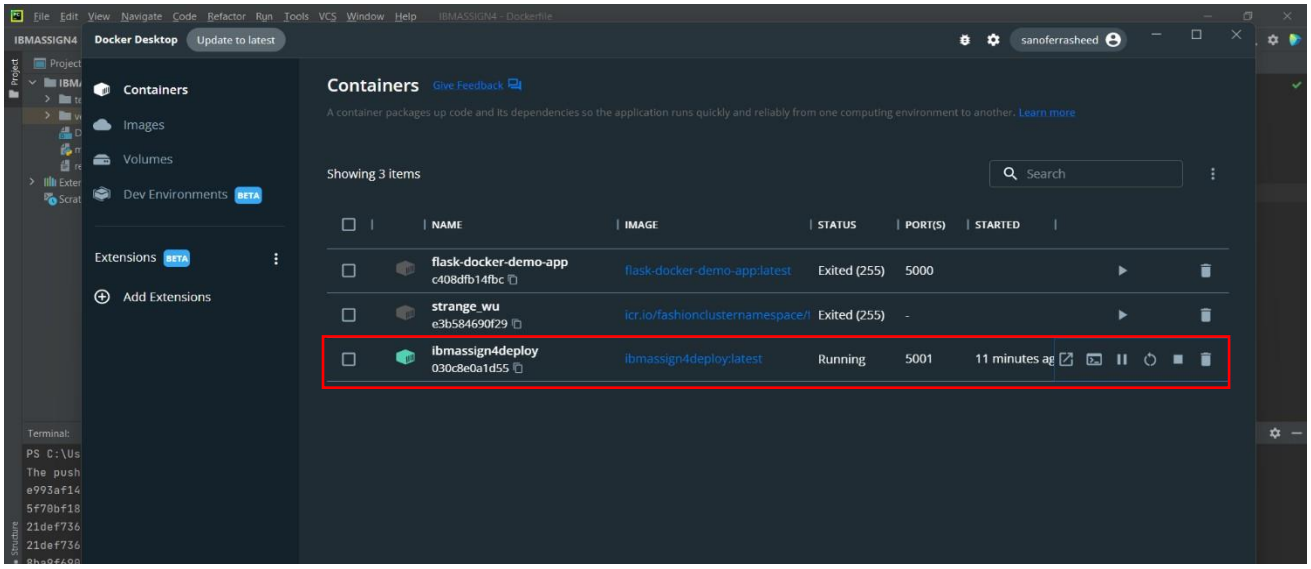
ASSIGNMENT 4

CLOUD APPLICATION DEVELOPMENT

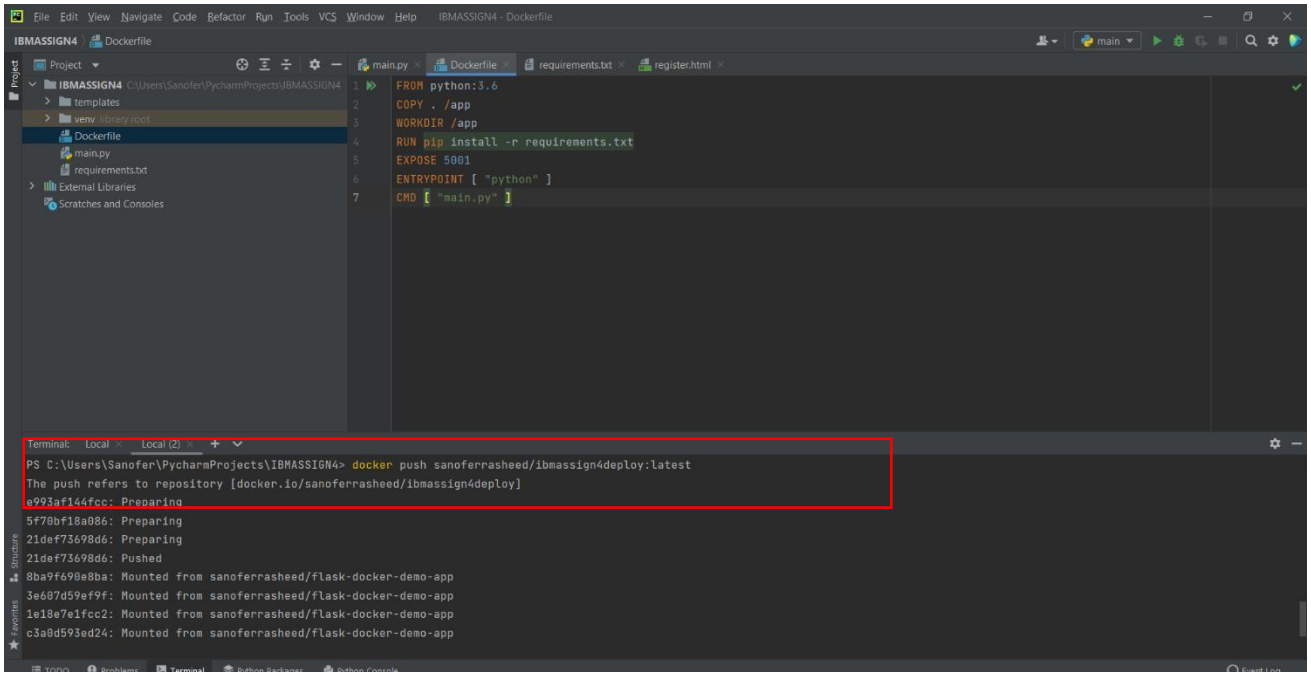
Team Id	PNT2022TMID09989
