ASSIGNMENT 4

|  |  |
| --- | --- |
| Student Name | JAINUL MUFAFIKA H |
| Student Roll Number | 920419104034 |
| Maximum Marks | 2 Marks |
| Project Name | Containment Zone Alerting Application |

# Questions:

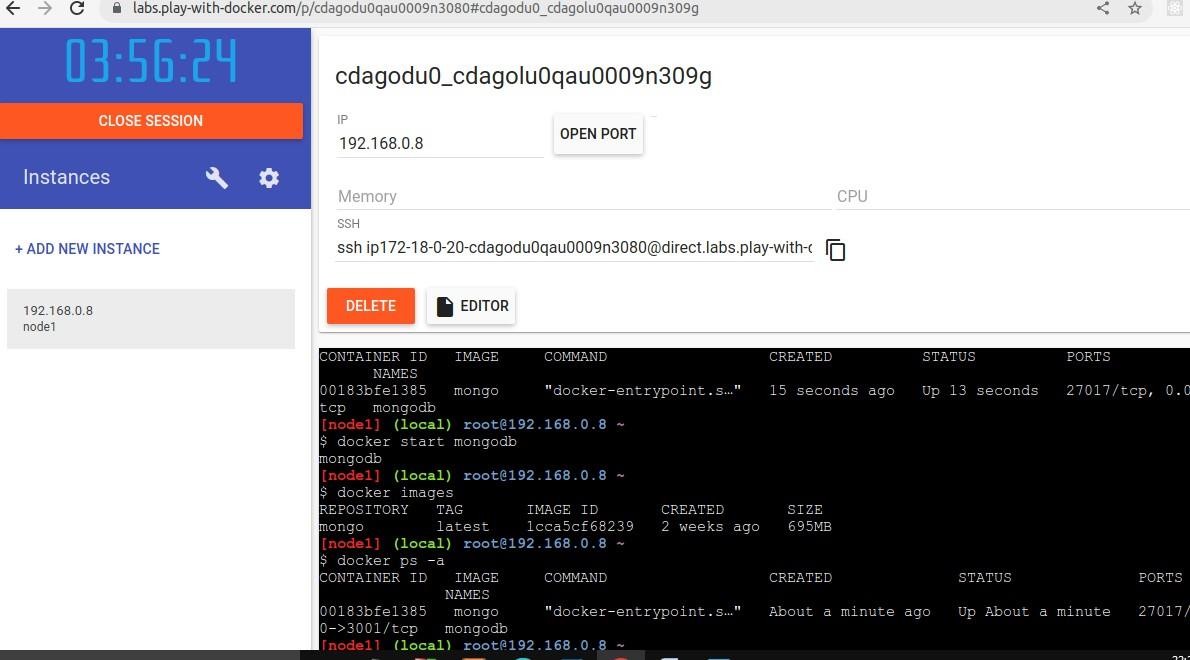
1. Pull an image from docker hub and run it in docker playground.
2. Create a docker file for the jobportal application and deploy it in Docker desktop application.
3. Create a docker file for the jobportal application and deploy it in Docker desktop application.
4. Create a IBM container registry and deploy helloworld app or jobportal app.
5. Create a Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the same app to run in nodeport

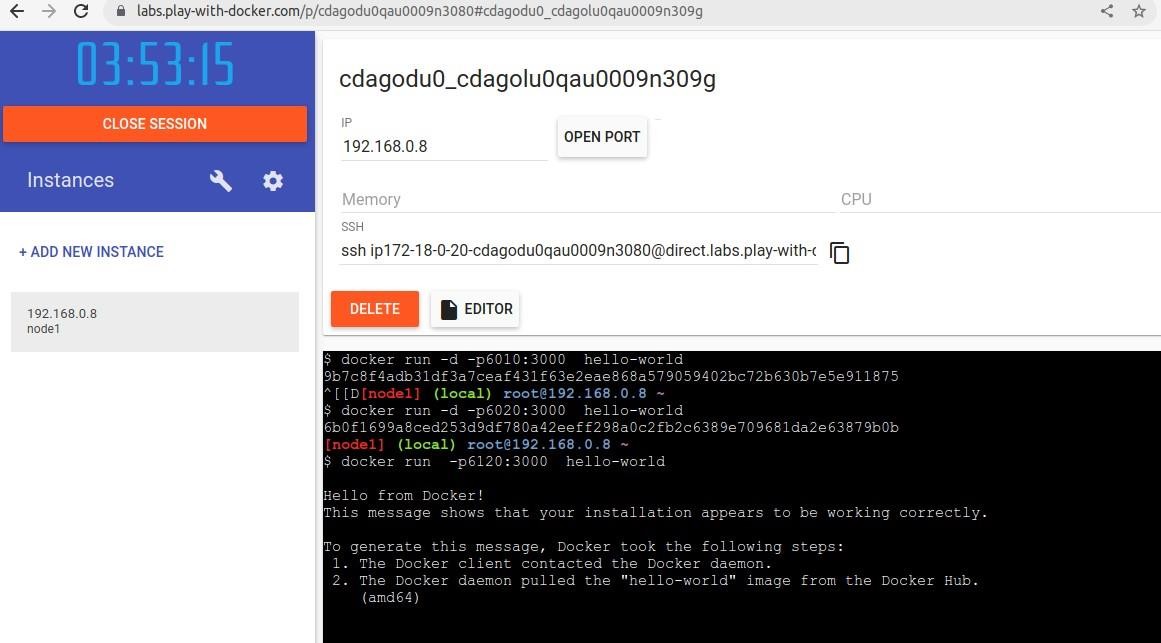
# Output:

**Pull image in docker:**

# C:\Users\ELCOT\Documents\OneDrive\Links\Pictures\Pictures\Screenshots\Screenshot (61).pngRun Docker:

**Docker Playground**:





# C:\Users\ELCOT\Documents\OneDrive\Links\Pictures\Pictures\Screenshots\Screenshot (51).pngContainer 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