Assignment 4

|  |  |
| --- | --- |
| Team ID | PNT2022TMID14121 |
| Project Name | Project – Inventory Management System for  retailers. |

Question:

1. Pull an Image from docker hub and run it in docker playground.
2. Create a docker file for the job portal application and deploy it in Docker desktop application.
3. Create an IBM container registry and deploy hello world app or job portal app.
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.

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

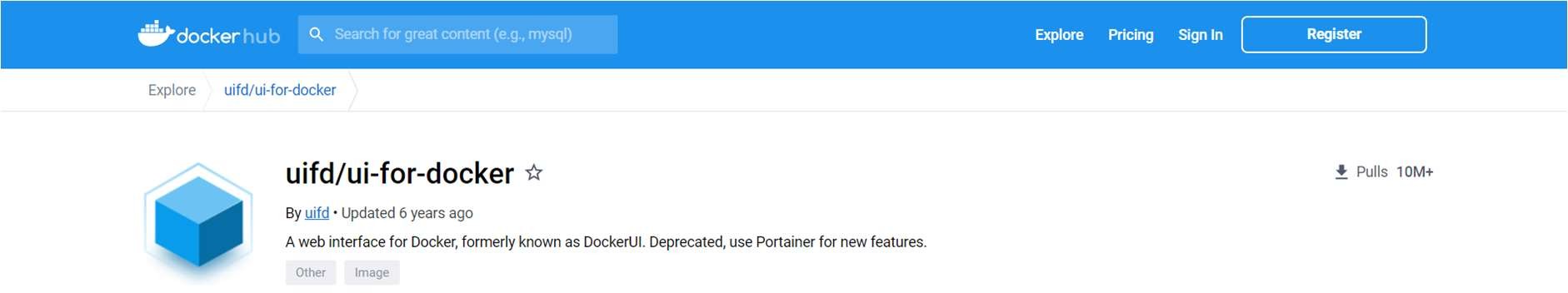
**uifd/ui-for-docker: UI Foí Dockeí is a web inteíface foí the Dockeí Remote API. ľhe goal is to píovide a puíe client-side implementation so it is effoítless to connect and manage dockeí.**

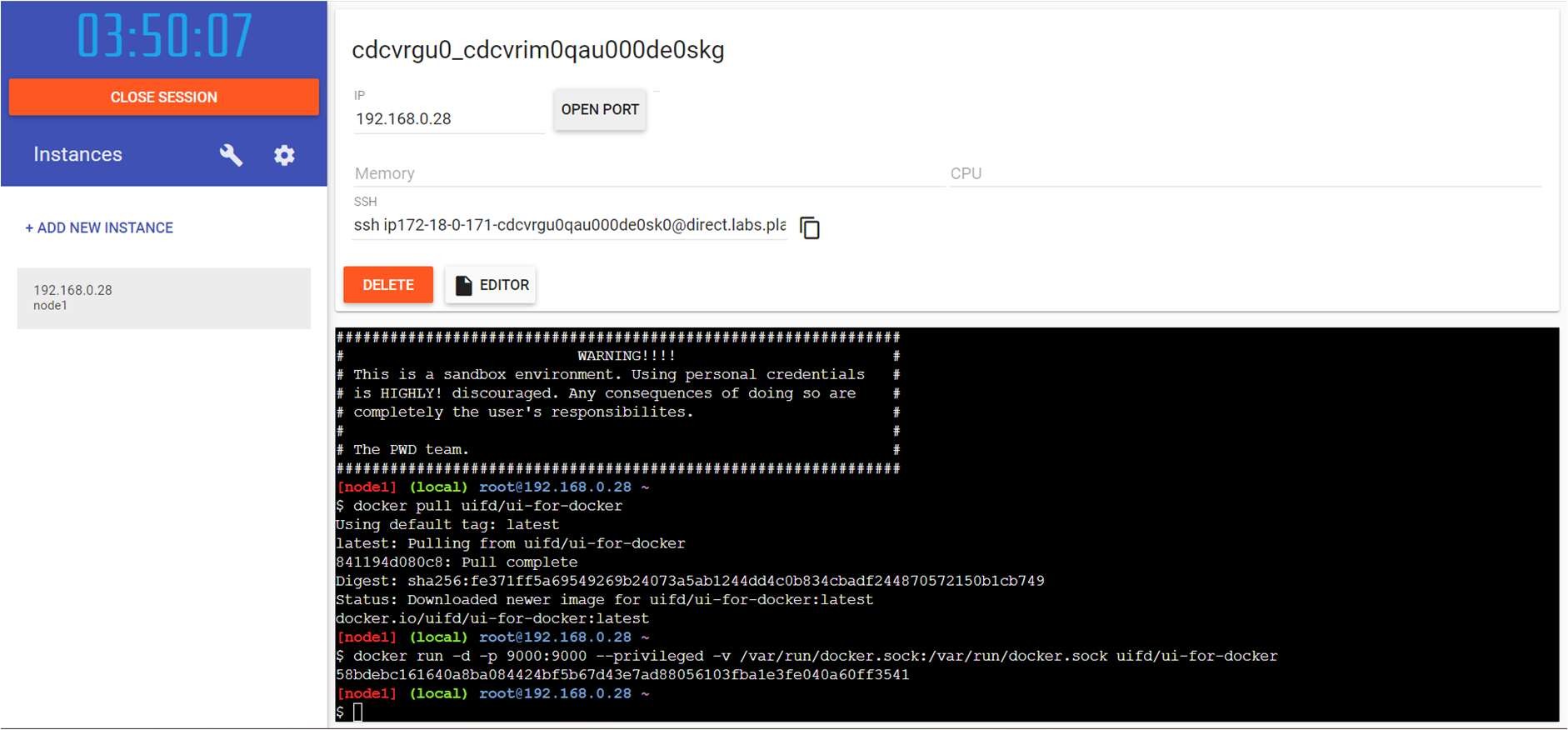
**Pull the** uifd/ui-for-docker image from the docker hub.

