Assignment - 4

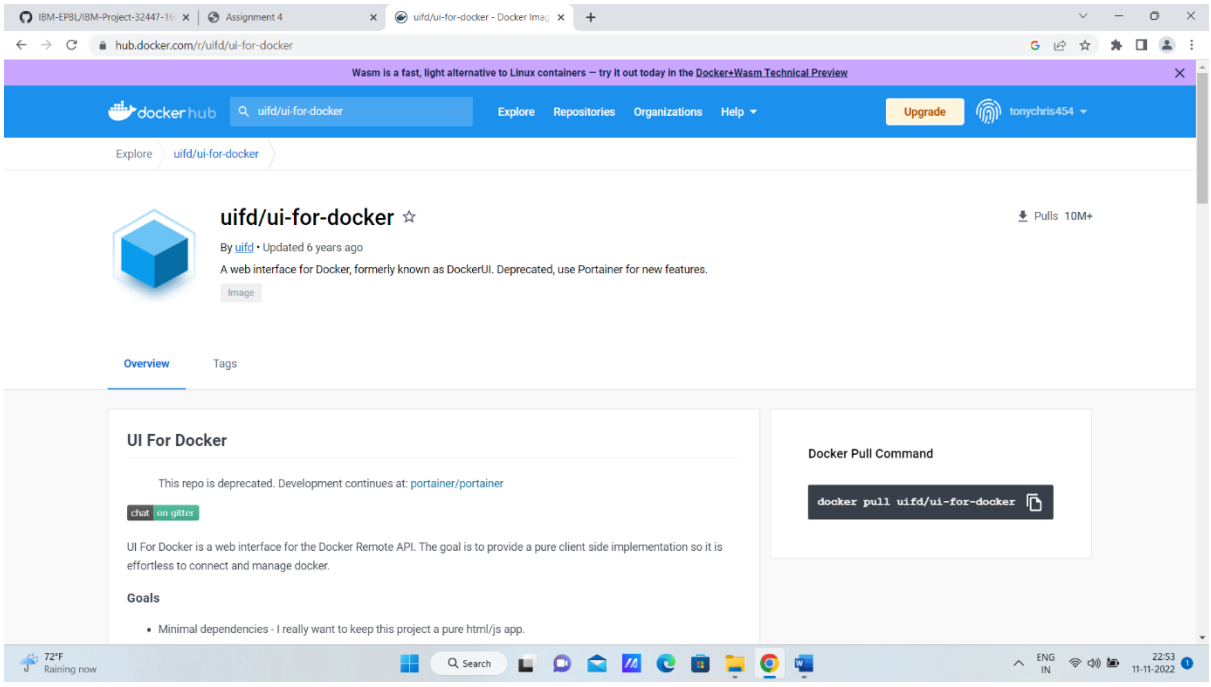
Assignment Date	22 October 2022
Student Name	Nirmal Kumar M
Student Roll No	812419104701
Maximum Marks	2 Marks

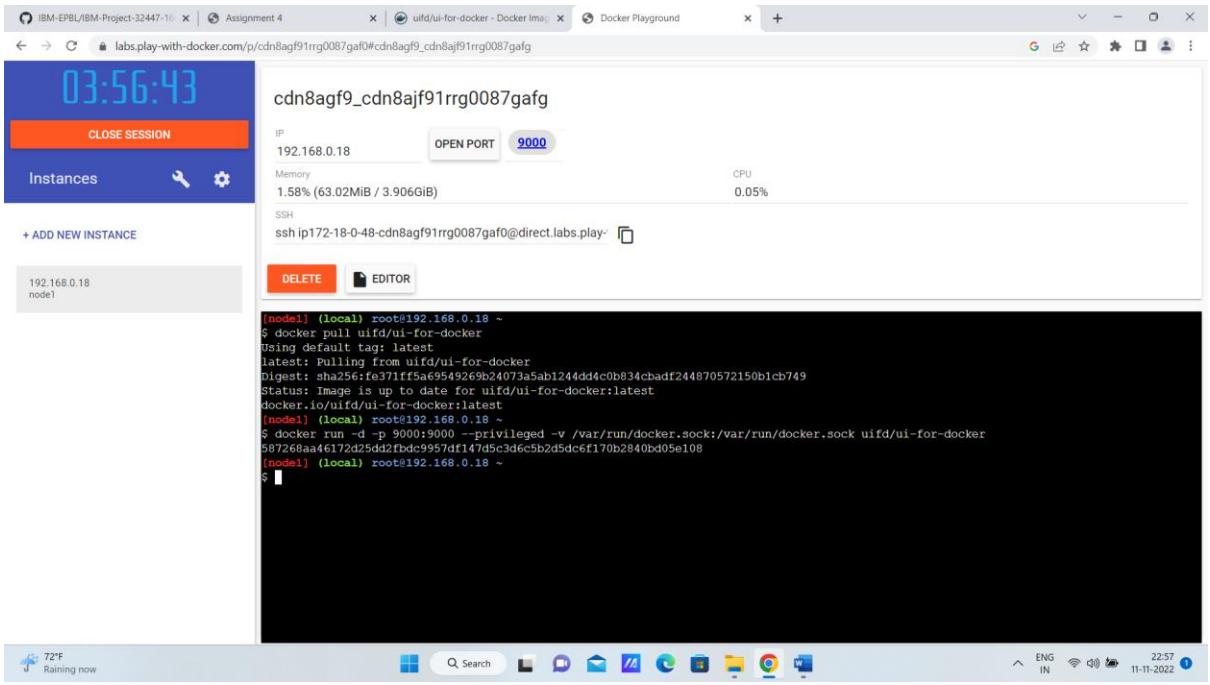
Question-1:

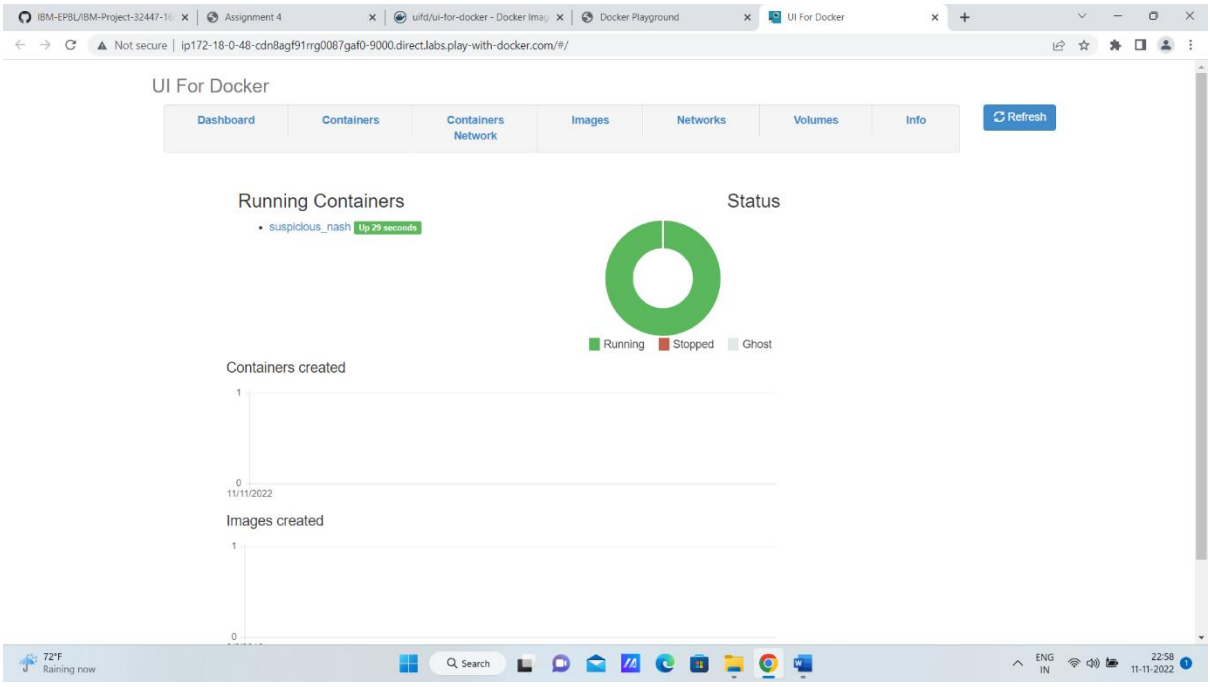
Pull an Image from docker hub and run it in Docker Playground