Project Name	Smart Fashion Recommender Application
Maximum Marks	2 Marks

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

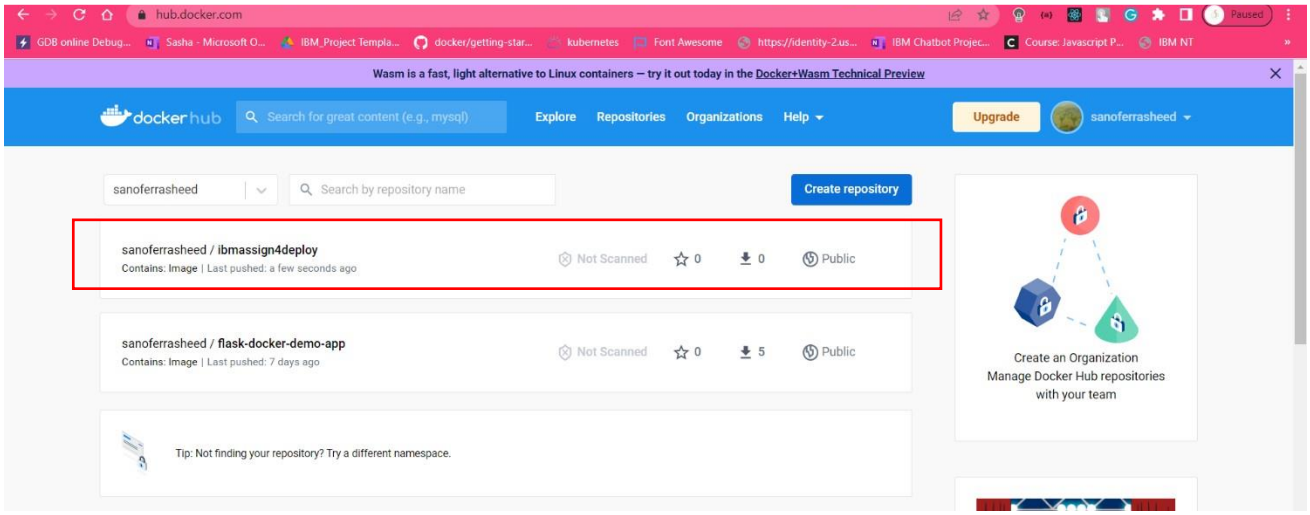
The image is built.