**Pull uifd/ui-for-docker:** docker pull uifd/ui-for-docker.

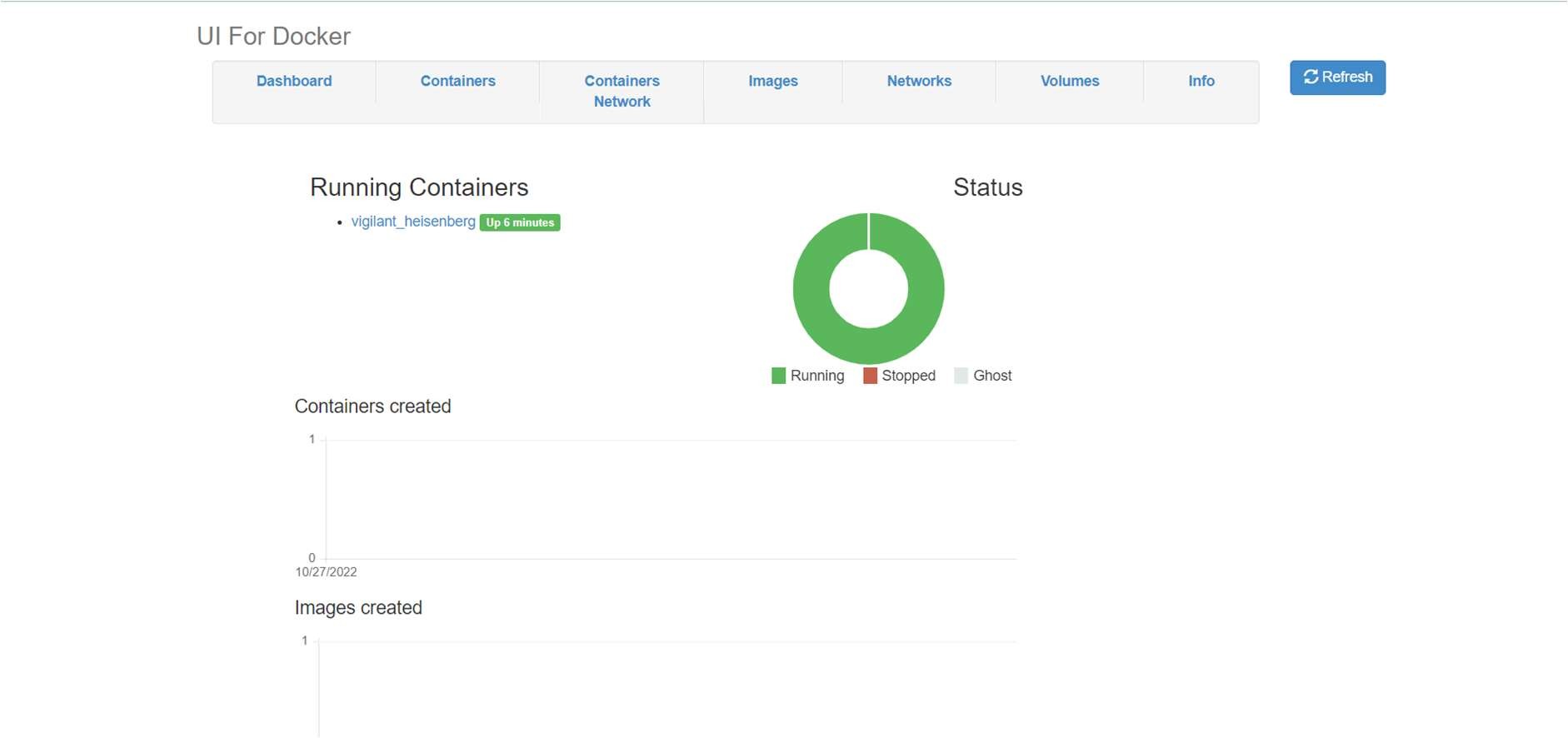
**Run uifd/ui-for-docker:** docker run -d -p 9000:9000 --privileged -v

