Name	Rohit A S
Roll No	SSNCE195001308
Date	22 October 2022
Team ID	P2022 MID 53149
Project Name	Project - Personal Expense Tracker

Assignment - 4
Kubernetes and Docker

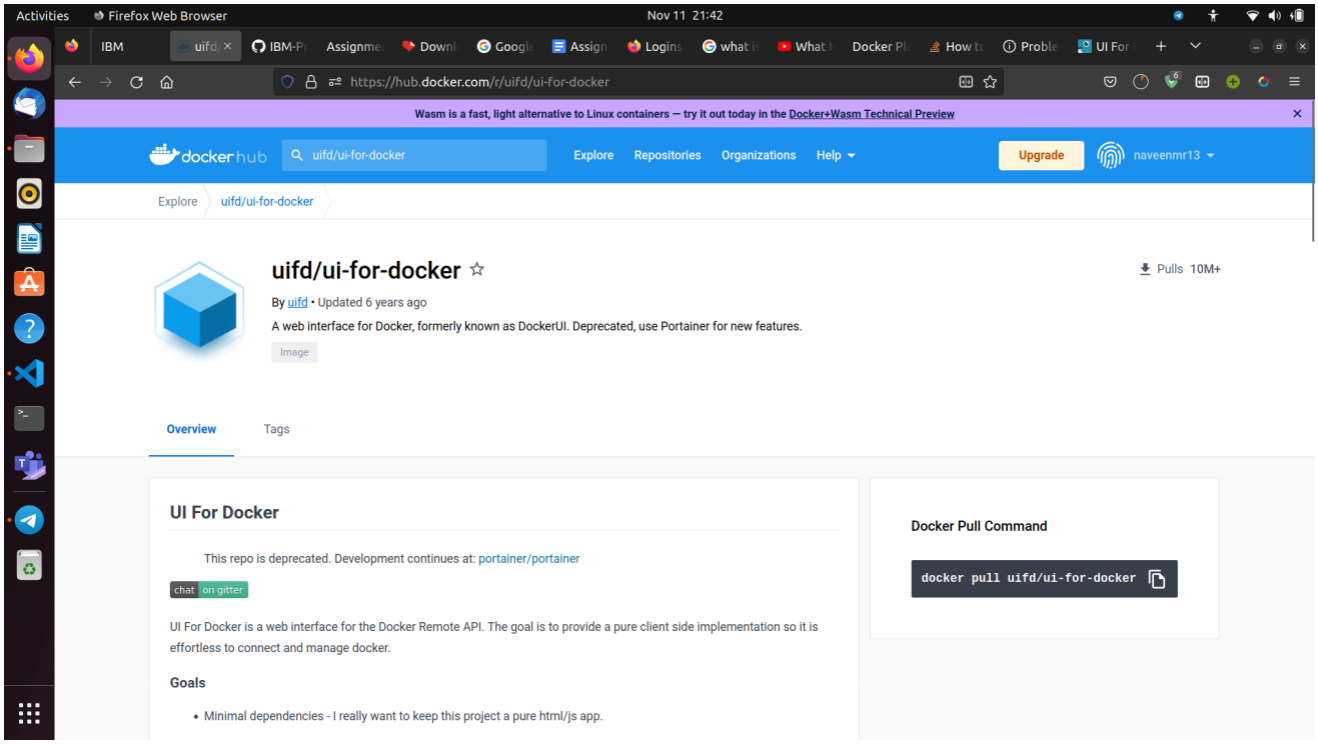
Question

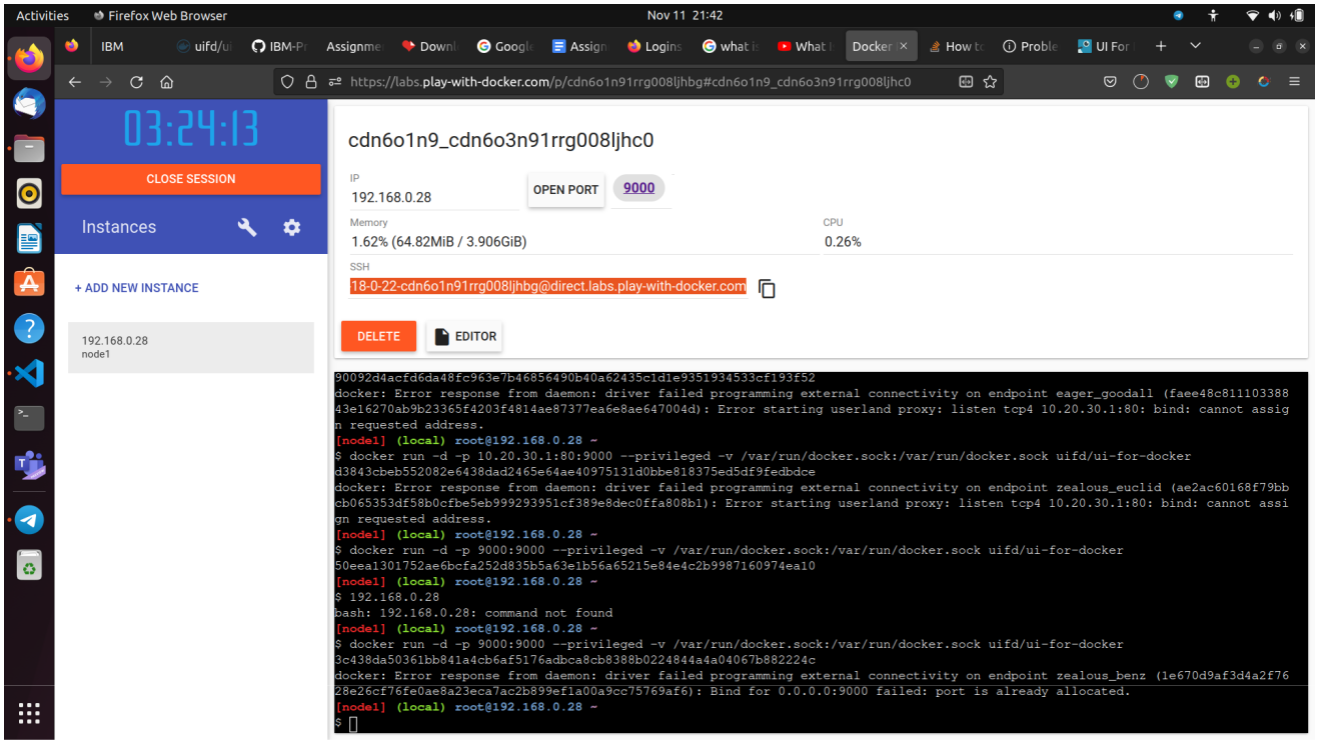
1. Pull an Image from docker hub and run it in Docker Playground
2. Create a docker file for the job portal application and deploy it in Docker desktop application
3. Create a IBM container registry and deploy hello world app or job portal app
4. Create a Kubernetes cluster in IBM cloud and deploy hello world image or job portal image and also expose the same app to run in nodeport

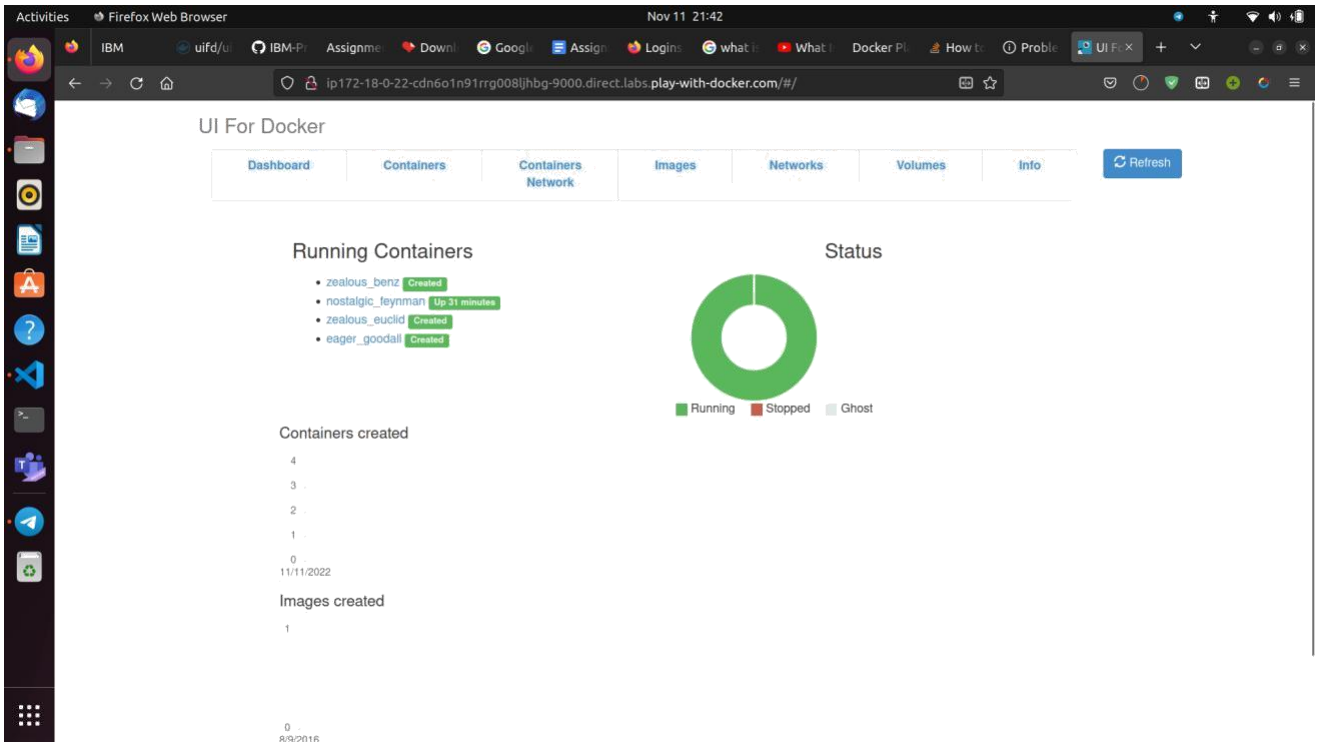
Solutions

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

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


Activities
Firefox Web Browser
Nov 11 21:42
https://hub.docker.com/r/uifd/ui-for-docker
Wasm is a fast, light alternative to Linux containers — try it out today in the Docker+Wasm Technical Preview
docker hub
uifd/ui-for-docker
Explore
Repositories
Organizations
Help
Upgrade
naveenmr13
Explore
uifd/ui-for-docker
uifd/ui-for-docker
By uifd • Updated 6 years ago
