ASSIGNMENT – 4

KUBERNETES, DOCKER

Assignment Date	28 October 2022
Student Name	Tharmarajan S
Student Roll Number	921319104210`
Maximum Marks	2 Mark

Question-1

Pull an image from Dockers hub and run it in Dockers playground.

SOLUTION:

STEP: 1

Login to Dockers hub and get an image

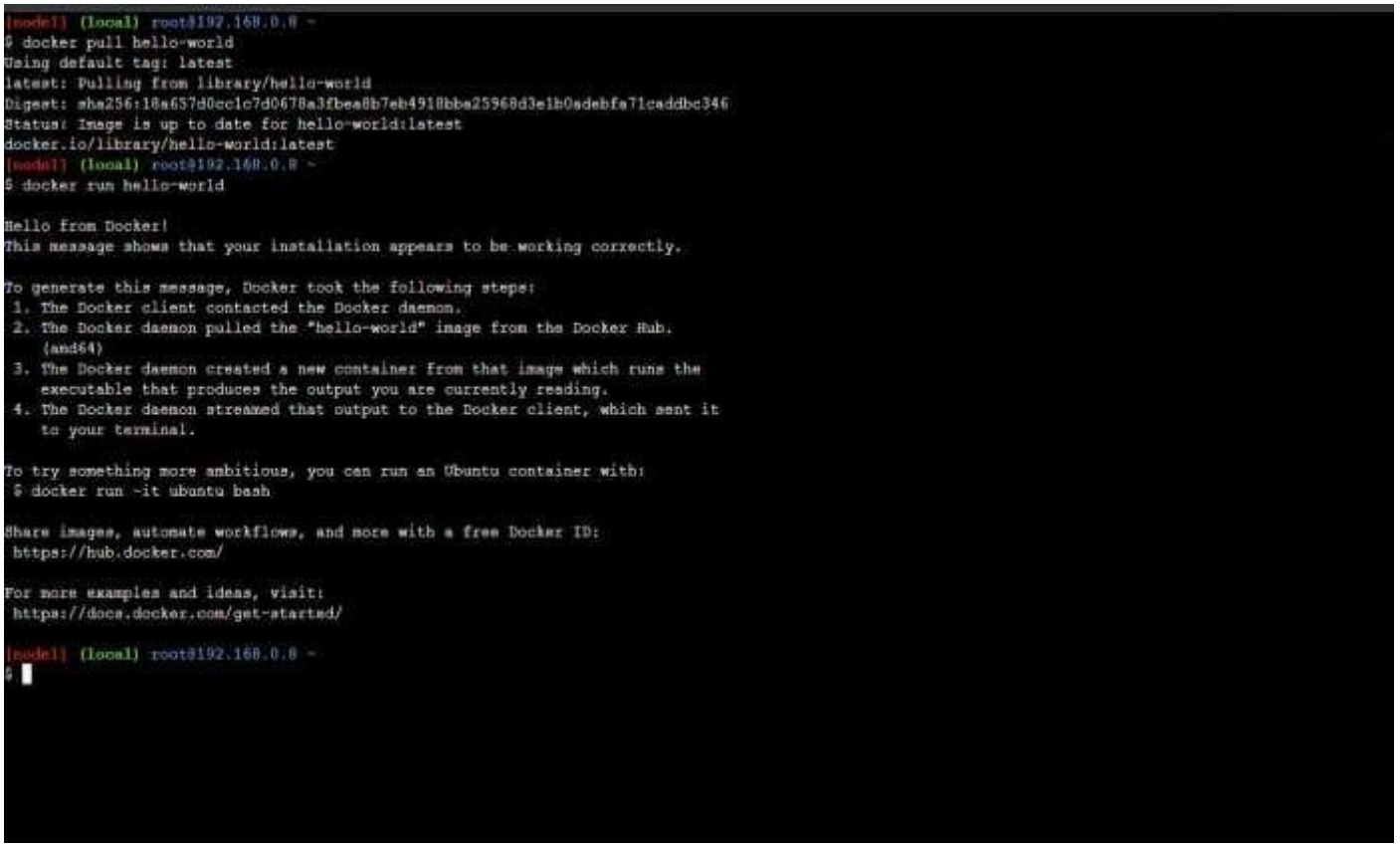
STEP: 2

- **Open Dockers playground**
- **Login with Dockers**
- **Create new instance**

STEP: 3

In the command prompt run the following:

$ **docker pull hello-world**
$ **docker run hello-world**



QUESTION 2:

Create a Dockers file and deploy it in Dockers desktop application

SOLUTION:

STEP: 1

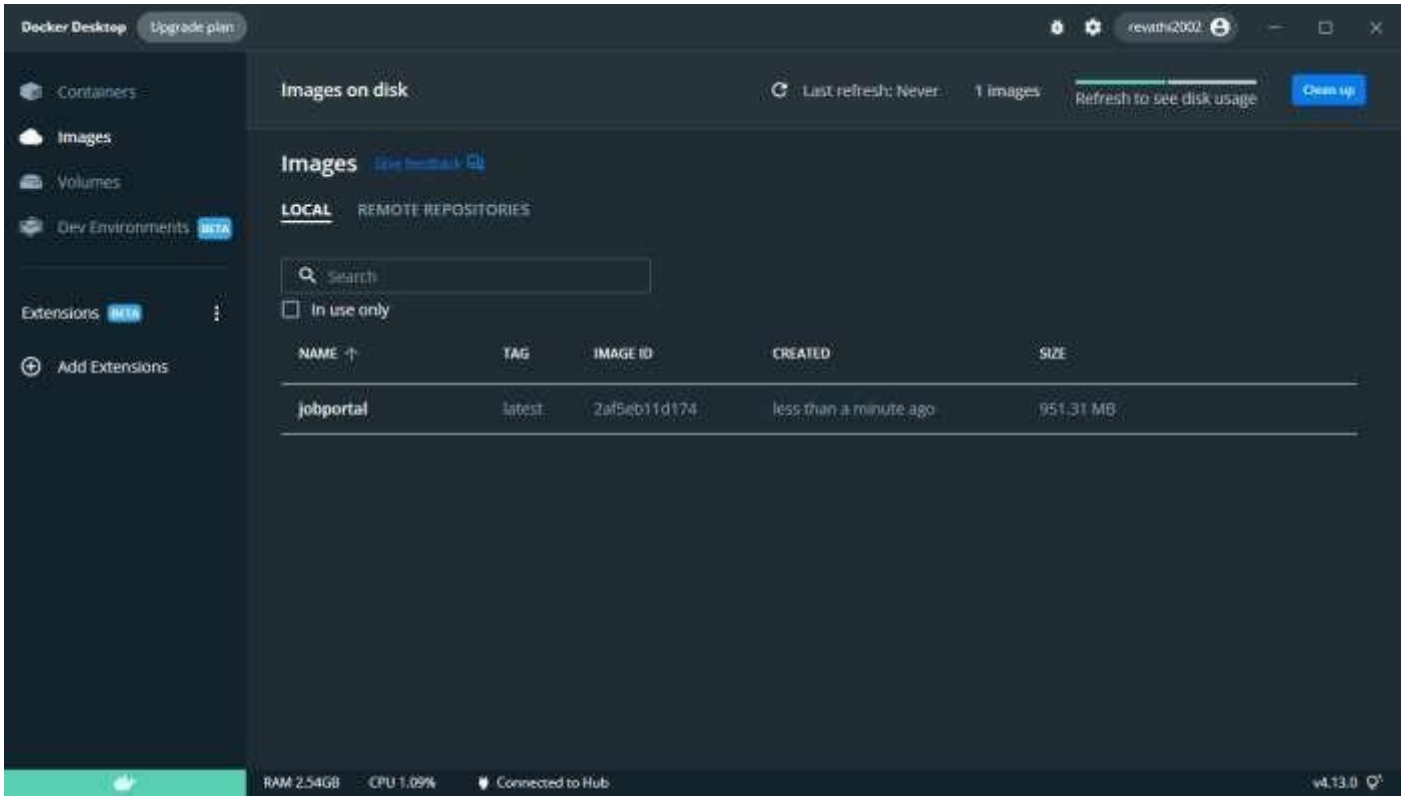
- **Create a flask application**
- **Create a Dockerfile in the same folder**