/var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker.





**Open your browser to** [http://ip172-18-0-171-cdcvrgu0qau000de0sk0-](http://ip172-18-0-171-cdcvrgu0qau000de0sk0-/) 9000.direct.labs.play-with-docker.com/#/



# Create a docker file for the job portal application and deploy it in Docker desktop application.

**Docker File:**

FROM python:latest WORKDIR Job\_Application

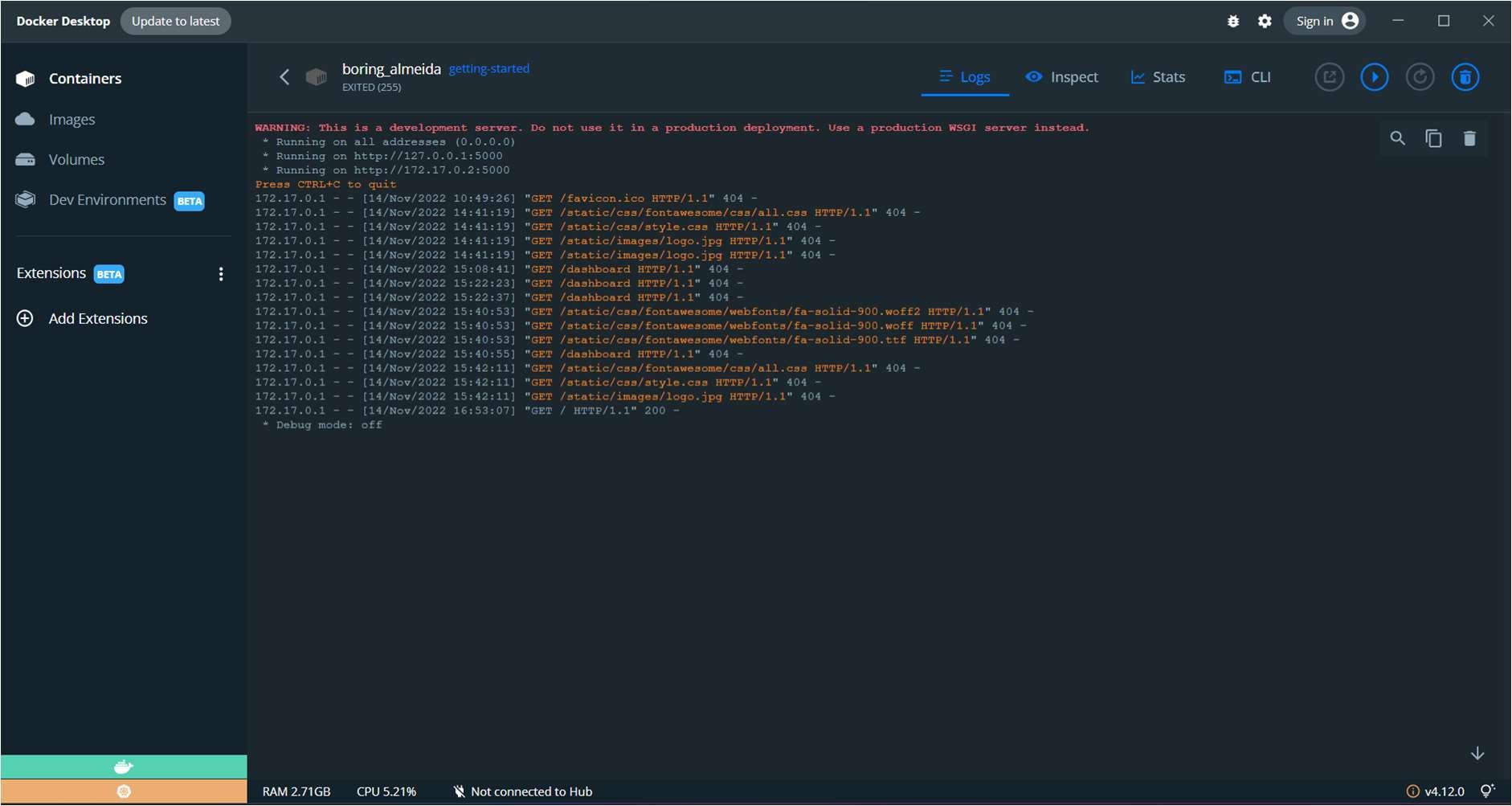
COPY requirements.txt requirements.txt RUN pip3 install -r requirements.txt COPY . .