A web interface for Docker, formerly known as DockerUI. Deprecated, use Portainer for new features.
Image
Pulls 10M+
Overview
Tags
UI For Docker
This repo is deprecated. Development continues at: portainer/portainer
chat on gitter
UI For Docker is a web interface for the Docker Remote API. The goal is to provide a pure client side implementation so it is effortless to connect and manage docker.
Goals
Minimal dependencies - I really want to keep this project a pure html/js app.
Docker Pull Command
docker pull uifd/ui-for-docker


Activities
Firefox Web Browser
Nov 11 21:42
https://labs.play-with-docker.com/p/cdn6o1n91rrg008ljhbg#cdn6o1n9_cdn6o3n91rrg008ljhc0
03:24:13
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.28
node1
cdn6o1n9_cdn6o3n91rrg008ljhc0
IP
192.168.0.28
OPEN PORT
9000
Memory
1.62% (64.82MiB / 3.906GiB)
CPU
0.26%
SSH
ip172-18-0-22-cdn6o1n91rrg008ljhbg@direct.labs.play-with-docker.com
DELETE
EDITOR
90092d4acfd6da48fc963e7b46856490b40a62435c1d1e9351934533cf193f52
docker: Error response from daemon: driver failed programming external connectivity on endpoint eager_goodall (faee48c811103388
43e16270ab9b23365f4203f4814ae87377ea6e8ae647004d): Error starting userland proxy: listen tcp4 10.20.30.1:80: bind: cannot assig
n requested address.
[node1] (local) root@192.168.0.28 ~
$ docker run -d -p 10.20.30.1:80:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
d3843cbeb552082e6438dad2465e64ae40975131d0bbe818375ed5df9fedbdce
docker: Error response from daemon: driver failed programming external connectivity on endpoint zealous_euclid (ae2ac60168f79bb
cb065353df58b0cfbe5eb999293951cf389e8dec0ffa808b1): Error starting userland proxy: listen tcp4 10.20.30.1:80: bind: cannot assi
gn requested address.
[node1] (local) root@192.168.0.28 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
50eea1301752ae6bcfa252d835b5a63e1b56a65215e84e4c2b998716097 4ea10
[node1] (local) root@192.168.0.28 ~
$ 192.168.0.28
bash: 192.168.0.28: command not found
[node1] (local) root@192.168.0.28 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
3c438da50361bb841a4cb6af5176adbca8cb8388b0224844a4a04067b882224c
docker: Error response from daemon: driver failed programming external connectivity on endpoint zealous_benz (1e670d9af3d4a2f76
28e26cf76fe0ae8a23eca7ac2b899ef1a00a9cc75769af6): Bind for 0.0.0.0:9000 failed: port is already allocated.
