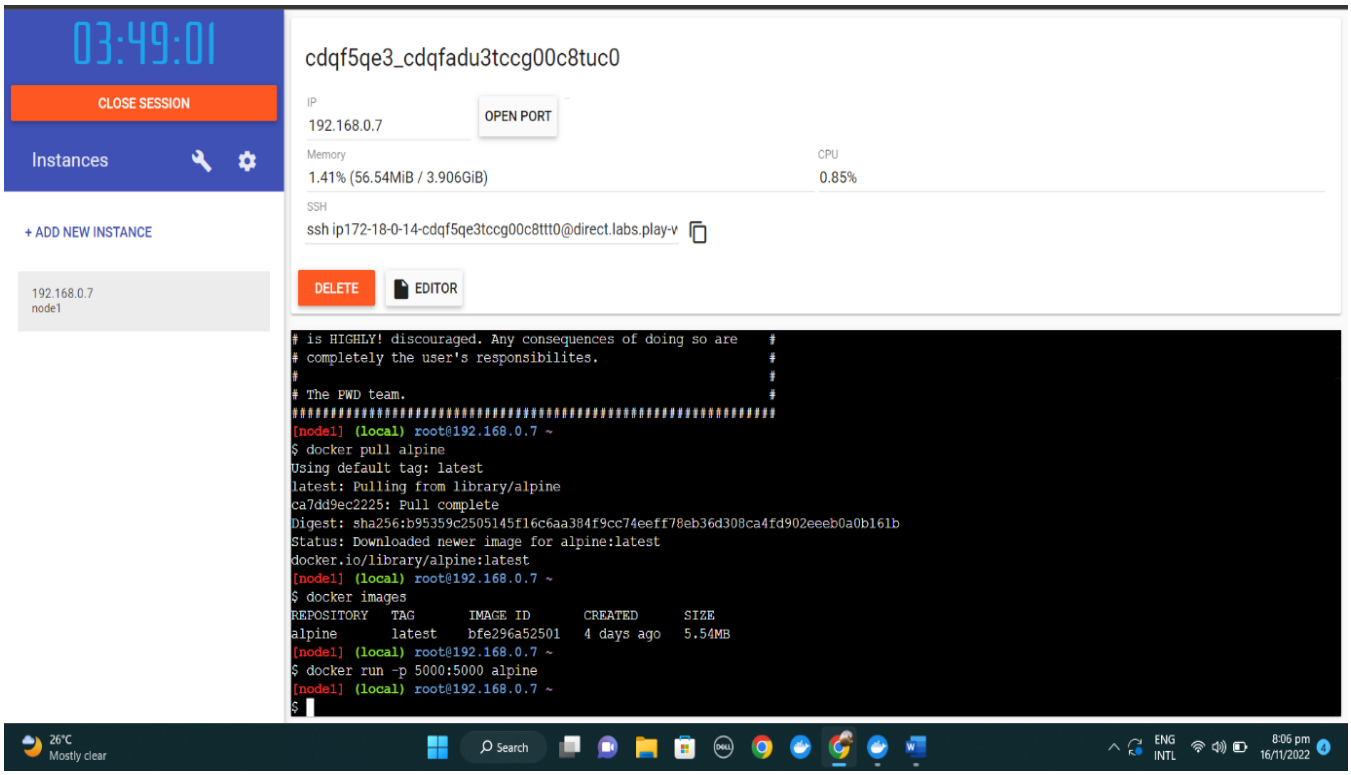
# ASSIGNMENT-4

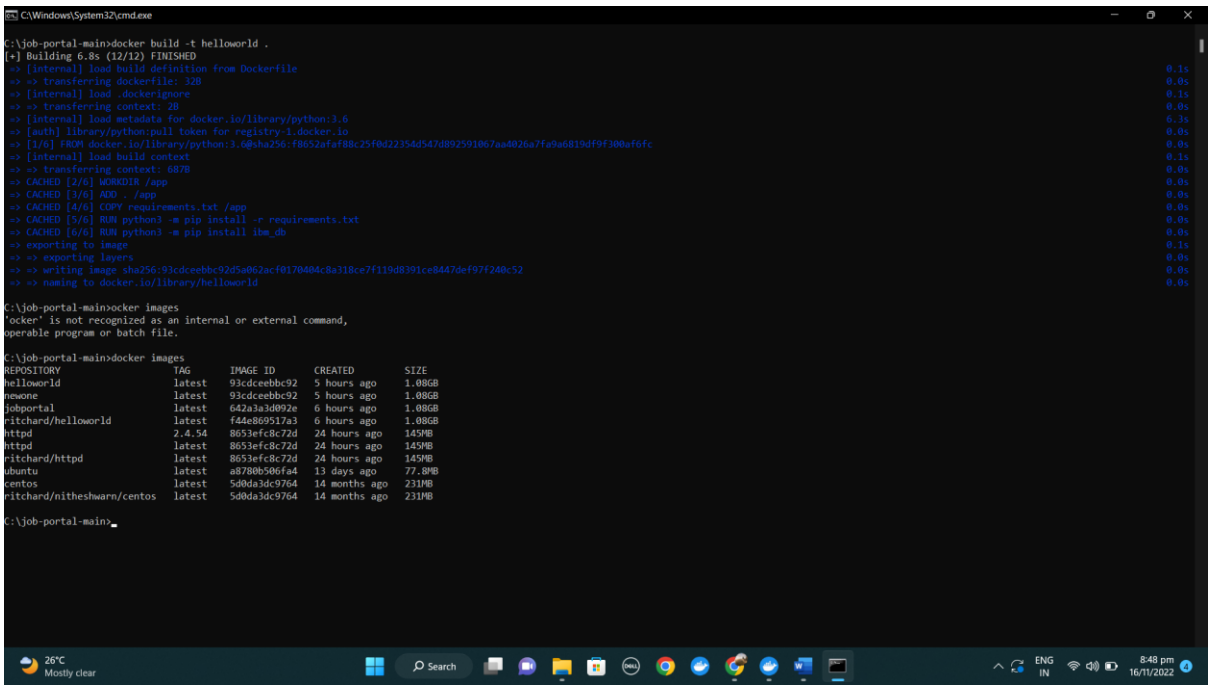
| | |
|---|---|
| TEAM ID | PNT2022TMID48539 |
| DATE | 15-11-22 |
| PROJECT NAME | Customer care registry |