Solution:

a. Pull an image *uifd/ui-for-docker* from the docker hub
b. This image is used for viewing and managing the docker engine
c. Use **docker pull image name**and **docker run -it image name**commands to run the above image in the Docker Playground




labs.play-with-docker.com/p/cdn8agf91rrg0087gaf0#cdn8agf9_cdn8ajf91rrg0087gafg
03:56:43
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
cdn8agf9_cdn8ajf91rrg0087gafg
IP
192.168.0.18
OPEN PORT
9000
Memory
1.58% (63.02MiB / 3.906GiB)
CPU
0.05%
SSH
ssh ip172-18-0-48-cdn8agf91rrg0087gaf0@direct.labs.play-
DELETE
EDITOR
[node1] (local) root@192.168.0.18 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749
Status: Image is up to date for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[node1] (local) root@192.168.0.18 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
587268aa46172d25dd2fbdc9957df147d5c3d6c5b2d5dc6f170b2840bd05e108
[node1] (local) root@192.168.0.18 ~
$
72°F
Raining now
Search
ENG
IN
22:57
11-11-2022


Not secure | ip172-18-0-48-cdn8agf91rrg0087gaf0-9000.direct.labs.play-with-docker.com/#/
UI For Docker
Dashboard
Containers
Containers Network
Images
Networks
Volumes
Info
Refresh
Running Containers
suspicious_nash
Up 29 seconds
Status
Running
Stopped
Ghost
Containers created
1
0
11/11/2022
Images created
1
0
72°F
Raining now
Search
ENG
IN
22:58
11-11-2022

Question-2:

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

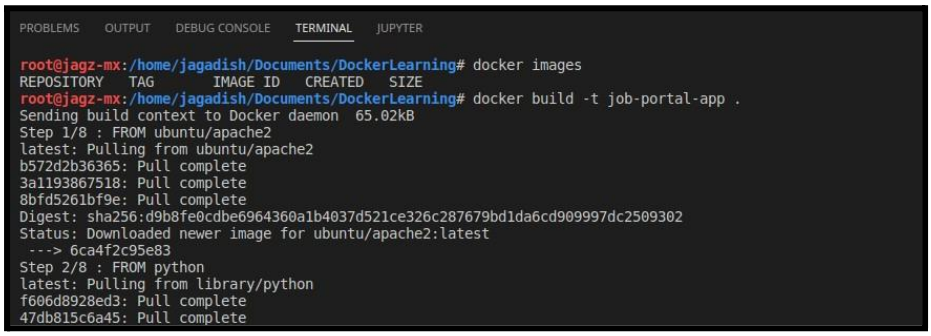
Solution:

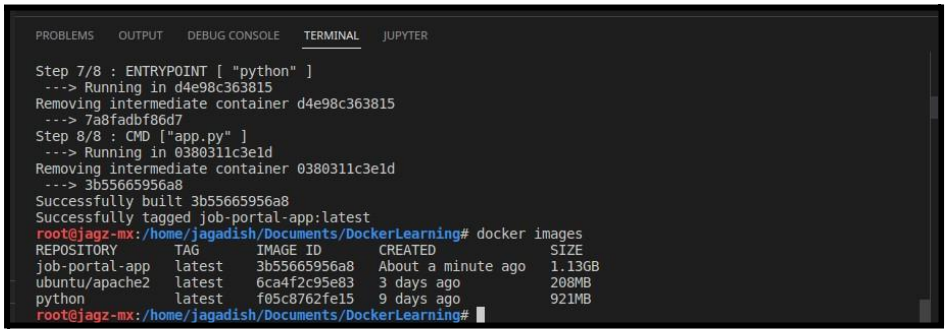
d. Create a docker file for build and deploy flask app.
e. Use **docker build -t image_name.**in the current directory to start building the docker image and deploy in our local docker
f. Use **docker run -p 5000:5000 image_name**to run in local system

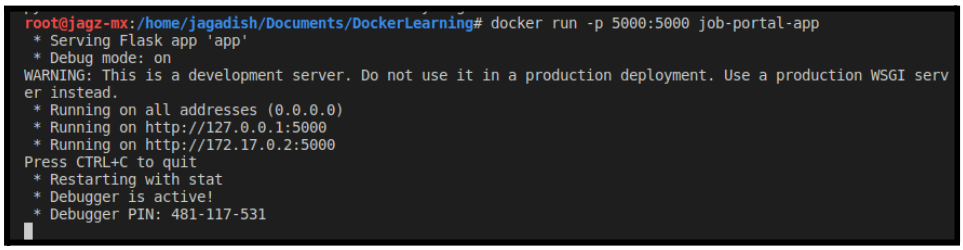
Dockerfile

```
FROM ubuntu/apache2
FROM python
COPY ./requirements.txt /flaskApp/requirements.txt WORKDIR /flaskApp
RUN pip install -r requirements.txt COPY . /flaskApp
ENTRYPOINT [ "python" ] CMD
["app.py" ]
```