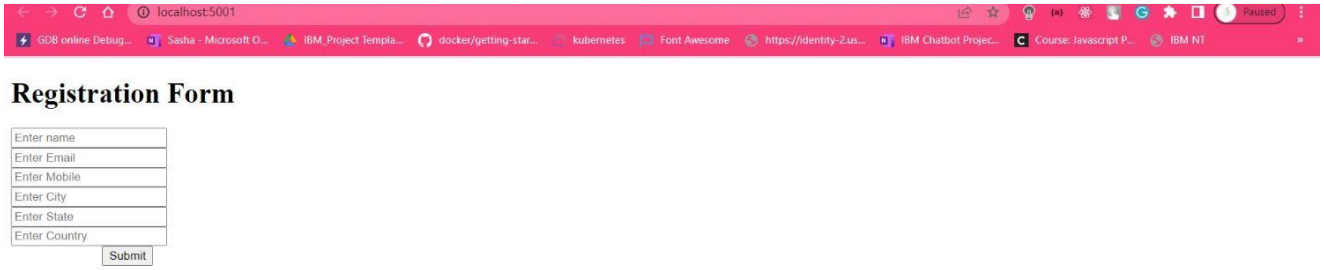
The same image is pushed to docker hub using the command



Here image name is ibmassign4deploy. Thus it is pushed in docker hub.



The app is running at the specified port.

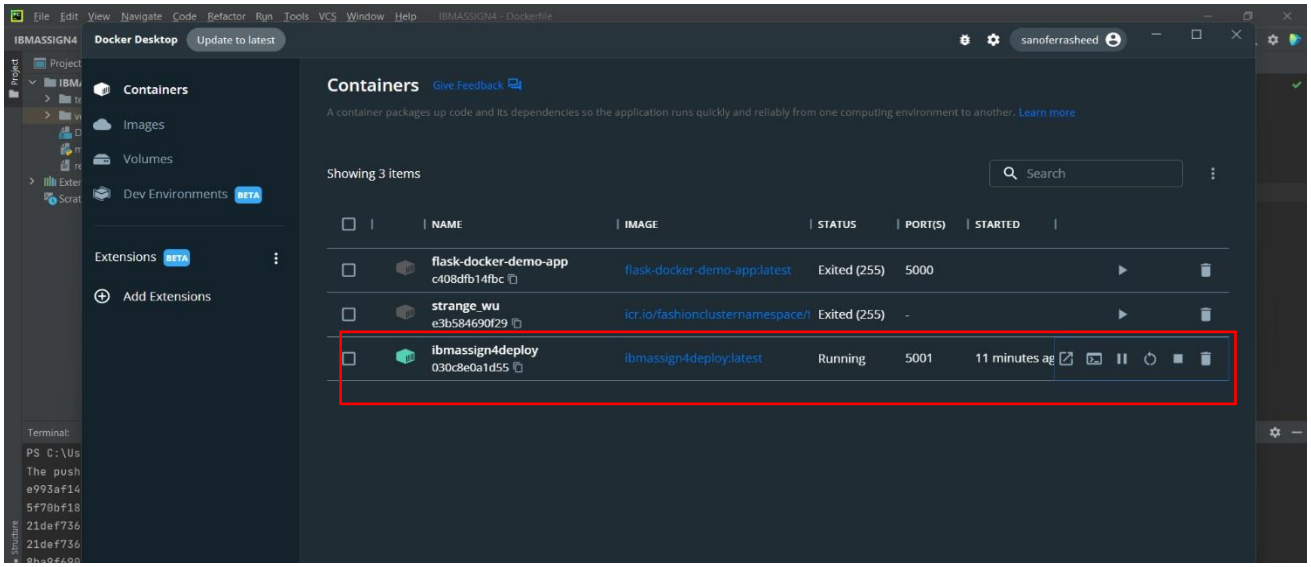


2. Create a docker file for the application and deploy it in Docker desktop application.

Dockerfile:

```
FROM python:3.6
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
EXPOSE 5001
ENTRYPOINT [ "python" ]
CMD [ "main.py" ]
```

Thus docker file created and deployed in docker desktop.