STEP: 2

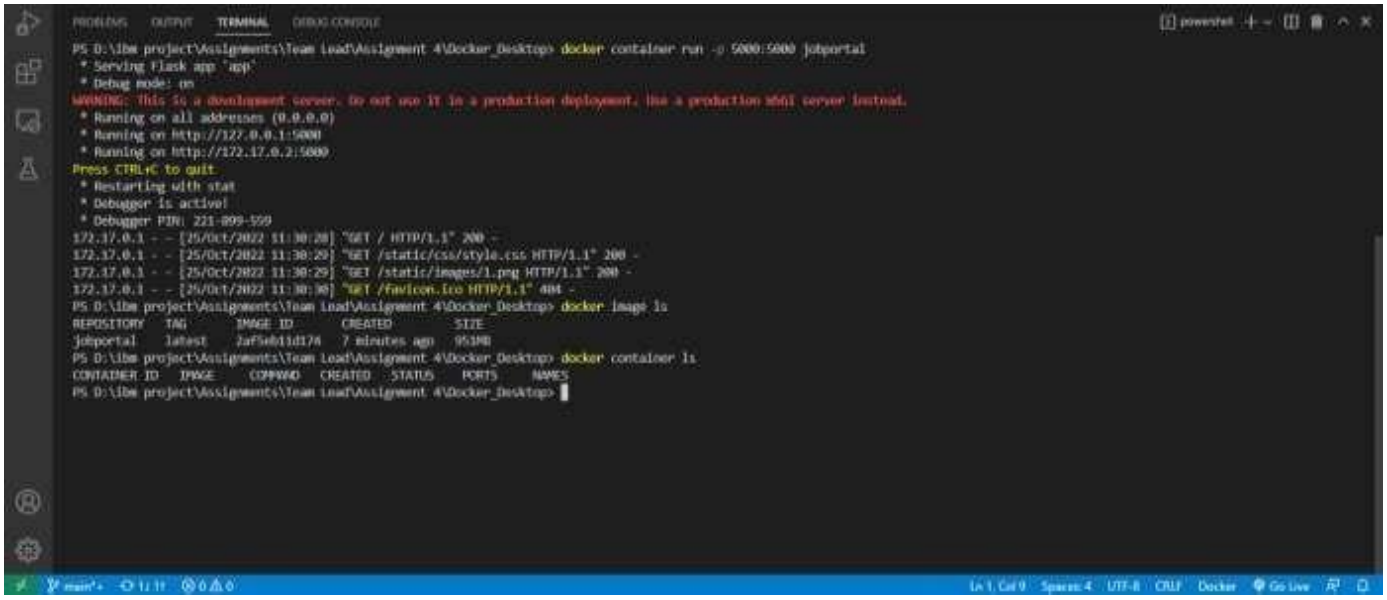
Run the following commands to deploy it in docker desktop