## 1.PULL AN IMAGE FROM DOCKER HUB AND RUN IT IN DOCKER PLAYGROUND



## 2.CREATE A DOCKER FILE FOR THE JOBPORTAL APPLICATION AND DEPLOY IT IN DOCKER DESKTOP APPLICATION.

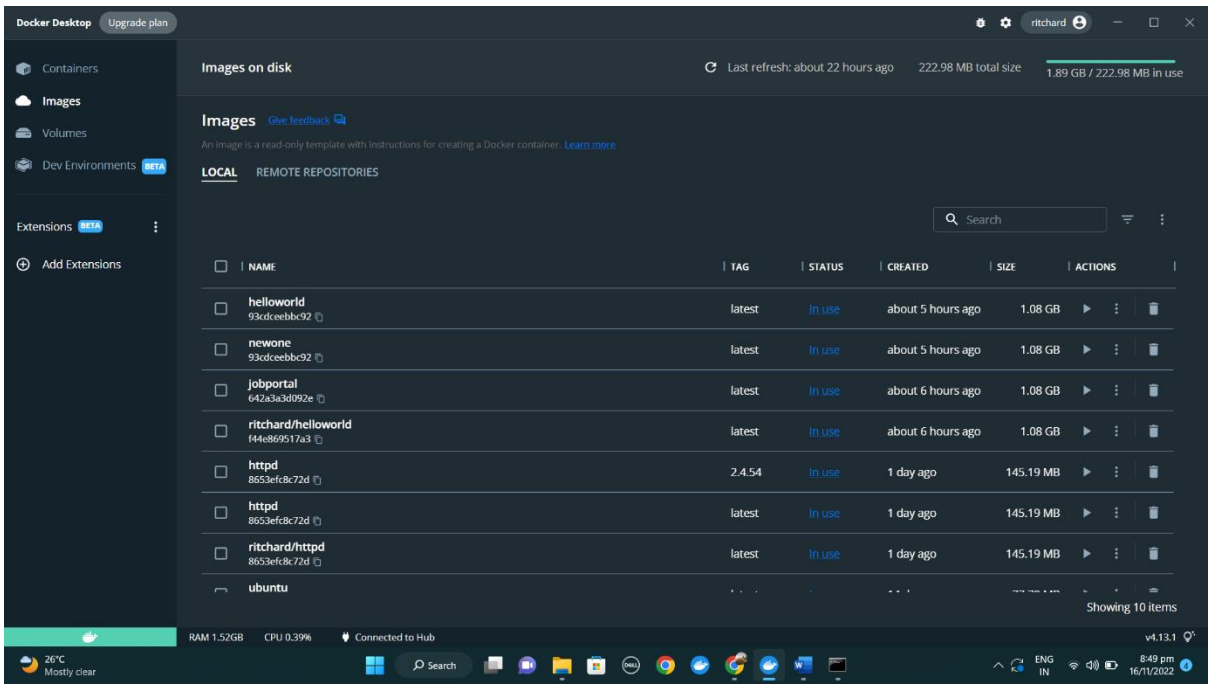
### Building the image hello world:




### Run:

```
C:\Users\nithe\AppData\Local\Programs\Python\Python310\job-portal>docker run -p 8080:8080 helloworld
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
```