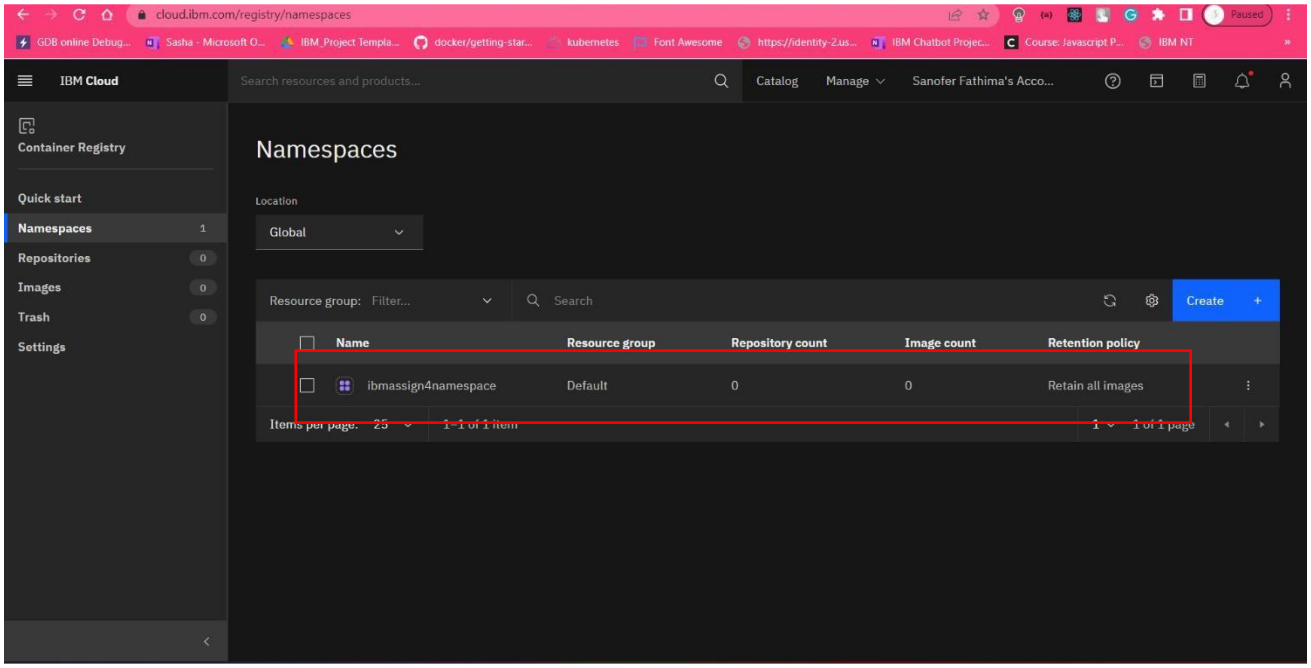


3. Create a IBM container registry and deploy hello world app.

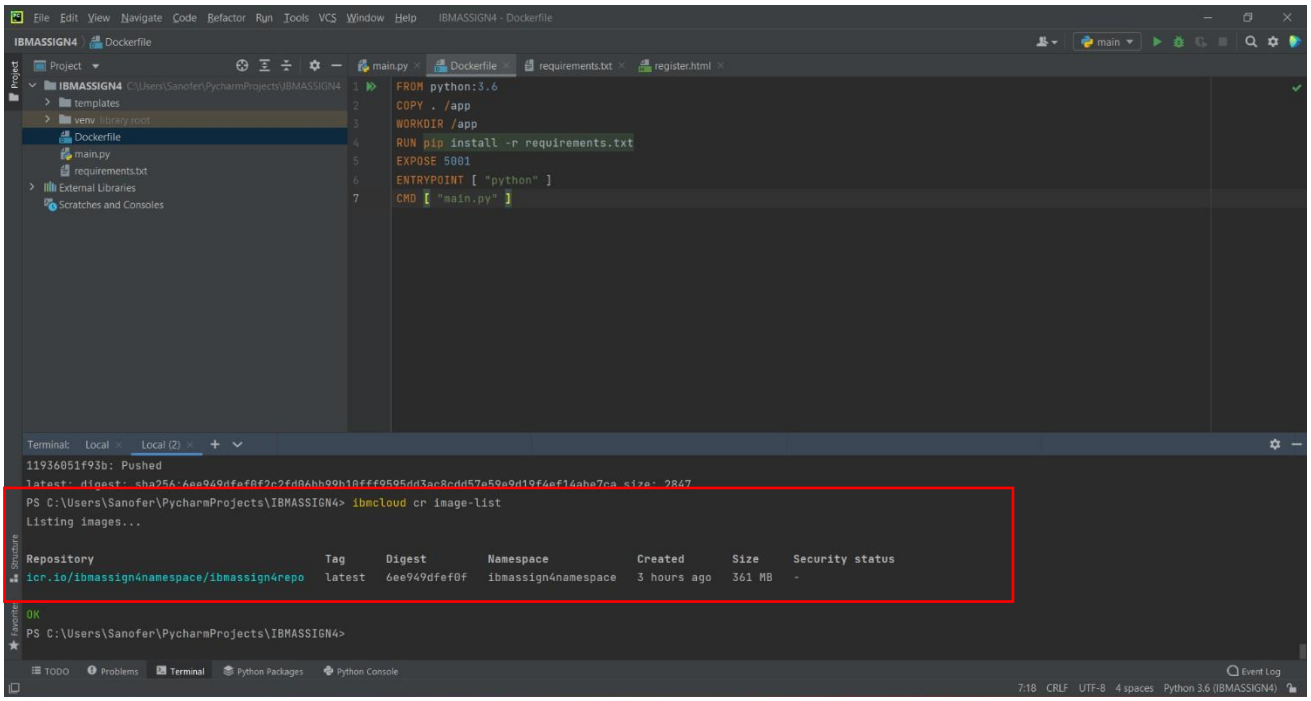
Container registry created using

> docker tag sanoferrasheed/ibmassign4deploy:latest icr.io/ibmassign4namespace/ibmassign4repo:latest

