    feb5d9fea6a5   13 months ago   13.3kB
$ docker run -it hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

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

FROM ubuntu

RUN apt-get update

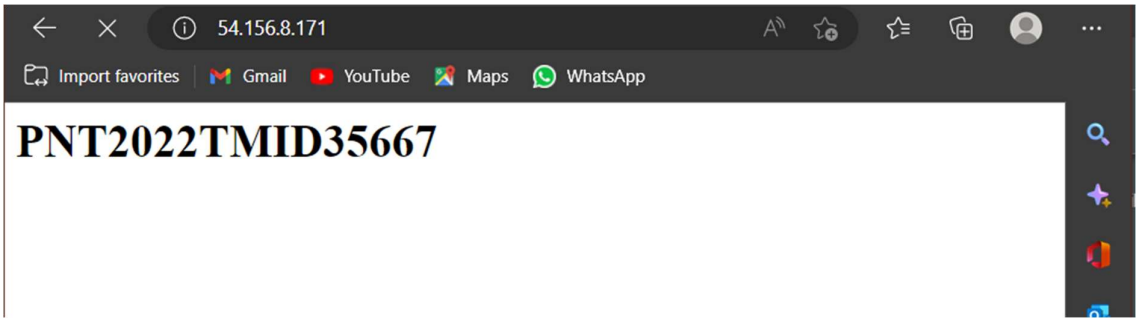
RUN apt-get apache2 -y

ADD ./index.html /var/www/html

CMD apachectl -D FOREGROUND

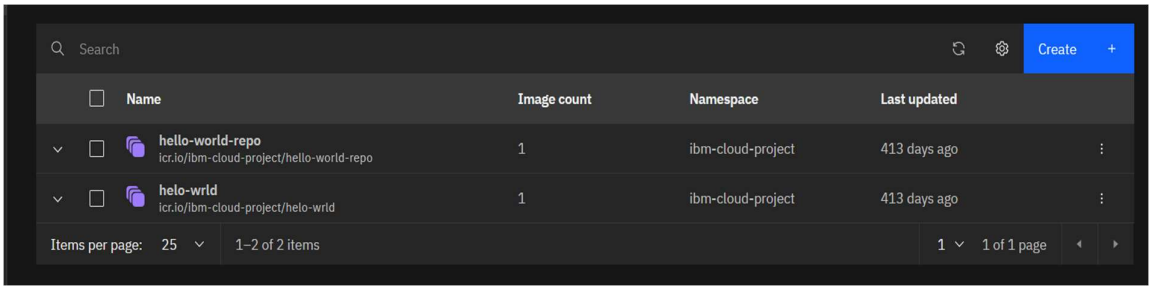
```
ubuntu@ip-172-31-28-246:~$ docker build . -t apache2
Sending build context to Docker daemon  15.87kB
Step 1/5 : FROM  ubuntu
 ---> a8780b506fa4
Step 2/5 : RUN  apt-get update
 ---> Using cache
 ---> 981b376d63ad
Step 3/5 : RUN  apt install apache2 -y
 ---> Using cache
 ---> e6dc16c6e4bc
Step 4/5 : ADD  ./index.html   /var/www/html
 ---> 7c2be22cde03
Step 5/5 : CMD apachectl  -D  FOREGROUND
 ---> Running in ad83f7238a24
Removing intermediate container ad83f7238a24
 ---> f874c46d2056
Successfully built f874c46d2056
Successfully tagged apache2:latest
```

```
ubuntu@ip-172-31-28-246:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                   NAMES
4c5680e607f2   apache2   "/bin/sh -c 'apachec…"   2 minutes ago   Up 2 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp   thirsty_dubinsky
ubuntu@ip-172-31-28-246:~$
```

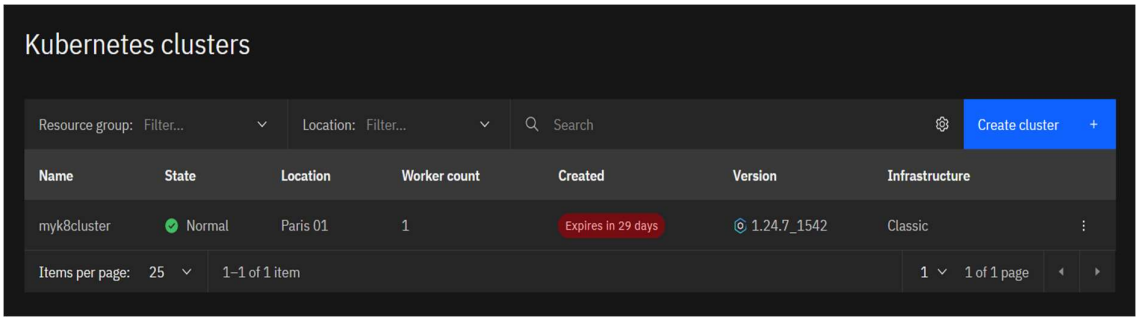


3. Create a IBM container registry and deploy helloworld app or jobportalapp

```
┌──(siva㉿kali)-[~]
└─$ docker tag hello-world icr.io/ibm-cloud-project/helo-wrld:v1

┌──(siva㉿kali)-[~]
└─$ docker images
REPOSITORY                                   TAG       IMAGE ID       CREATED         SIZE
hello-world                                  latest    feb5d9fea6a5   13 months ago   13.3kB
icr.io/ibm-cloud-project/hello-world-repo    latest    feb5d9fea6a5   13 months ago   13.3kB
icr.io/ibm-cloud-project/helo-wrld           v1        feb5d9fea6a5   13 months ago   13.3kB

┌──(siva㉿kali)-[~]
└─$ ibmcloud cr login
Logging 'docker' in to 'icr.io' ...
Logged in to 'icr.io'.

OK

┌──(siva㉿kali)-[~]
└─$ docker push icr.io/ibm-cloud-project/helo-wrld:v1
The push refers to repository [icr.io/ibm-cloud-project/helo-wrld]
e07ee1baac5f: Mounted from ibm-cloud-project/hello-world-repo
v1: digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525
```



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.




```
ubuntu@ip-172-31-28-246:~$ kubectl config current-context
myk8cluster/cdls28cf0rjkfc1fiuag
ubuntu@ip-172-31-28-246:~$
```

```
ubuntu@ip-172-31-28-246:~/assignment4/jobportal$ kubectl create -f deployment.yaml
deployment.apps/flask-node-deployment created
ubuntu@ip-172-31-28-246:~/assignment4/jobportal$ kubectl create -f service.yaml
service/flask-node-deployment created
```

```
ubuntu@ip-172-31-28-246:~/assignment4/jobportal$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
flask-node-deployment-668f76c67-zwzv5   1/1     Running   0          14m
ubuntu@ip-172-31-28-246:~/assignment4/jobportal$
```

```
ubuntu@ip-172-31-28-246:~/assignment4/jobportal$ kubectl get service
NAME                   TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
flask-node-deployment  ClusterIP   172.21.160.114   <none>        5000/TCP   21m
kubernetes             ClusterIP   172.21.0.1       <none>        443/TCP    26h
ubuntu@ip-172-31-28-246:~/assignment4/jobportal$
```