$ **docker build –t jobportal**

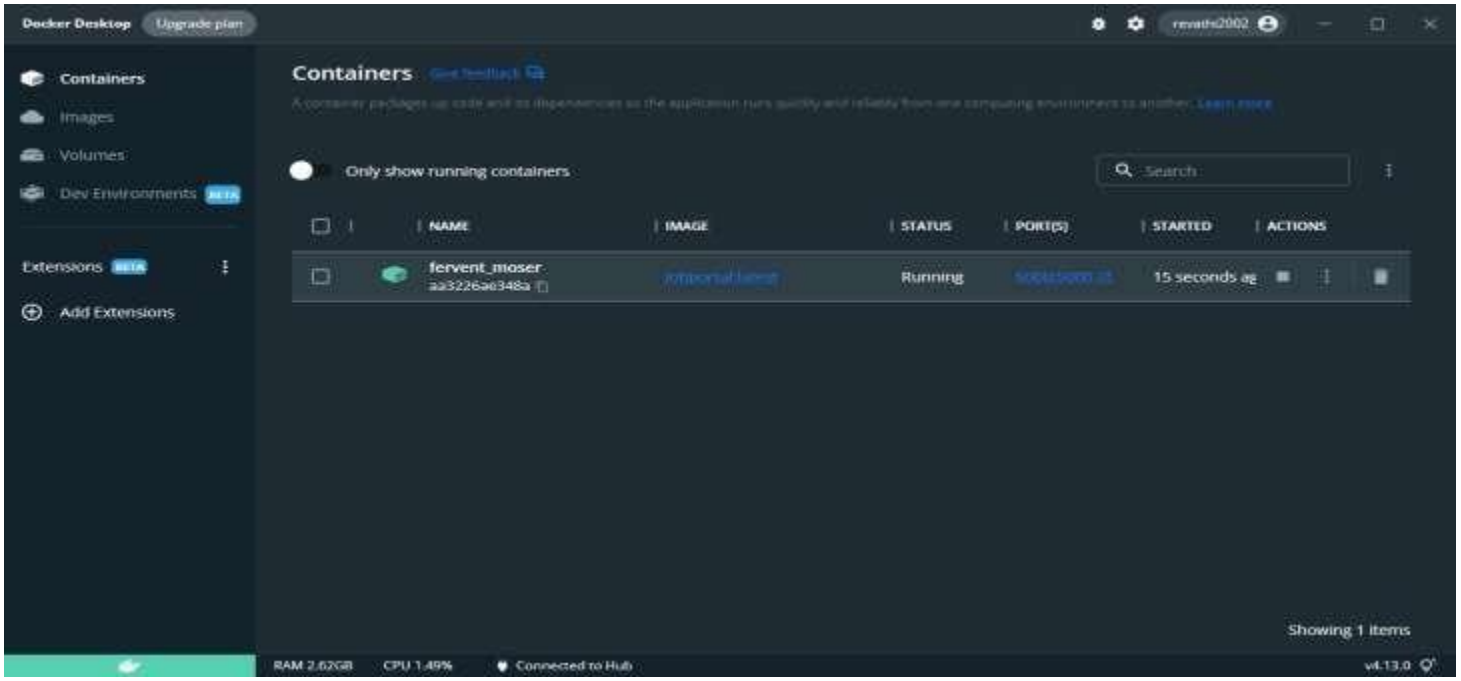
$ **docker image ls**



$ **docker container run –p 5000:5000 jobportal**