> docker push icr.io/ibmassign4namespace/ibmassign4repo:latest

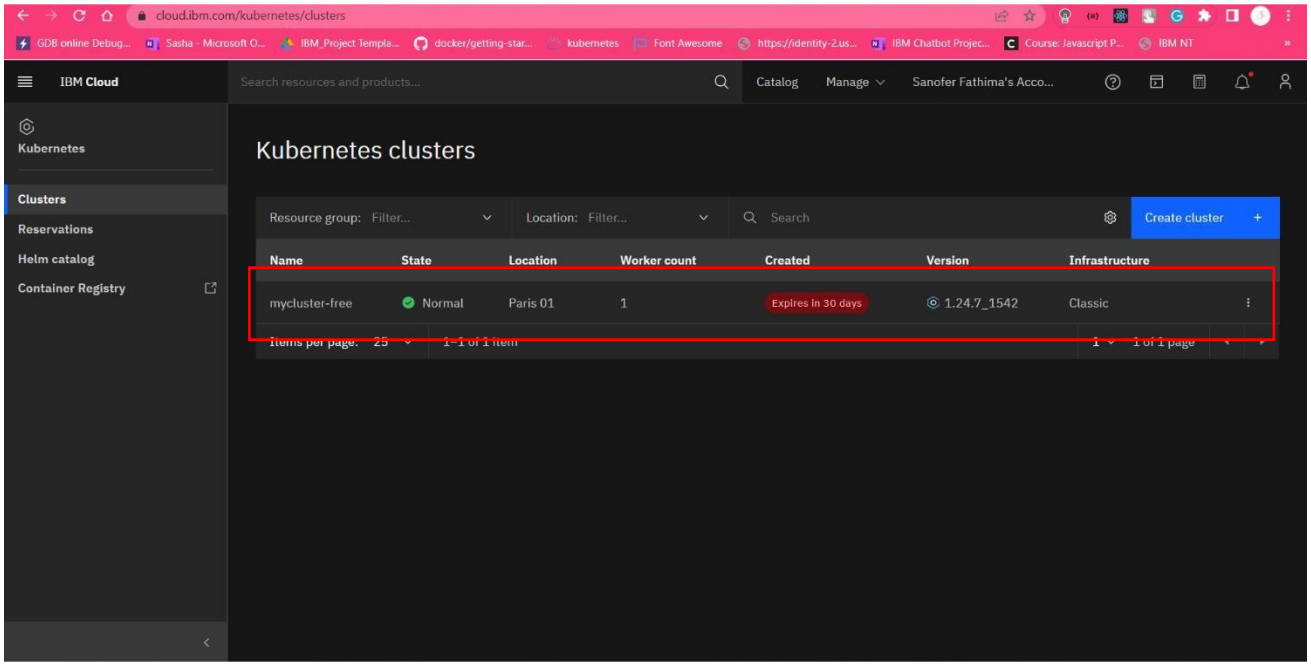


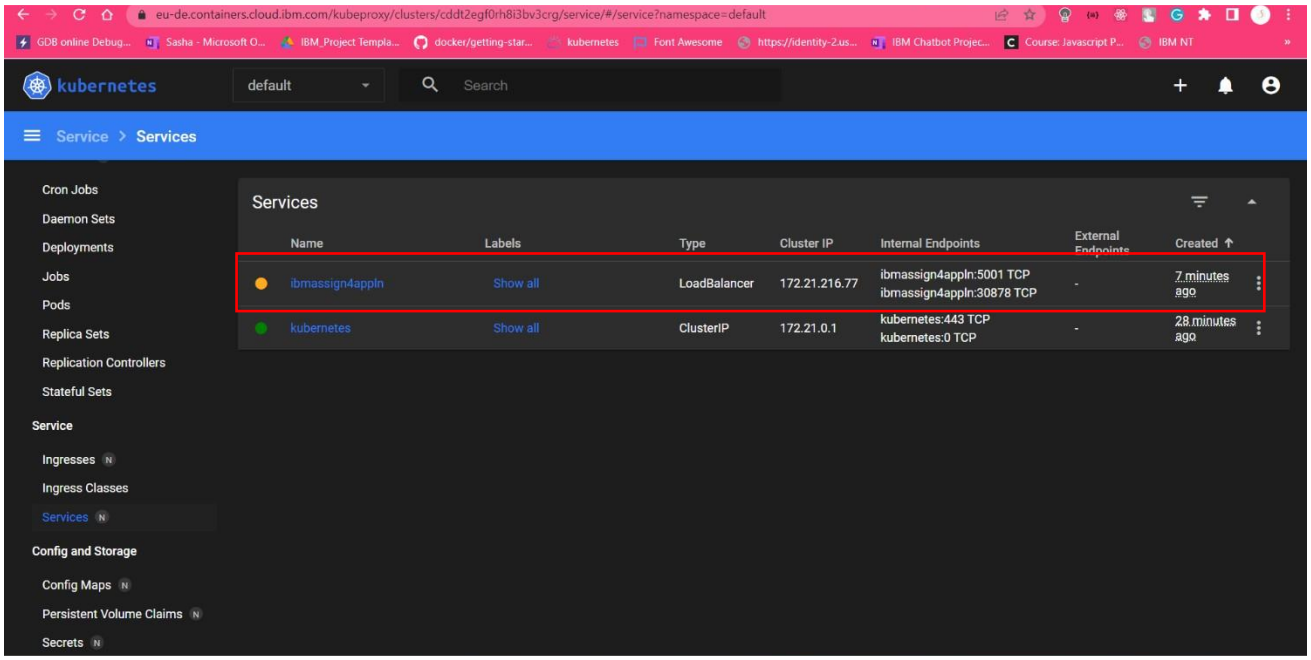
Thus, images in container registry are listed



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.

Thus, cluster is created.




eu-de.containers.cloud.ibm.com/kubeproxy/clusters/cddt2egf0rh8i3bv3crg/service/#/service?namespace=default
kubernetes
default
Search
Service > Services
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Ingress Classes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Services
Name
Labels
Type
Cluster IP
Internal Endpoints
External Endpoints
Created
ibmassign4appln
Show all
LoadBalancer
172.21.216.77
ibmassign4appln:5001 TCP
ibmassign4appln:30878 TCP
7 minutes ago
kubernetes
Show all
ClusterIP
172.21.0.1
kubernetes:443 TCP
kubernetes:0 TCP
28 minutes ago