## Docker desktop:

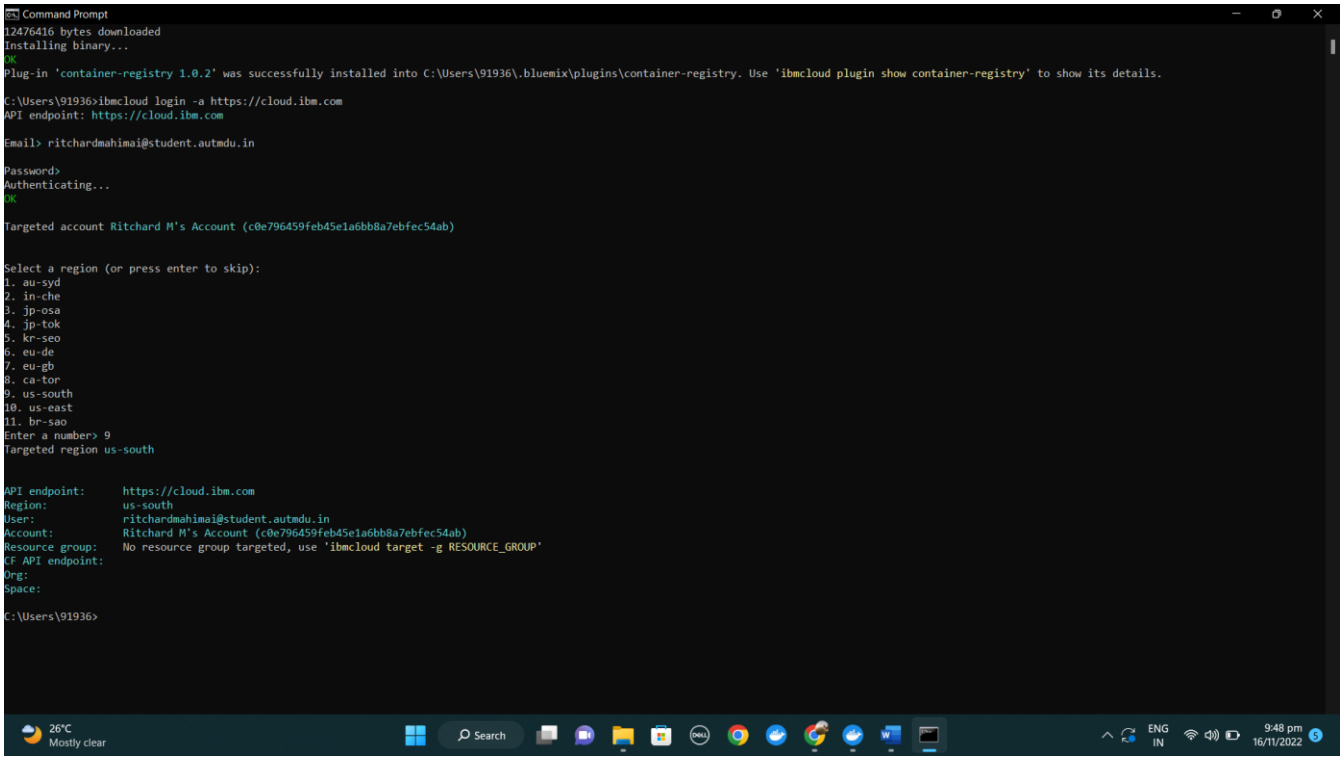



## 3.CREATE A IBM CONTAINER REGISTRY AND DEPLOY HELLO WORLD APP.

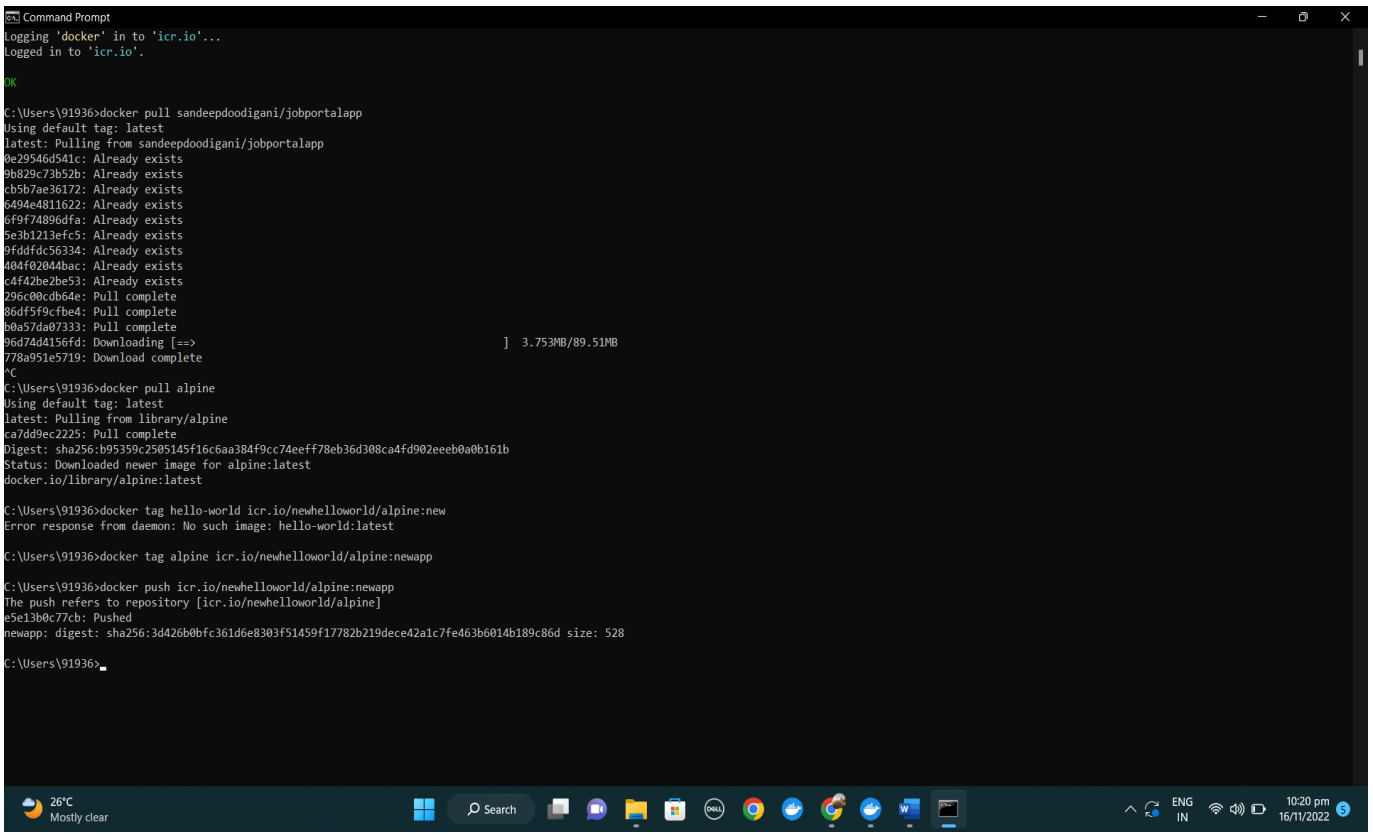
### Install Container registry and create namespace

```
C:\Users\91936>ibmcloud cr region-set global
The region is set to 'global', the registry is 'icr.io'.

OK

C:\Users\91936>ibmcloud cr namespace-add newhelloworld
No resource group is targeted. Therefore, the default resource group for the account ('Default') is targeted.

Adding namespace 'newhelloworld' in resource group 'Default' for account Ritchard M's Account in registry icr.io...

Successfully added namespace 'newhelloworld'

OK

C:\Users\91936>
```