[node1] (local) root@192.168.0.28 ~
$


Activities
Firefox Web Browser
Nov 11 21:42
IBM
ip172-18-0-22-cdn6o1n91rrg008ljhbg-9000.direct.labs.play-with-docker.com/#/
UI For Docker
Dashboard
Containers
Containers Network
Images
Networks
Volumes
Info
Refresh
Running Containers
zealous_benz Created
nostalgic_feynman Up 31 minutes
zealous_euclid Created
eager_goodall Created
Status
Running
Stopped
Ghost
Containers created
4
3
2
1
0
11/11/2022
Images created
1
0
8/9/2016

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

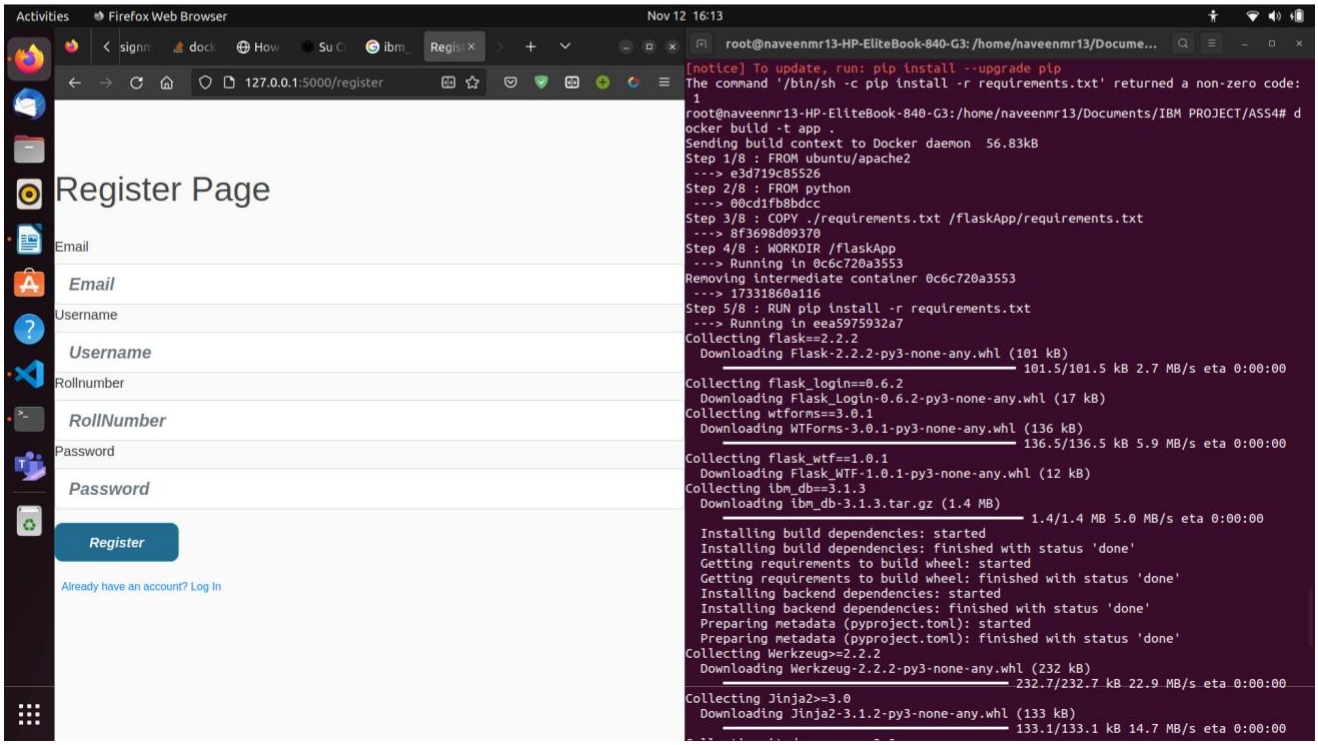
a. Create a docker file for build and deploy flask app.