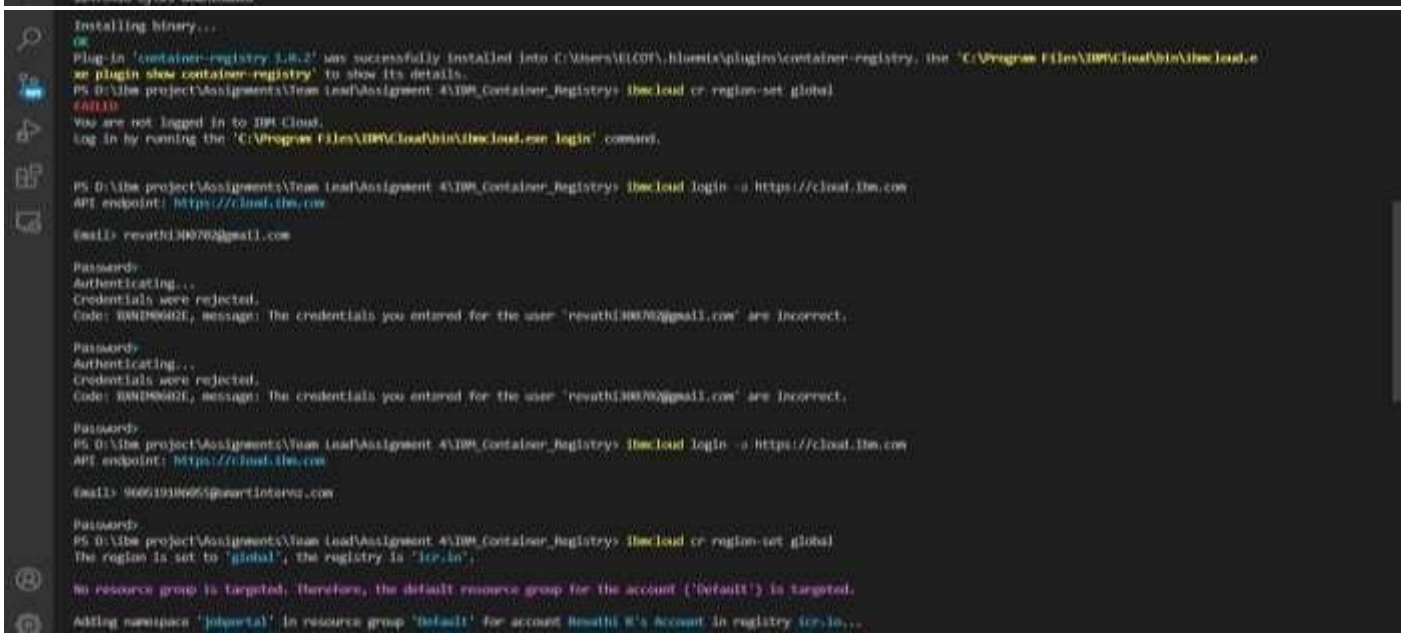
$ **docker container ls**

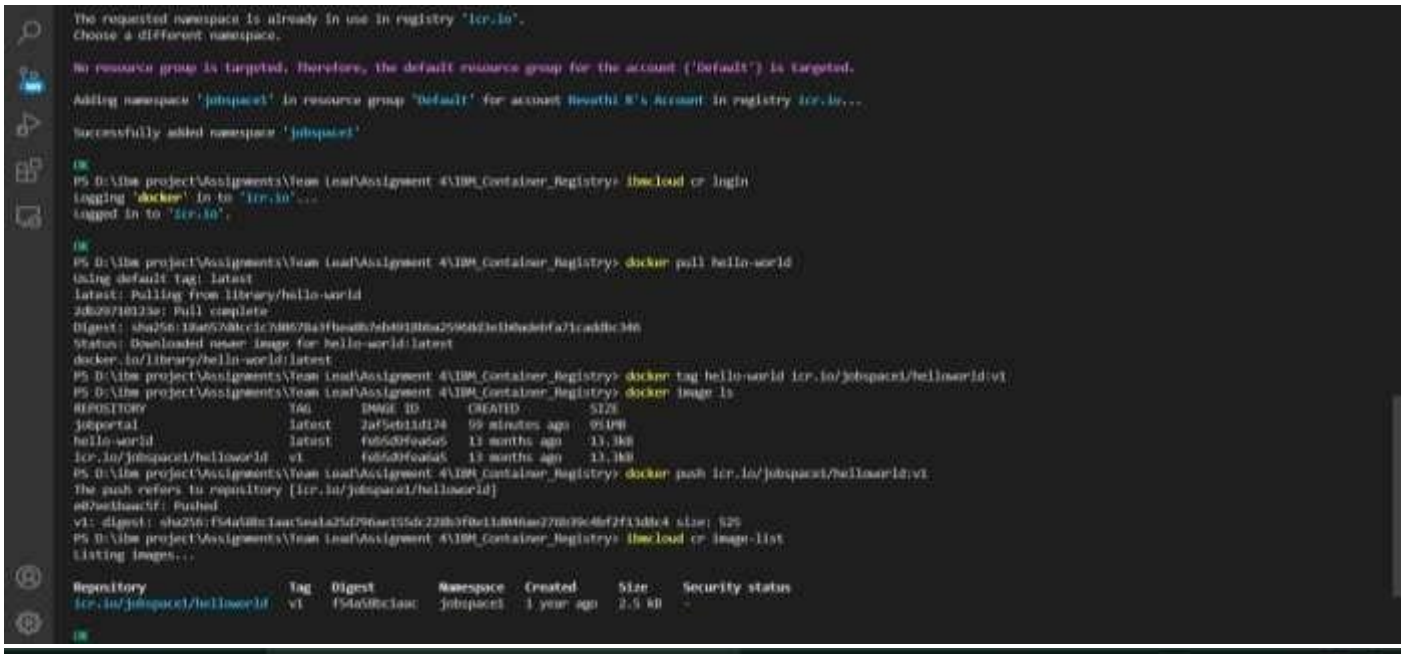


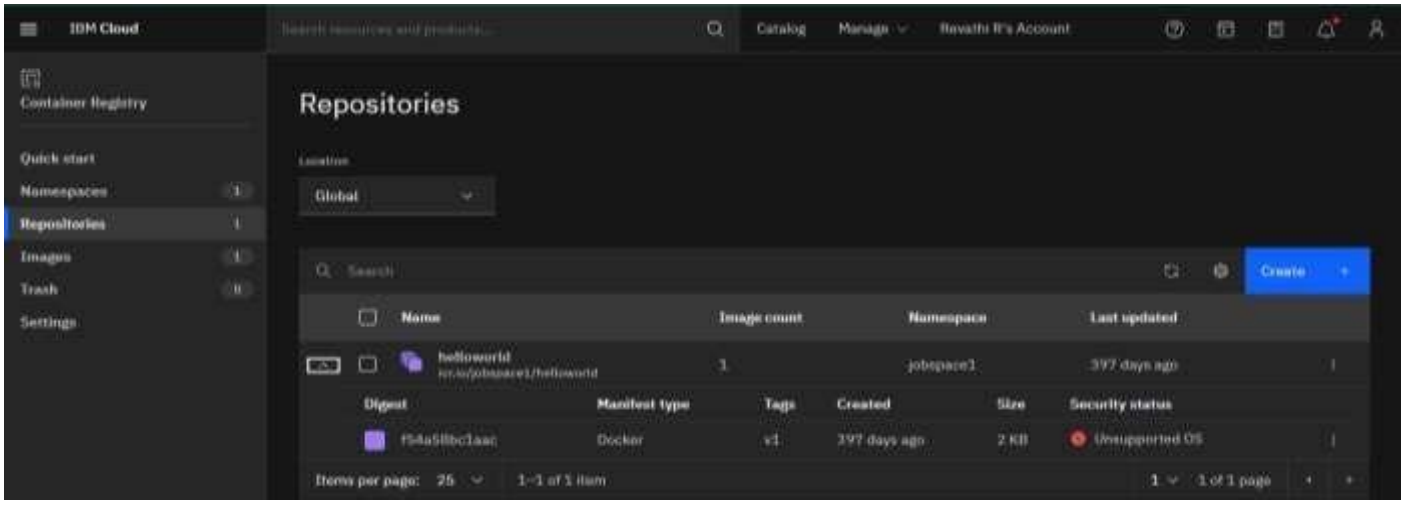
QUESTION 3:

Create an IBM container registry and deploy hello-world-app or job-portal-app

SOLUTION:





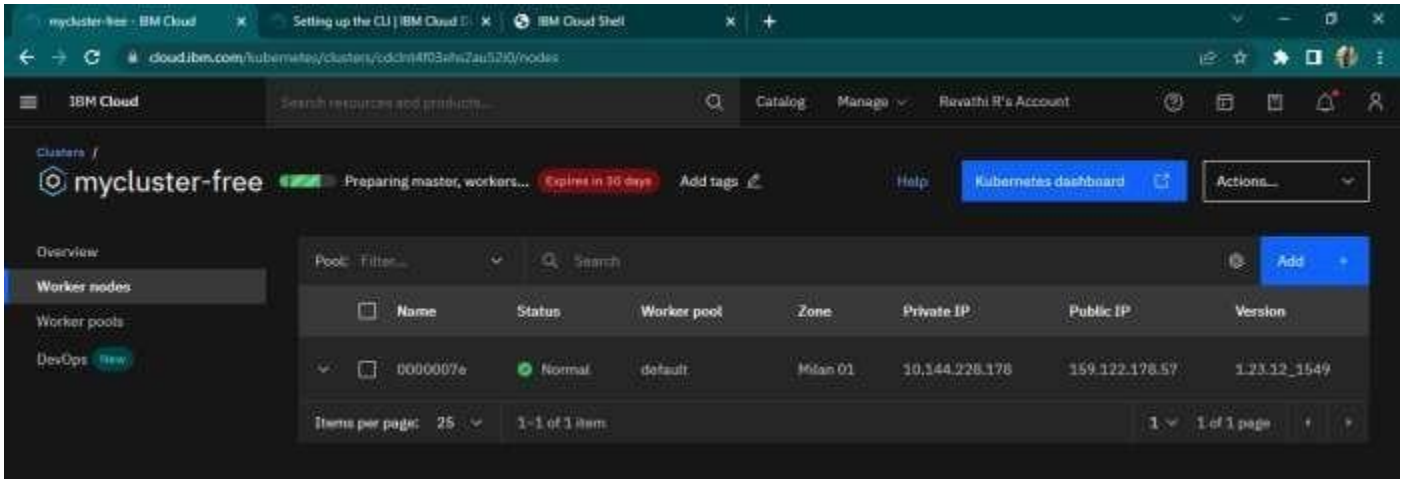


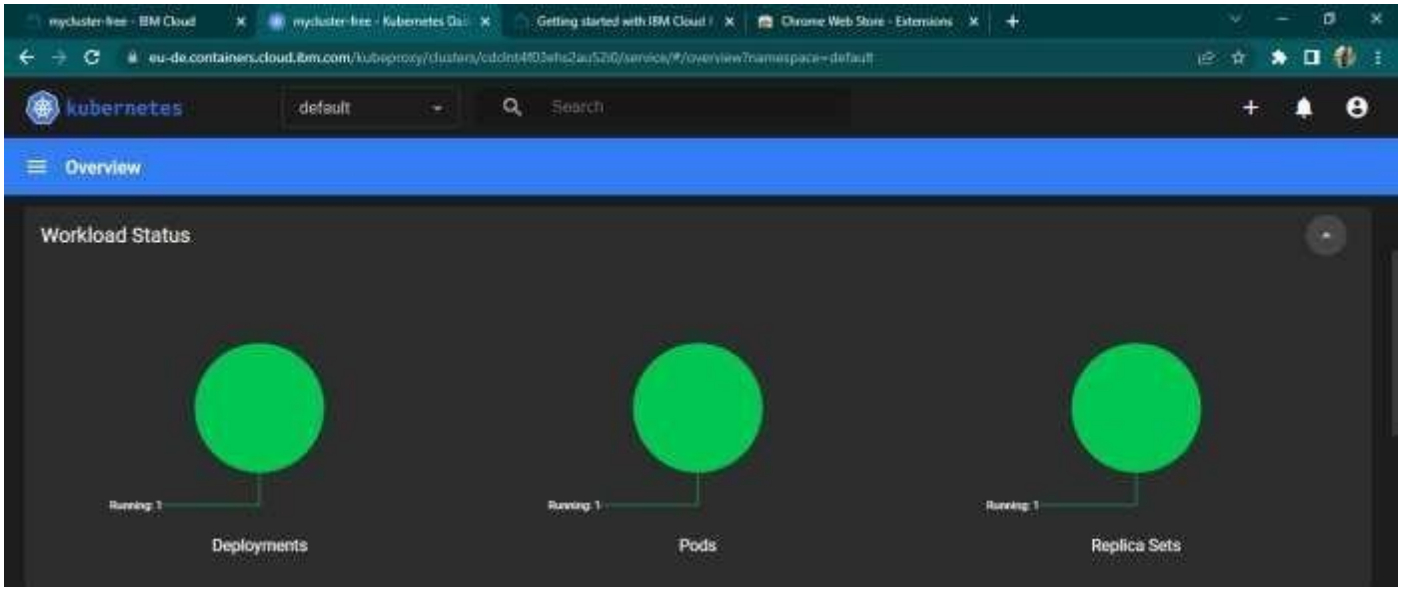
QUESTION 4:

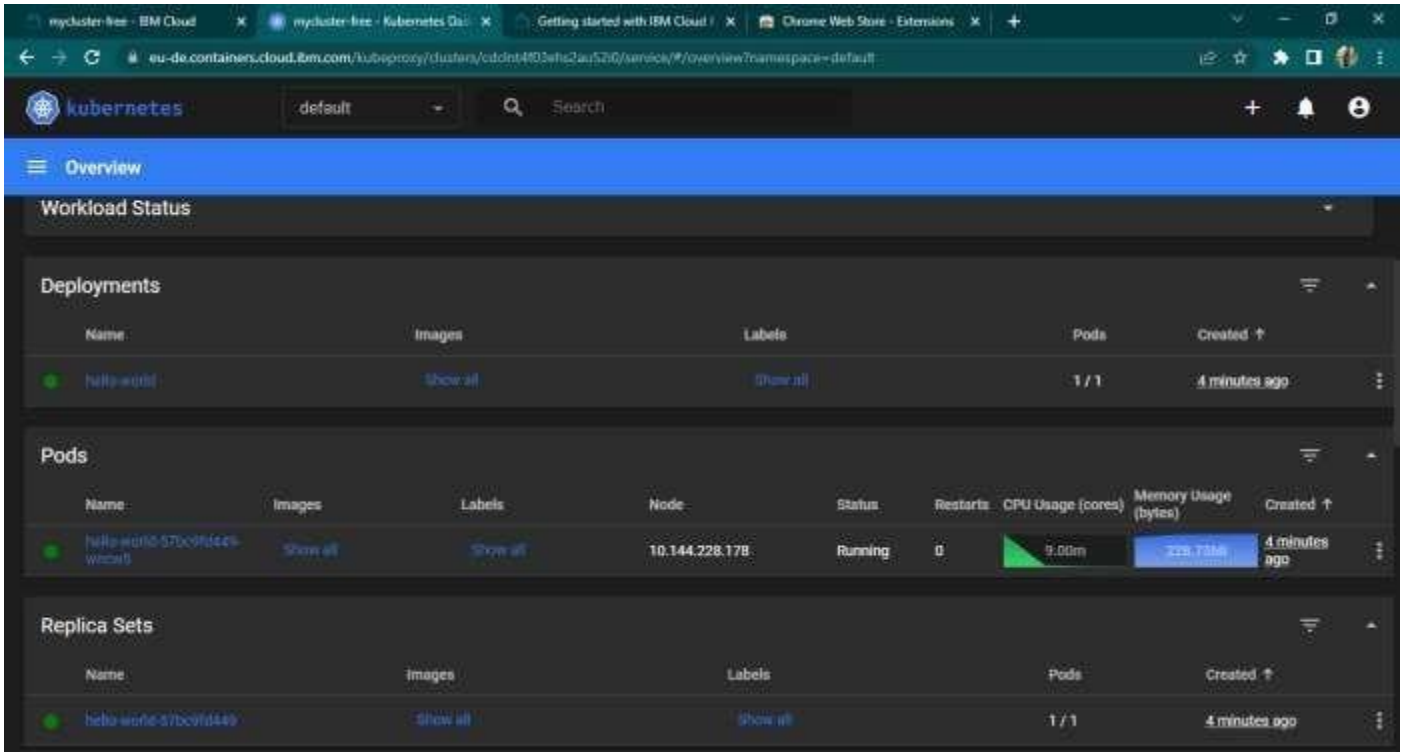
Create a Kubernetes cluster in IBM cloud and deploy hello-world-image or job-portal-image and also expose the same app to run in node-port.