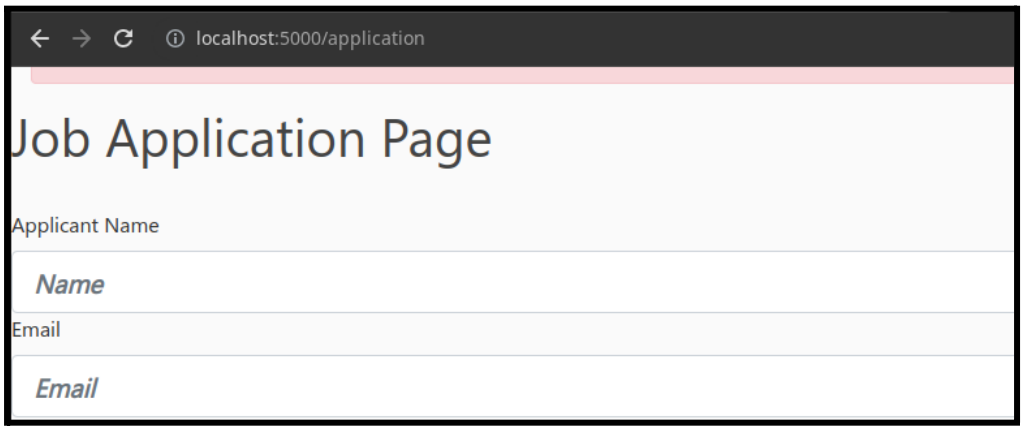
Steps Involved







Run locally using docker

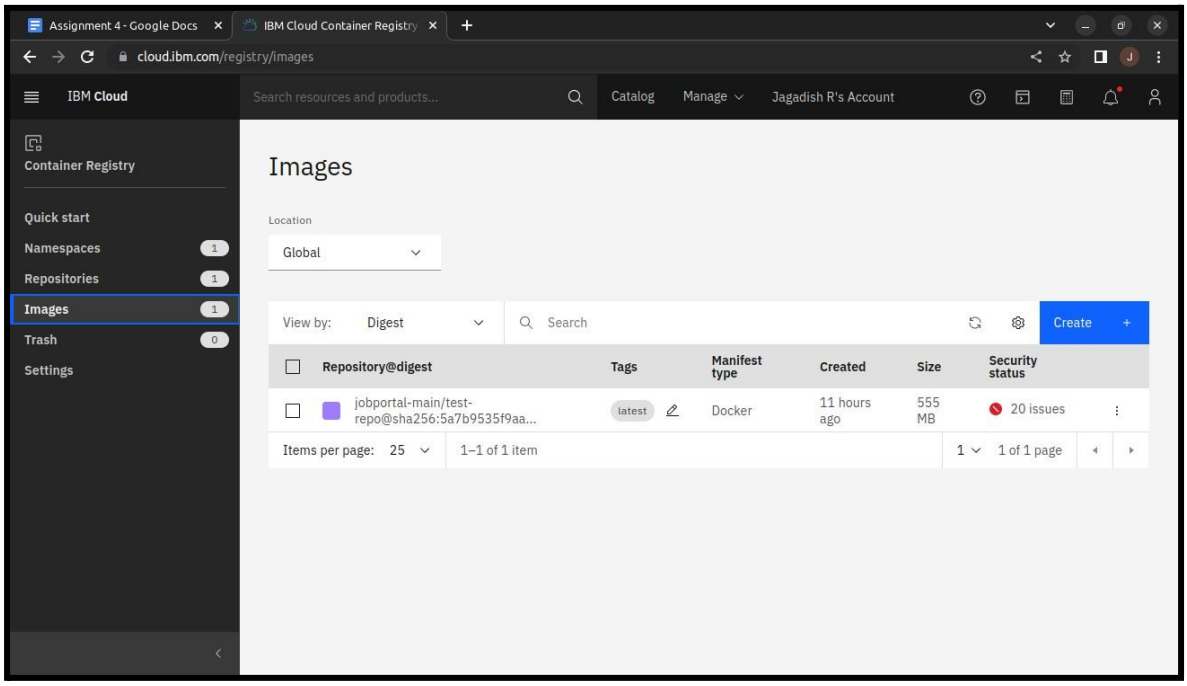


Question-3:

Create a IBM container registry and deploy helloworld app or jobportal app

Solution:

g. Log into IBM cloud
h. Create a **container registry**
i. Using IBM Cloud CLI, install the **container registry plugin** in our system
j. Push our docker image into the created container registry using **docker push**
k. So, our job portal app is deployed in the IBM container registry



Question-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

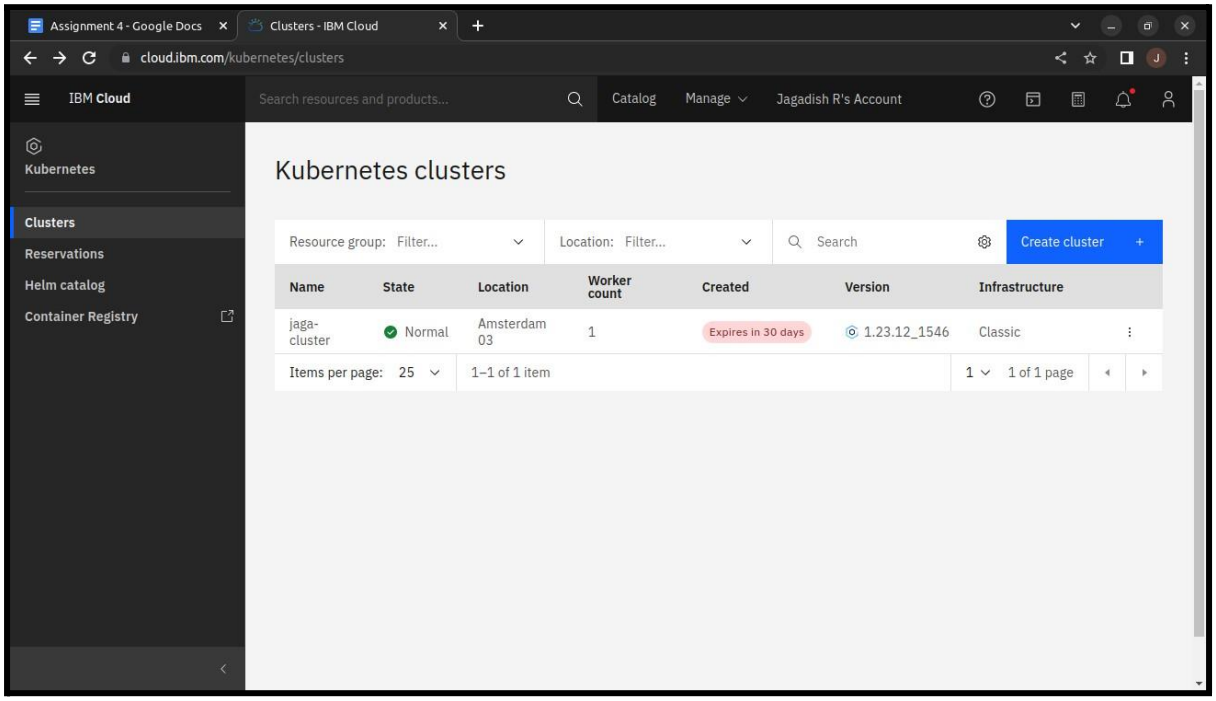
Solution:

l. Log into IBM cloud
m. Create a **kubernete**
n. Using IBM Cloud CLI, install the **ks plugin** in our system
o. Create **a cluster** in the kubernetes
p. Now, go to the **kubernetes dashboard** where we need to create a service based on a yml file (given below)
q. In that file, we have to mention *which image we are going to use* and the *app name*
r. Take the **public IP address** and **Nodeport** since we exposed the *flask app innodeport*
s. Finally, we got the **url address** where our flask app is hosted