b. Use **`docker build -t image_name .`** in the current directory to start building the docker image and deploy in our local docker
c. Use **`docker run -p 5000:5000 image_name`** to run in local system

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”]
```

Run locally using docker

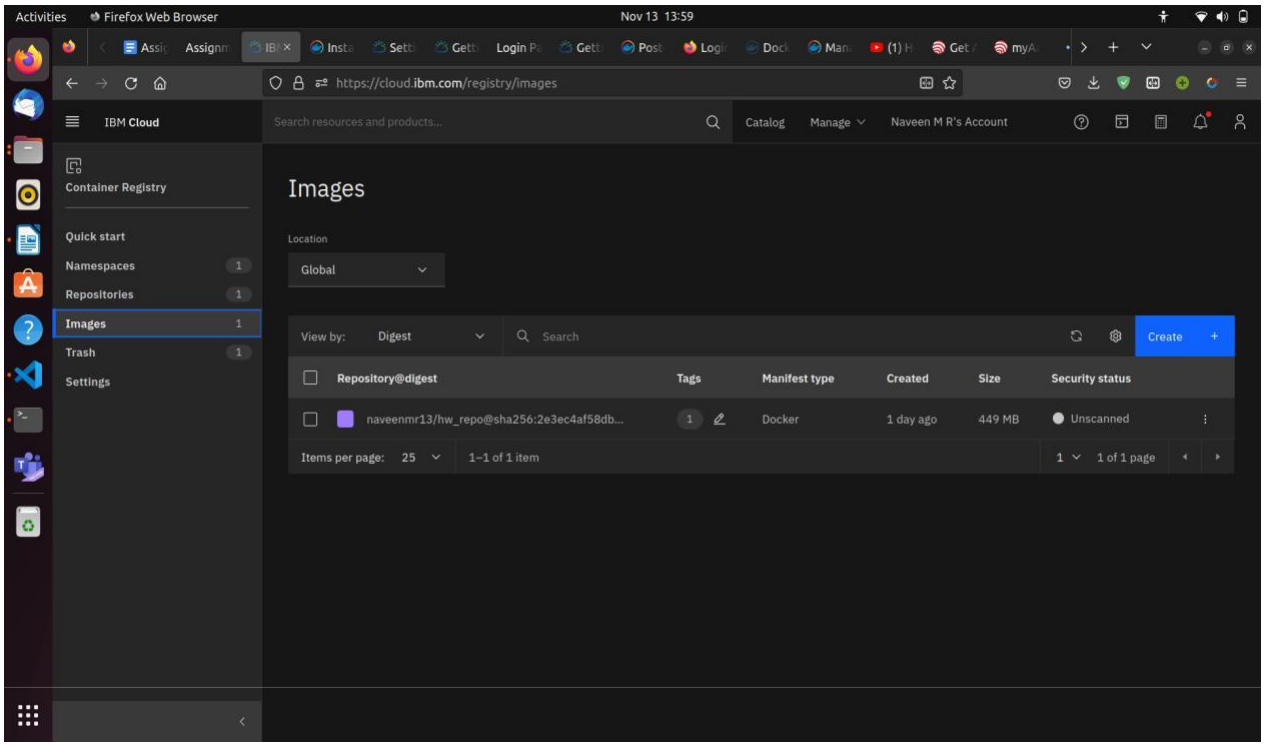
```
root@naveenmr13-HP-EliteBook-840-G3:/home/naveenmr13/Documents/IBM PROJECT/ASS4# d
ocker run -p 5000:5000 app
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. U
se a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 107-635-278
172.17.0.1 - - [12/Nov/2022 10:42:44] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [12/Nov/2022 10:42:44] "GET /static/style.css HTTP/1.1" 304 -
172.17.0.1 - - [12/Nov/2022 10:43:34] "GET /register HTTP/1.1" 200 -
172.17.0.1 - - [12/Nov/2022 10:43:34] "GET /static/style.css HTTP/1.1" 304 -
```



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

a. Log into IBM cloud
b. Create a **container registry**
c. Using IBM Cloud CLI, install the **container registry plugin** in our system
d. Push our docker image into the created container registry using **docker push**
e. So, our job portal app is deployed in the IBM container registry

f.