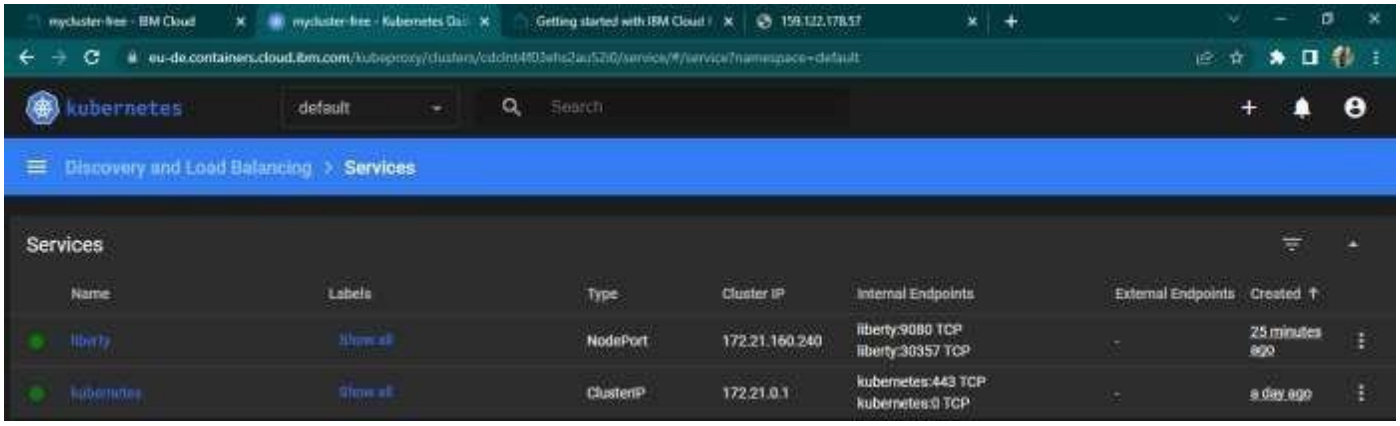
SOLUTION:

1. Select your cluster from the cluster list to open the details for your cluster.
2. Click Kubernetes dashboard.
3. From the menu bar, click the Create new resource icon (+).
4. Select the Create from form tab.
    a) Enter a name for your app, i.e **hello-world**.
    b) Enter **websphere-liberty** for your container image.
    c) Enter the number of pods for your app deployment, such as 1.
    d) Leave the Service drop-down menu set to None.
5. Click Deploy. During the deployment, the cluster downloads the websphere-liberty container image from Docker Hub and deploys the app in your cluster.
6. Create a node port so that your app can be accessed by other users internally or externally. Because your cluster is a free cluster, you can only expose an app with a node port, not a load balancer or Ingress.
    a) Click the Create new resource icon (+).
    b) Copy the node port YAML from GitHub.
    c) In the Create from input box, paste the node port YAML that you copied in the previous step.
    d) Click Upload. The node port service is created.
7. From the menu, click Services, and note the TCP endpoint port of your liberty service in the node port range 30000 - 32767, i.e **liberty:30357 TCP.**
8. From the menu, click Pods, and note the Node that your pod runs on, such as 10.xxx.xx.xxx.
9. Return to the IBM Cloud clusters console, select your cluster, and click the Worker Nodes tab. Find the Public IP of the worker node that matches the private IP of the node that the pod runs on.
10. In a tab in your browser, form the URL of your app by combining http:// with the public IP and TCP port that you previously retrieved i.e. **http:// 159.122.178.57: 30357**. The Welcome to Liberty page is displayed. Great job! You just deployed your first app in your Kubernetes cluster.




kubernetes
default
Search
Overview
Workload Status
Running: 1
Deployments
Running: 1
Pods
Running: 1
Replica Sets


kubernetes
default
Search
Overview
Workload Status
Deployments
Name
Images
Labels
Pods
Created
1 / 1
4 minutes ago
Pods
Name
Images
Labels
Node
Status
Restarts
CPU Usage (cores)
Memory Usage (bytes)
Created
10.144.228.178
Running
0
4 minutes ago
Replica Sets
Name
Images
Labels
Pods
Created
1 / 1
4 minutes ago


kubernetes
default
Search
Discovery and Load Balancing > Services
Services
Name
Labels
Type
Cluster IP
Internal Endpoints
External Endpoints
Created
liberty
NodePort
172.21.160.240
liberty:9080 TCP
liberty:30357 TCP
25 minutes ago
kubernetes
ClusterIP
172.21.0.1
kubernetes:443 TCP
kubernetes:0 TCP
a day ago