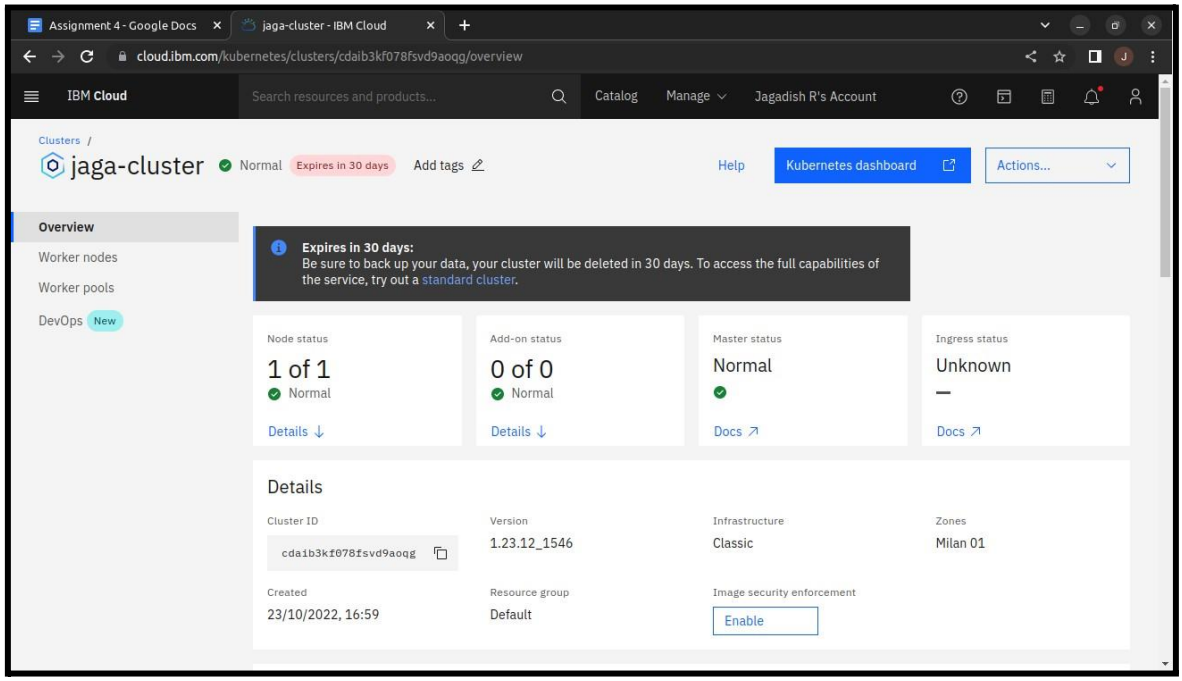
job-portal-app.yml

```
apiVersion: v1 kind:
Service metadata:
    name: job-portal-app spec:
    selector:
        app: job-portal-app ports:
    - port: 5000 type:
    NodePort
---
apiVersion: apps/v1 kind:
Deployment metadata:
    name: job-portal-app labels:
        app: job-portal-app spec:
    selector:
        matchLabels:
            app: job-portal-app replicas:
    1
    template:
        metadata:
            labels:
                app: job-portal-app spec:
            containers:
            - name: job-portal-app image:
                image_name ports:
                - containerPort: 5000 env:
                - name: DISABLE_WEB_APP
                  value: "false"
```