CMD [ "python", "-m" , "flask", "run", "--host=0.0.0.0"] EXPOSE 5000

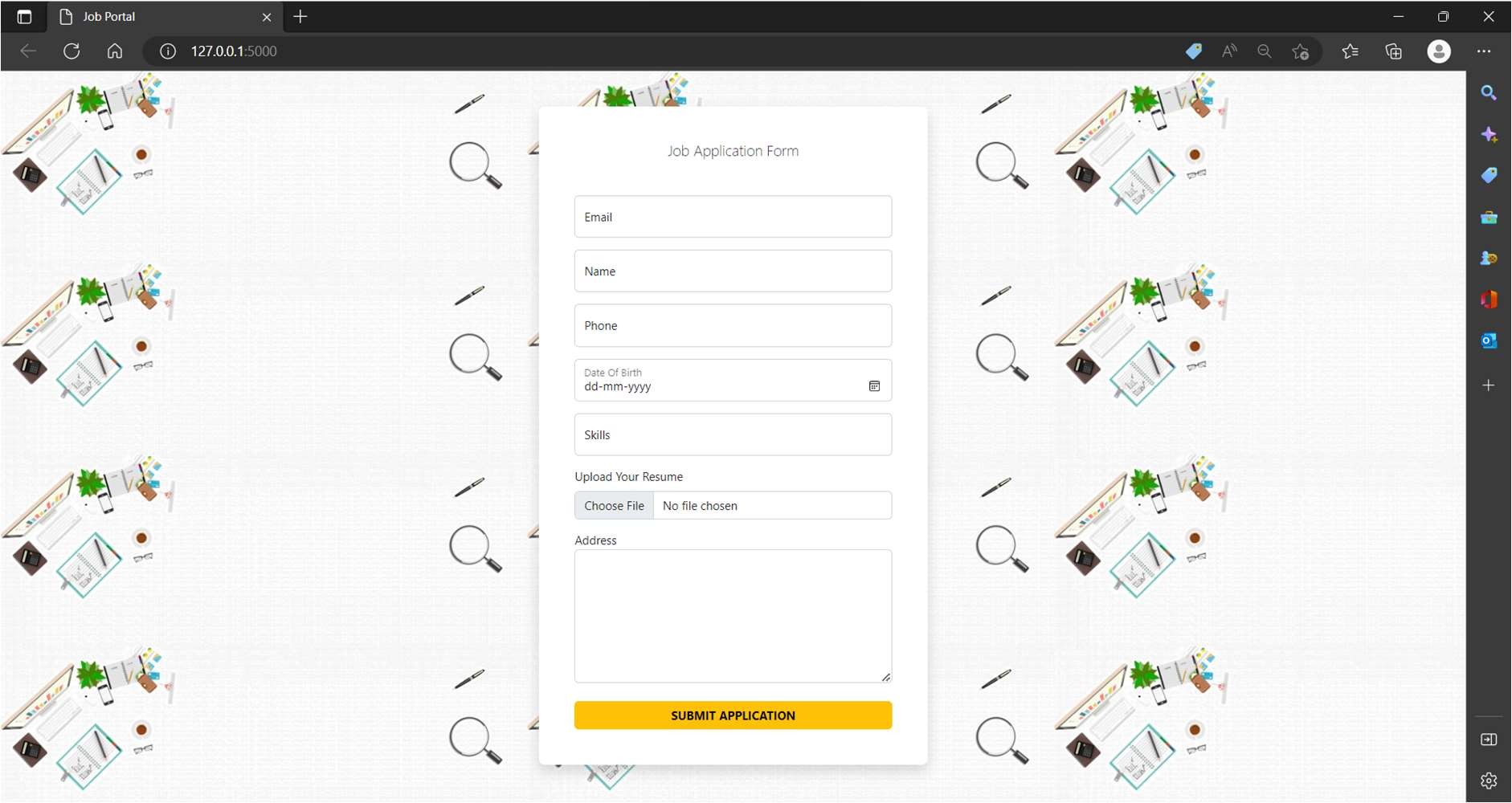
# Requirements:

flask

# Docker Desktop:



**Output:**



# a.yaml

apiVersion: apps/v1 kind: Deployment metadata:

name: flask-app spec:

replicas: 3 selector:

matchLabels:

app: flask-app template:

metadata:

labels:

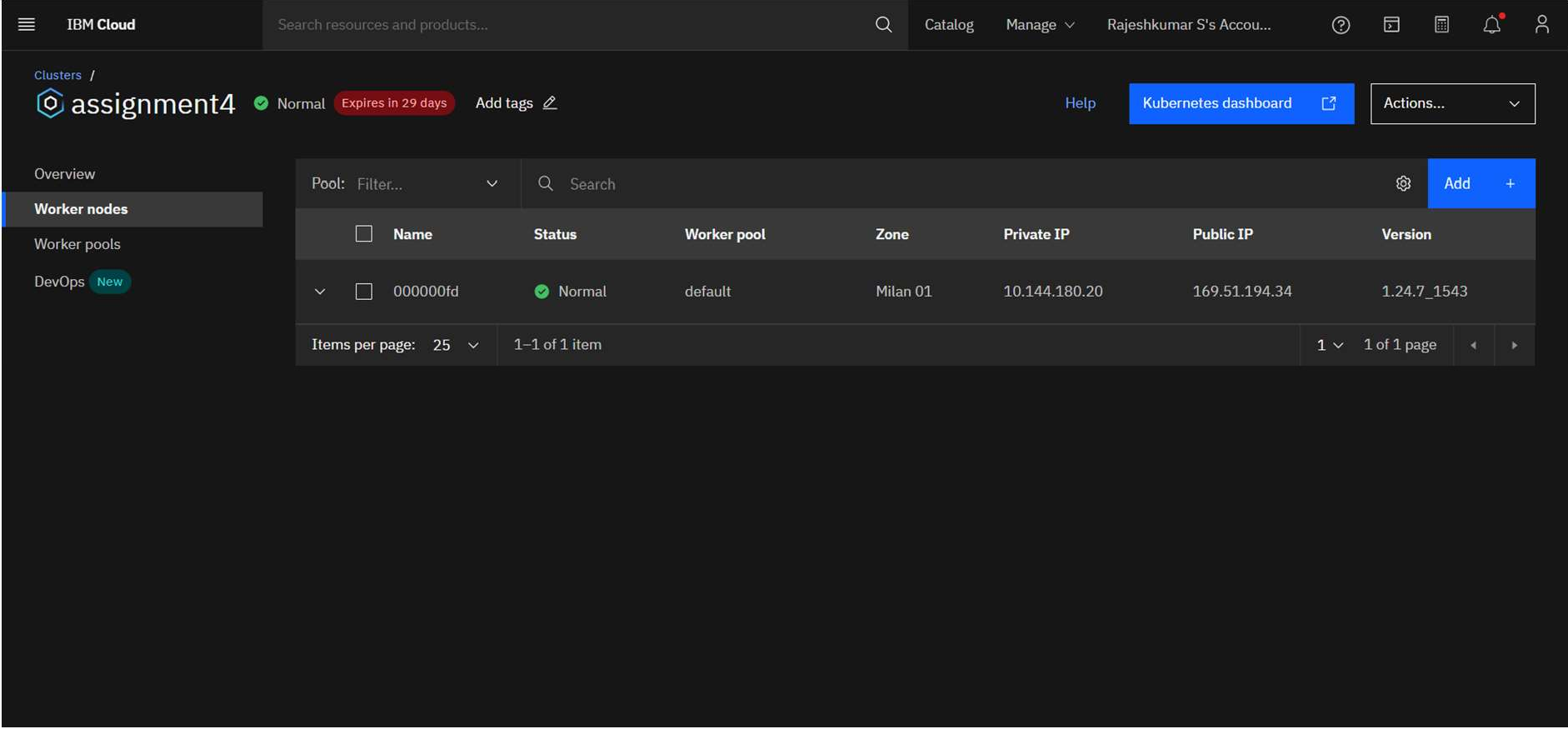
app: flask-app spec:

containers:

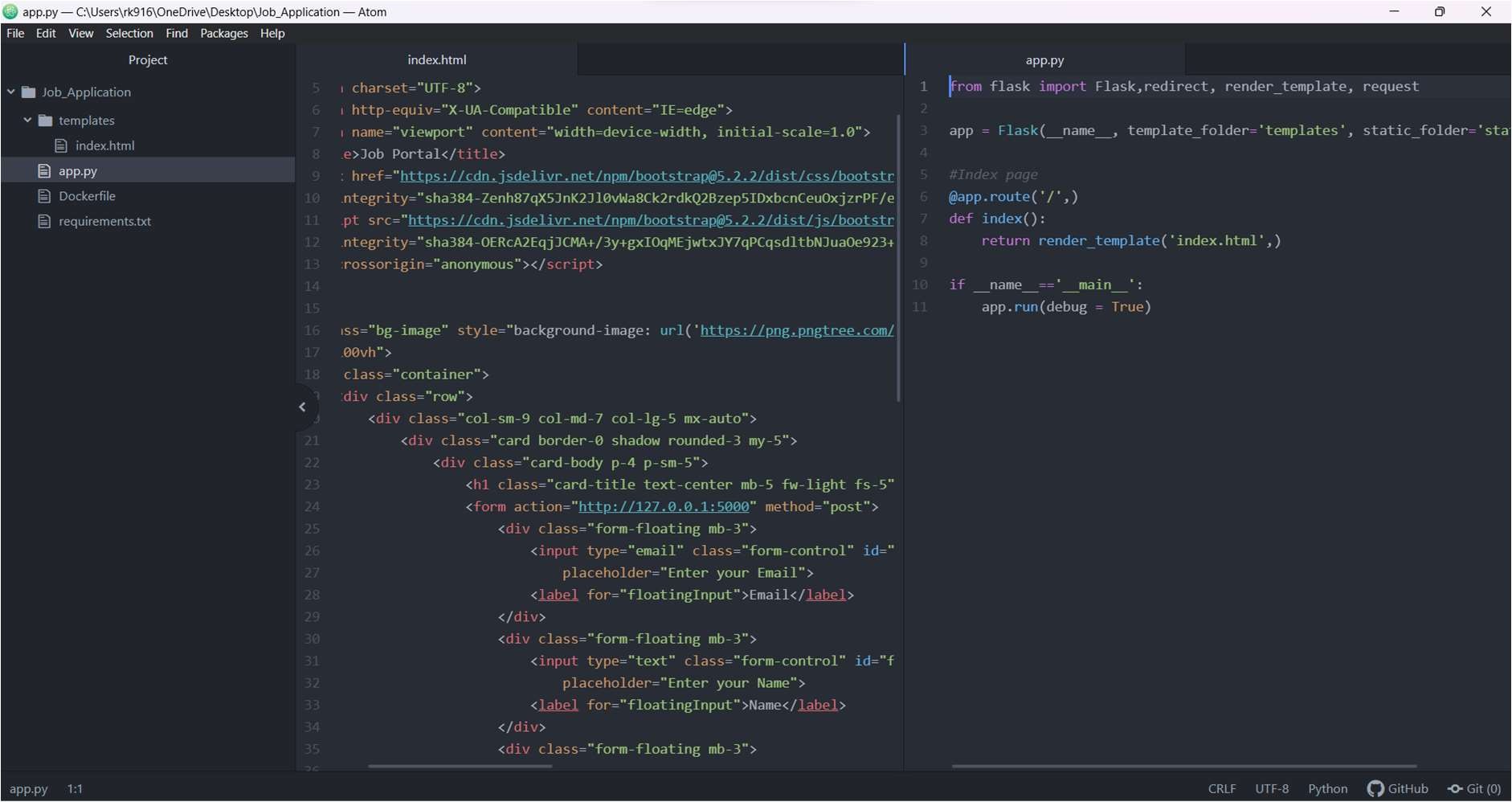
- name: repo2

image: docker.io/rajeshkumar2002/jobportalassignment4 ports:

- containerPort: 5000 protocol: TCP



# Code:



**Output:**