4. Create a Kubernetes cluster in IBM cloud and deploy helloworld image or job portal image and also expose the same app to run in nodeport

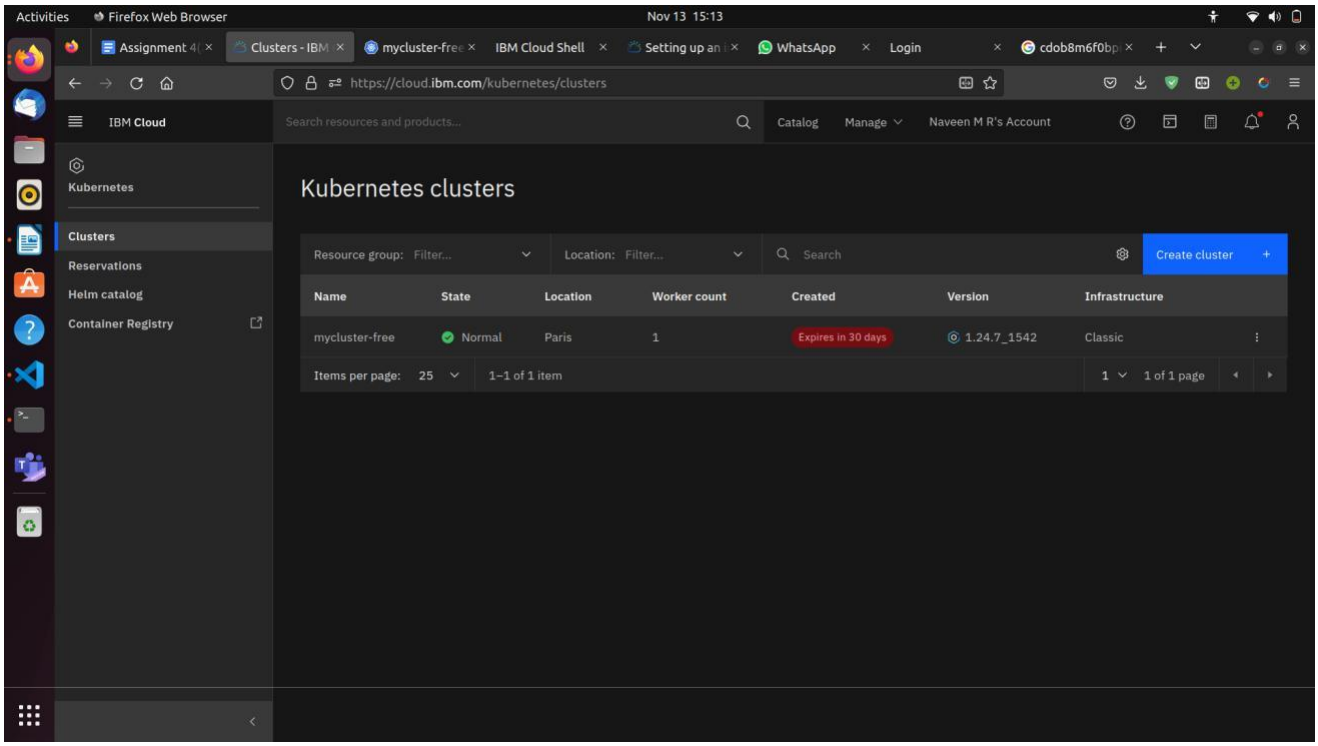
a. Log into IBM cloud
b. Create a **kubernete**
c. Using IBM Cloud CLI, install the **ks plugin** in our system
d. Create **a cluster** in the kubernetes
e. Now, go to the **kubernetes dashboard** where we need to create a service based on a yml file (given below)
f. In that file, we have to mention *which image we are going to use* and the *app name*