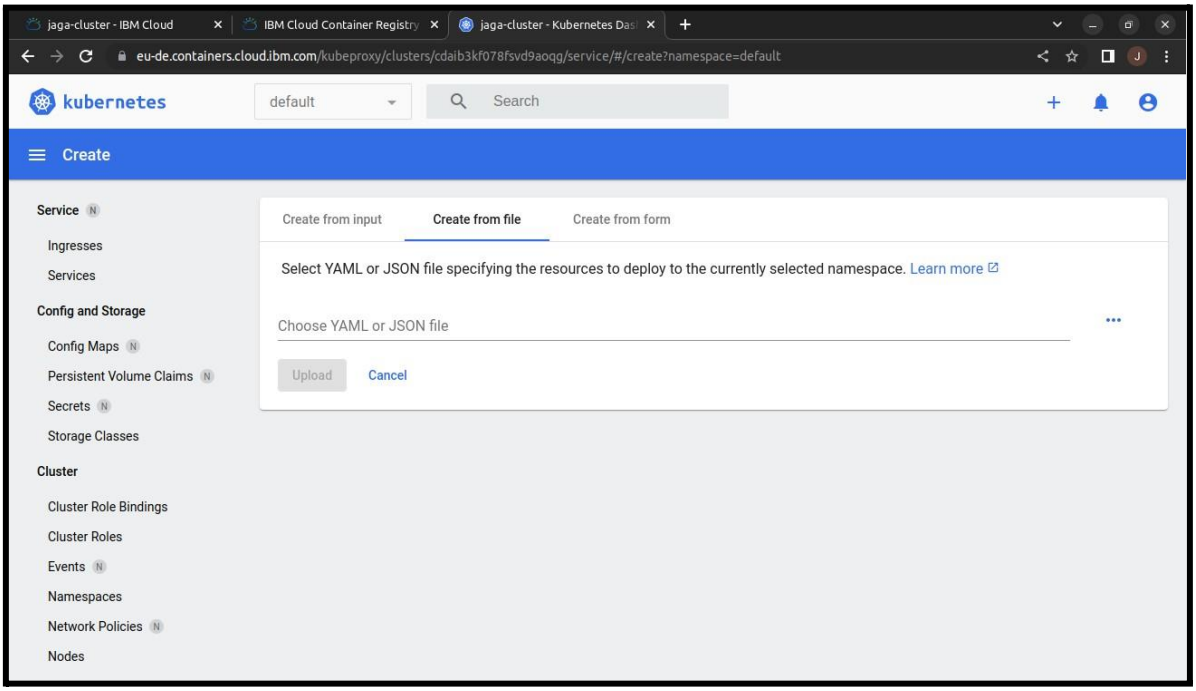
Cluster creation



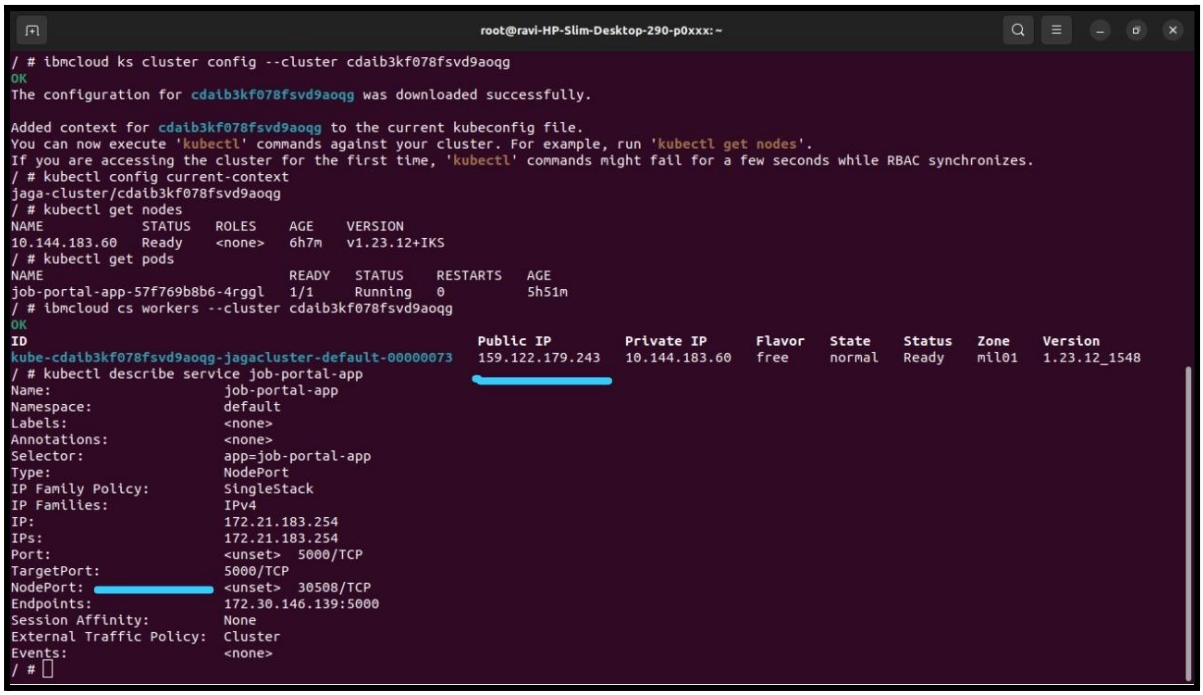
Configuring the cluster



Creating a service based on the yml file



Procedure to find the exposed url



Run our flask app in the IBM kubernetes