ASSIGNMENT 4

Assignment Date	17 October 2022
Student name	B Harsha Vardhan
Student roll no	211419104096
Maximum Marks	2 Marks

Questions:

1. Pull an image from docker hub and run it in docker playground.
2. Create a docker file for the jobportal application and deploy it in Docker desktop application.
3. Create a IBM container registry and deploy helloworld app or jobportal app.
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

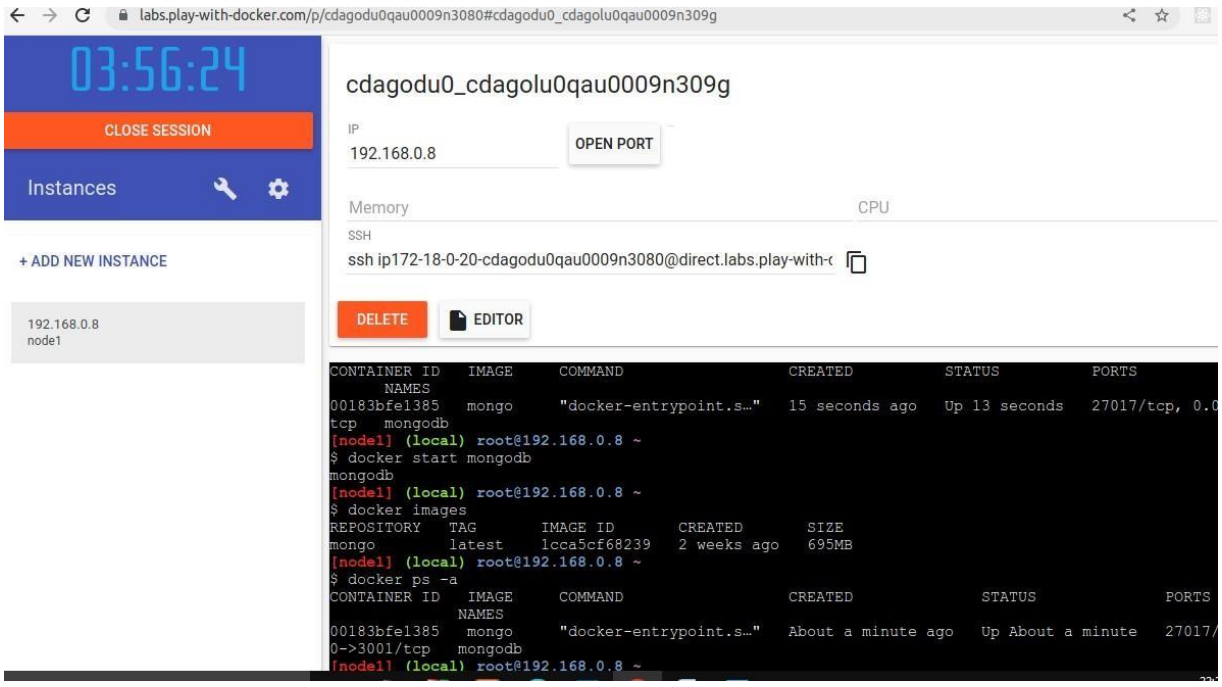
Output:

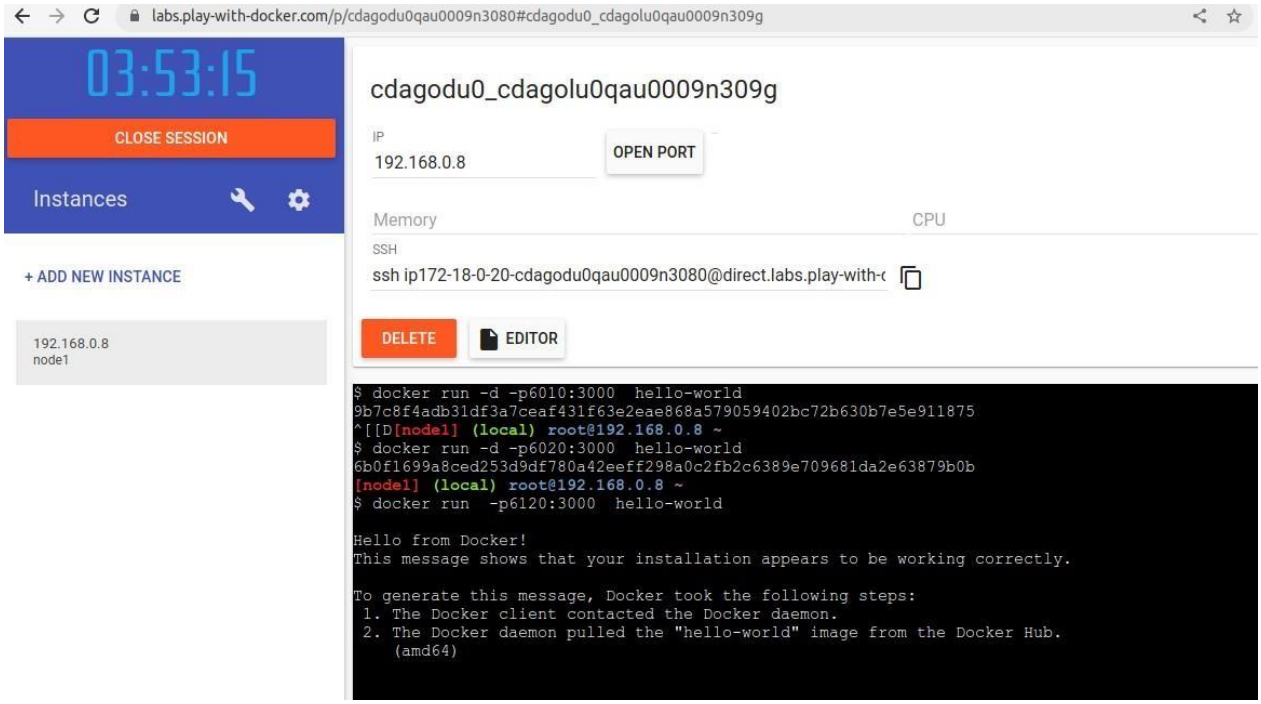
Pull image in docker:

Run Docker:

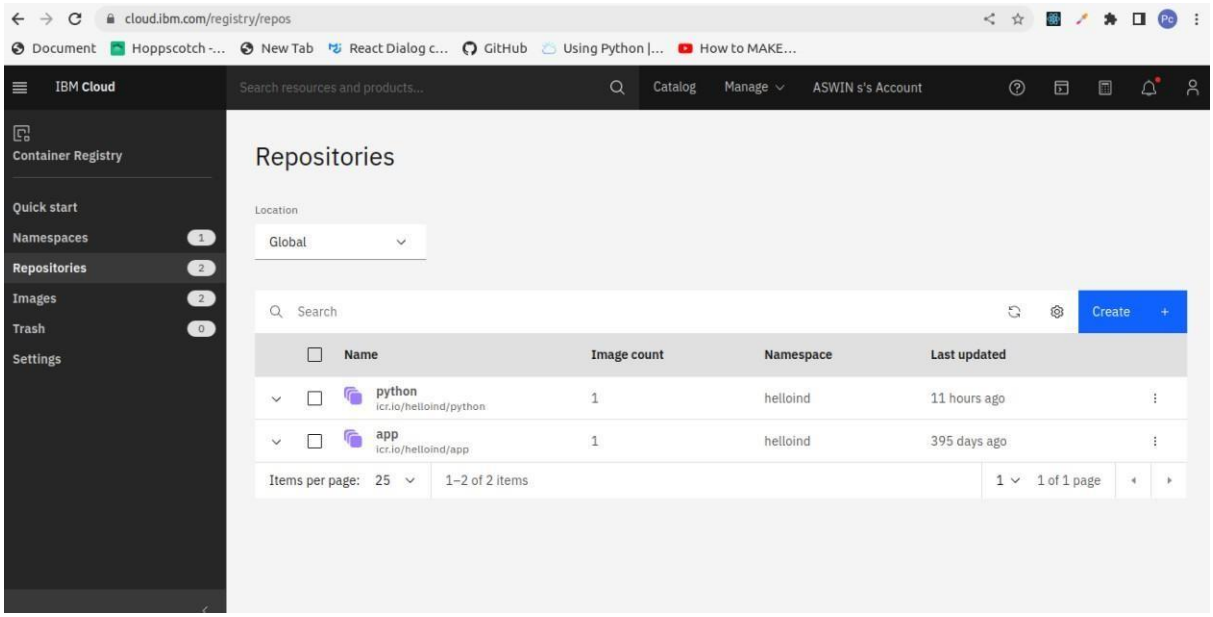
```
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/ac
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run -p5000:5000 flakapp:1.0
Unable to find image 'flakapp:1.0' locally
docker: Error response from daemon: pull access denied for flakapp, repository does not exist or may require 'docker lo
quested access to the resource is denied.
See 'docker run --help'.
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run -p5000:5000 -t  flakapp:1.0
Unable to find image 'flakapp:1.0' locally
docker: Error response from daemon: pull access denied for flakapp, repository does not exist or may require 'docker lo
quested access to the resource is denied.
See 'docker run --help'.
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run -p5000:5000 -t  flakapp
Unable to find image 'flakapp:latest' locally
docker: Error response from daemon: pull access denied for flakapp, repository does not exist or may require 'docker lo
quested access to the resource is denied.
See 'docker run --help'.
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run -p5000:5000 -t  flaskapp:1.0
docker: Error response from daemon: driver failed programming external connectivity on endpoint silly_bardeen (717385e0
d59728f1e3577afe801b18edb0257c07b377889d): Bind for 0.0.0.0:5000 failed: port is already allocated.
ERRO[0002] error waiting for container: context canceled
root@admin:/home/ash/Desktop/Flask ClassWorrk# docker run  -t  flaskapp:1.0
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.3:5000
Press CTRL+C to quit
172.17.0.1 - - [23/Oct/2022 17:21:29] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [23/Oct/2022 17:21:29] "GET /favicon.ico HTTP/1.1" 404 -
172.17.0.1 - - [23/Oct/2022 17:21:50] "GET / HTTP/1.1" 200 -
```

Docker Playground:





Container Registry:



KUBERNETES:

dashboard-adminuser.yaml

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
---
apiVersion: v1
kind: Secret
metadata:
  name: admin-user-token
  namespace: kubernetes-dashboard
  annotations:
    kubernetes.io/service-account.name: admin-user
type: kubernetes.io/service-account-token

---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
```

```
  namespace: kubernetes-dashboard
```

Flask-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: flask-app
spec:
 replicas: 3
 selector:
  matchLabels:
   app: flask-app
 template:
  metadata:
   labels:
    app: flask-app
  spec:
   containers:
    - name: flask-app-container
      image: flask-app-testing
      imagePullPolicy: Never
      ports:
       - containerPort: 5000
         protocol: TCP
```

Flask_ ingress.yaml:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
 name: flask-app-ingress
 annotations:
 kubernetes.io/ingress.class: nginx
  nginx.ingress.kubernetes.io/ssl-redirect: "false"

spec:
 # ingressClassName: nginx
 rules:
  - http:
      paths:
-      backend:
          service:
            name: flask-app-service
            port:
```

```
            number: 5000
        path: /
        pathType: Prefix
```

Flask_service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: flask-app-service
spec:
  type: ClusterIP
  ports:
    - port: 5000
  selector:
    app: flask-app
```