g. Take the **public IP address** and **Nodeport** since we exposed the *flask app in nodeport*
h. Finally, we got the **url address** where our flask app is hosted

job-portal-app.yml

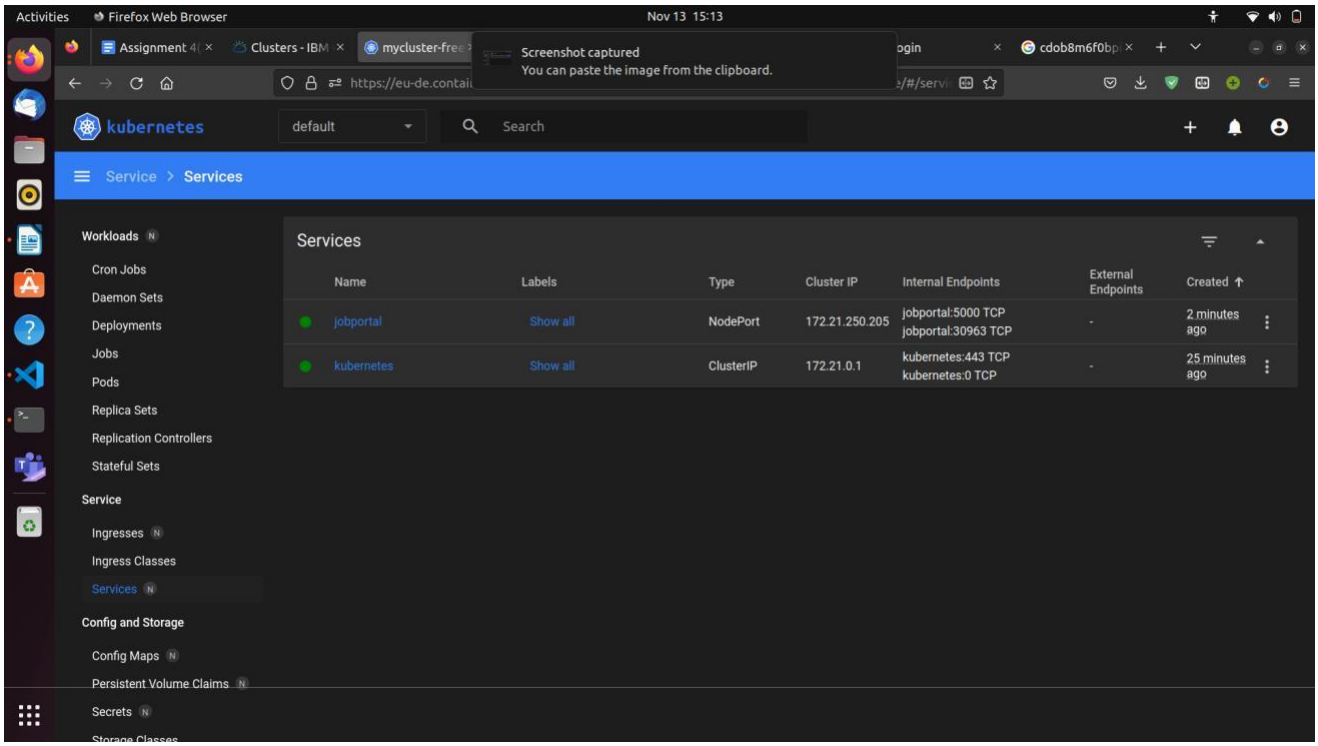
```
apiVersion:
v1 kind:
Service
metadata:
  name:
job-portal-app spec:
  selector:
    app:
  job-portal-app
  ports:
  - port: 5000
  type:
  NodePort
---
apiVersion:
apps/v1 kind:
Deployment
metadata:
  name:
  job-portal-app
  labels:
    app:
job-portal-app spec:
  selector:
    matchLabels
    :
      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"
```

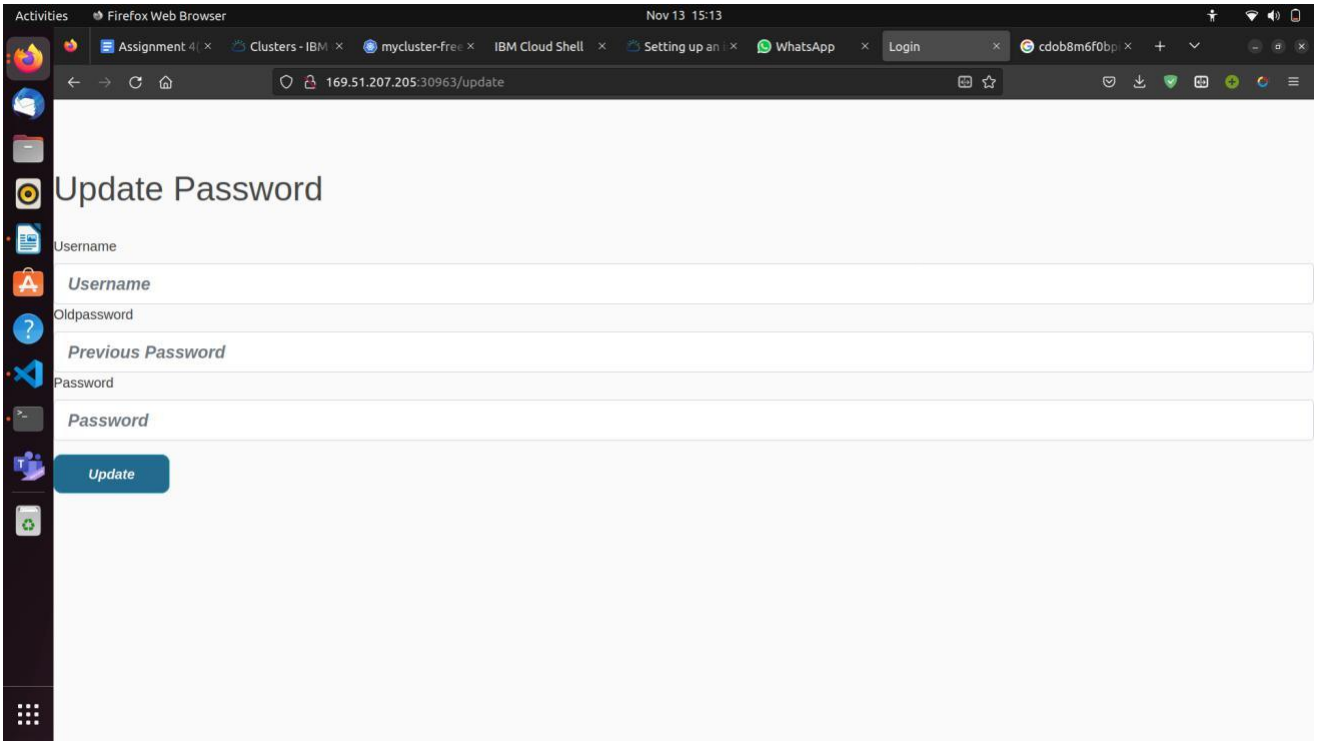
Cluster creation

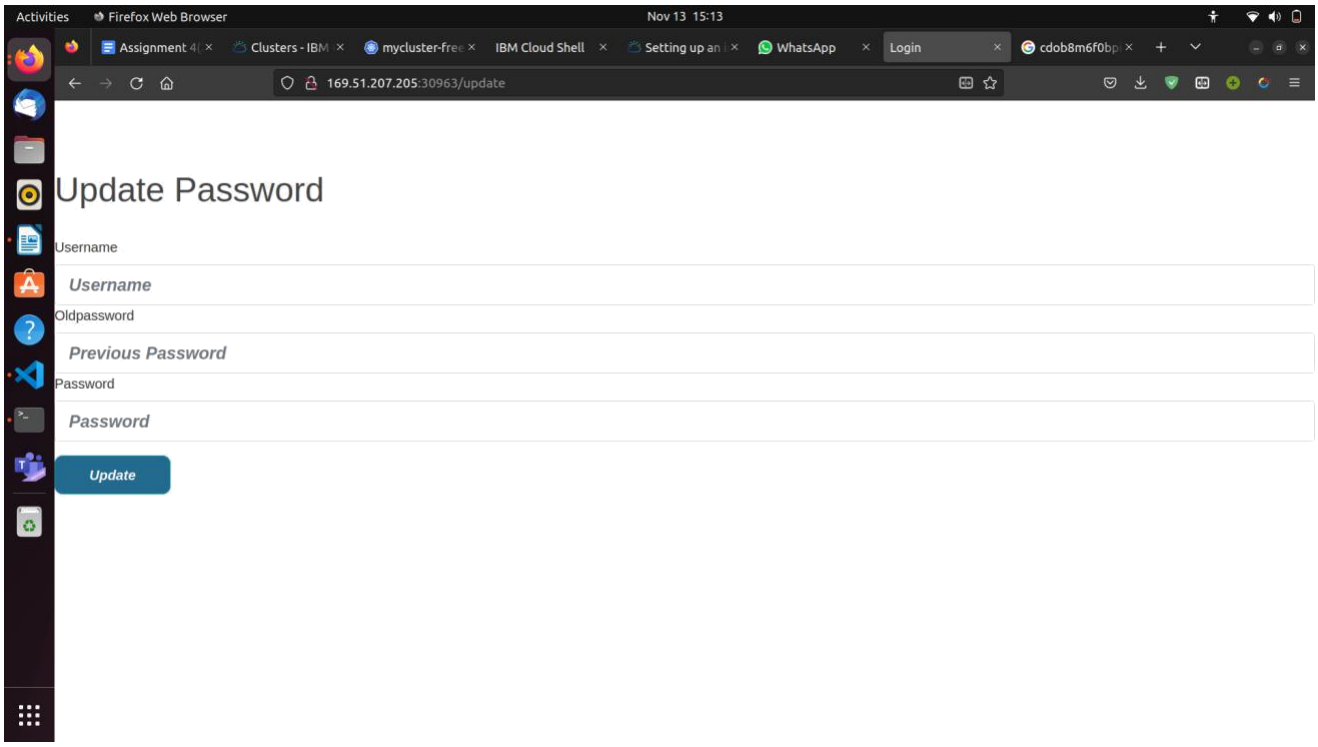


Configuring the cluster



Run our flask app in the IBM kubernetes




Activities
Firefox Web Browser
Nov 13 15:13
Assignment
Clusters - IBM
mycluster-free
IBM Cloud Shell
Setting up an
WhatsApp
Login
cdob8m6f0bp
169.51.207.205:30963/update
Update Password
Username
Username
Oldpassword
Previous Password
Password
Password
Update