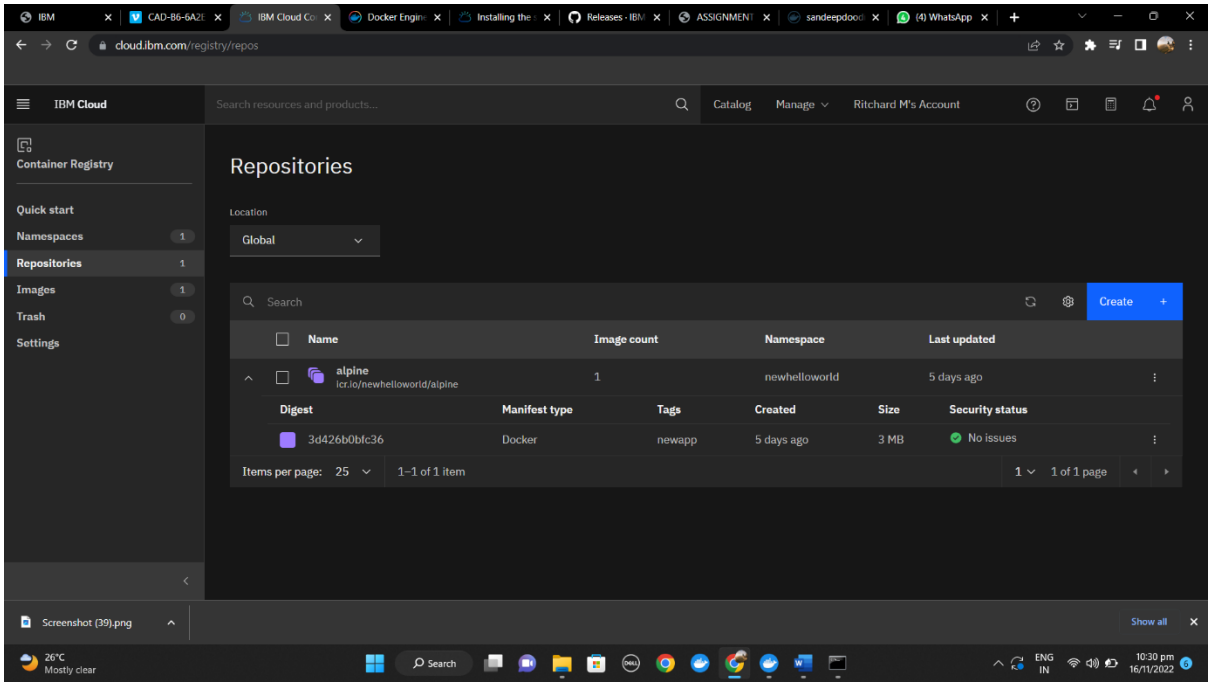
## IBM cloud login




## Pull and Push image

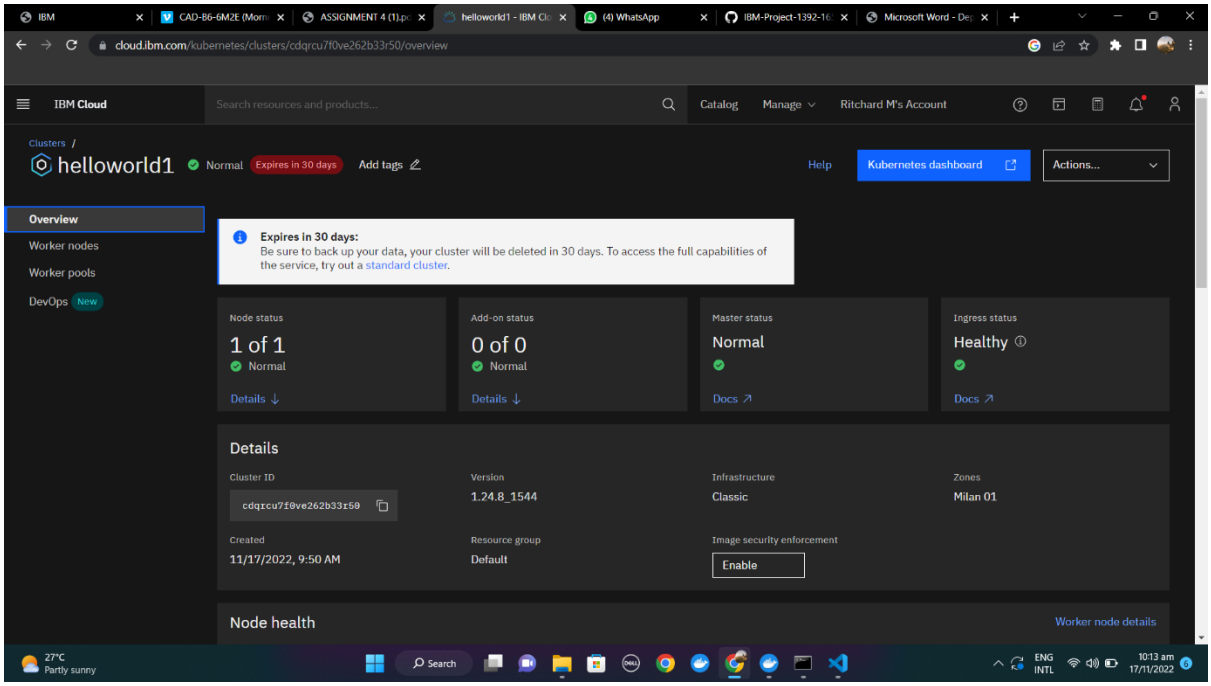


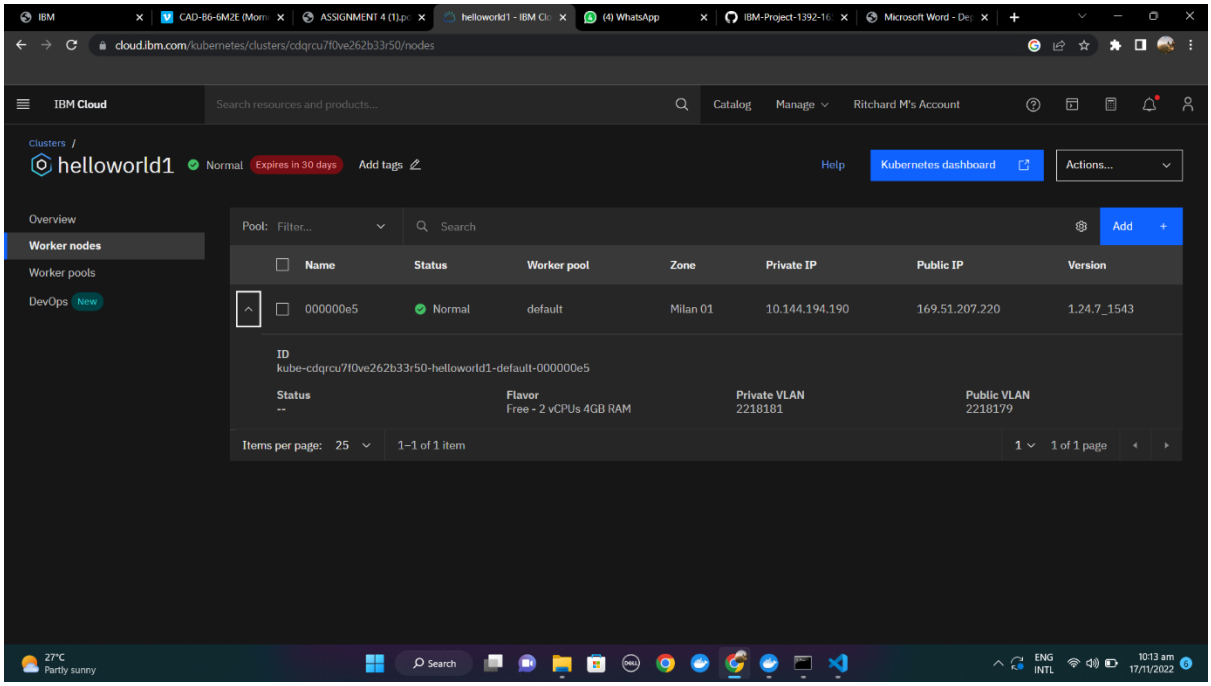
## IBM container registry




## 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

## Cluster creation

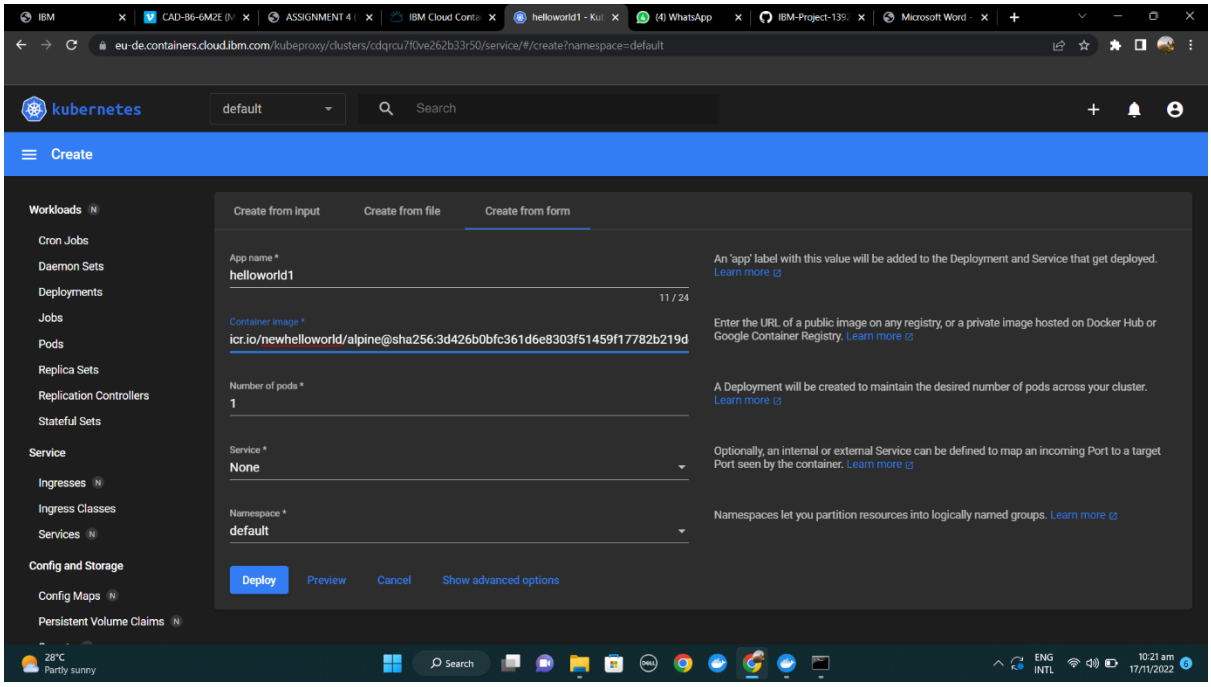
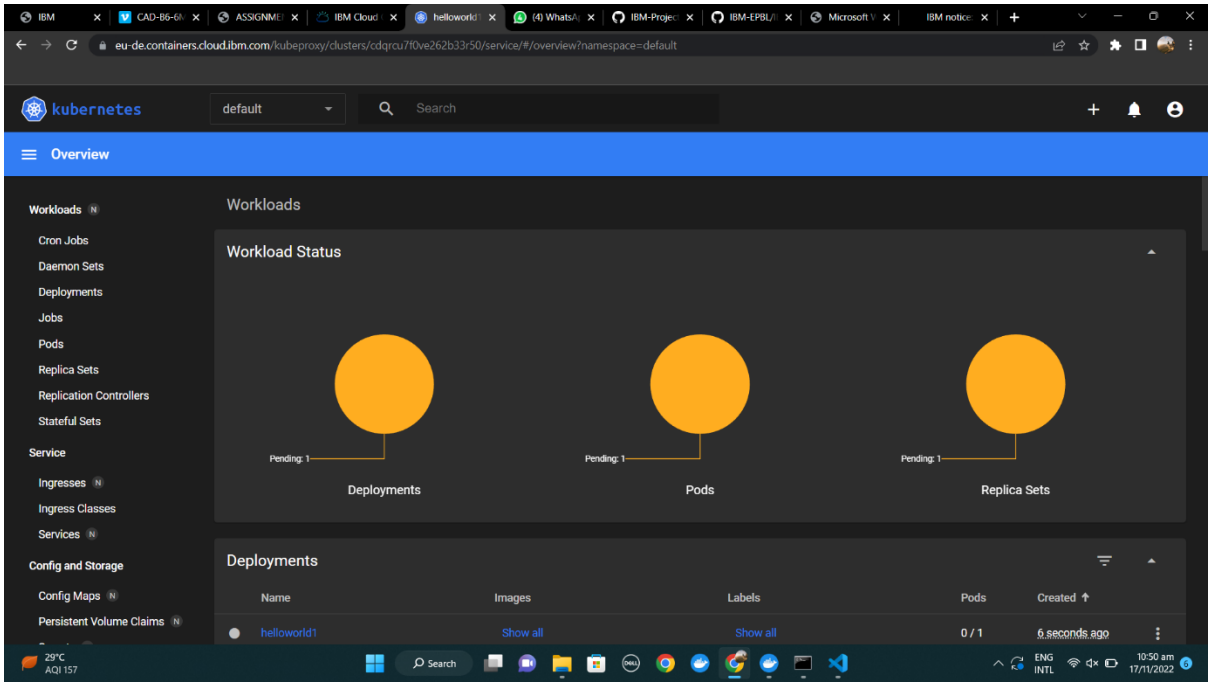





## Deployment.yaml

```yaml
apiVersion: apps/v1
 kind: Deployment
 metadata:
   name: jobportal
 spec:
   replicas: 1
   selector:
     matchLabels:
       app: flasknode
   template:
     metadata:
       labels:
         app: flasknode
     spec:
       containers:
       - name: flasknode
         image: icr.io/helloworld1/newhelloworld
         imagePullPolicy: Always
         ports:
         - containerPort: 5000
```

## Service.yaml

```yaml
 apiVersion: v1
 kind: Service
 metadata:
   name: flask-node-deployment
```

```yaml
spec:
  ports:
  - port: 5000
    targetPort: 5000
  selector:
    app: flasknode
```