ASSIGNMENT -4

JAYAPRAKSH P

JOYSON SAMUEL P

JAYA SIVAAKRISHNA B

SHAIKH FIRAAS A

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

STEP-1 : List out the< docker images>

```
C:\Users\jayap>docker images
REPOSITORY                          TAG       IMAGE ID       CREATED         SIZE
jayaprakash2019/docker101tutorial   latest    94ab66b2cb4d   3 weeks ago     28.9MB
docker/getting-started              latest    cb90f98fd791   6 months ago    28.8MB
```

STEP-2:Add the tag name for existing docker image to push the docker hub

```
C:\Users\jayap>docker tag jayaprakash2019/docker101tutorial:latest jayaprakash2019/jayaprakash2019/docker101tutorial
```

STEP-3: Push to docker hub

https://hub.docker.com/r/jayaprakash2019/docker101tutorial

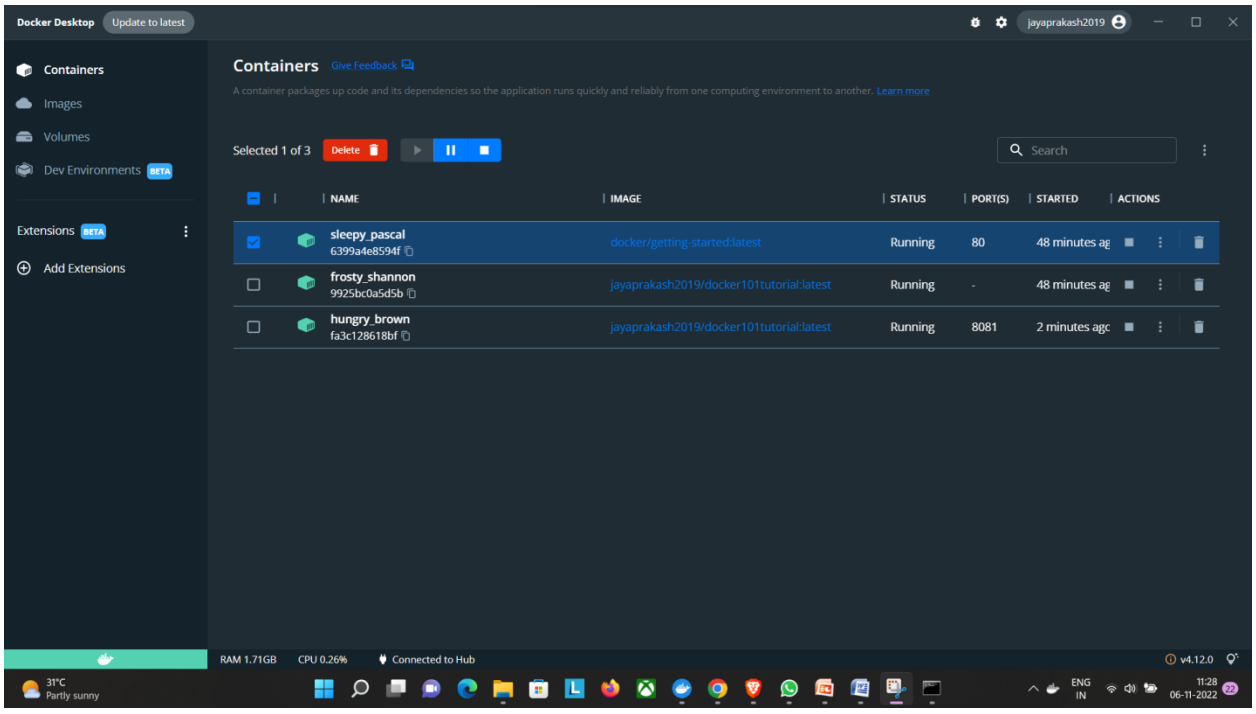
```
C:\Users\jayap>docker push jayaprakash2019/docker101tutorial
Using default tag: latest
The push refers to repository [docker.io/jayaprakash2019/docker101tutorial]
6bfa574ae9d8: Pushed
26be16e2f5ce: Pushed
00e62d52c55c: Layer already exists
403a87f26438: Layer already exists
b26a89445b38: Layer already exists
0f7dbcfcda2a: Layer already exists
492c31ac001f: Pushed
994393dc58e7: Layer already exists
latest: digest: sha256:ec5151b5ed5ba01931614d029ba15e99e319d514534fb93528b9d358b75e8b0f size: 1990
```

STEP-4: Pull the docker images from the docker hub to docker container

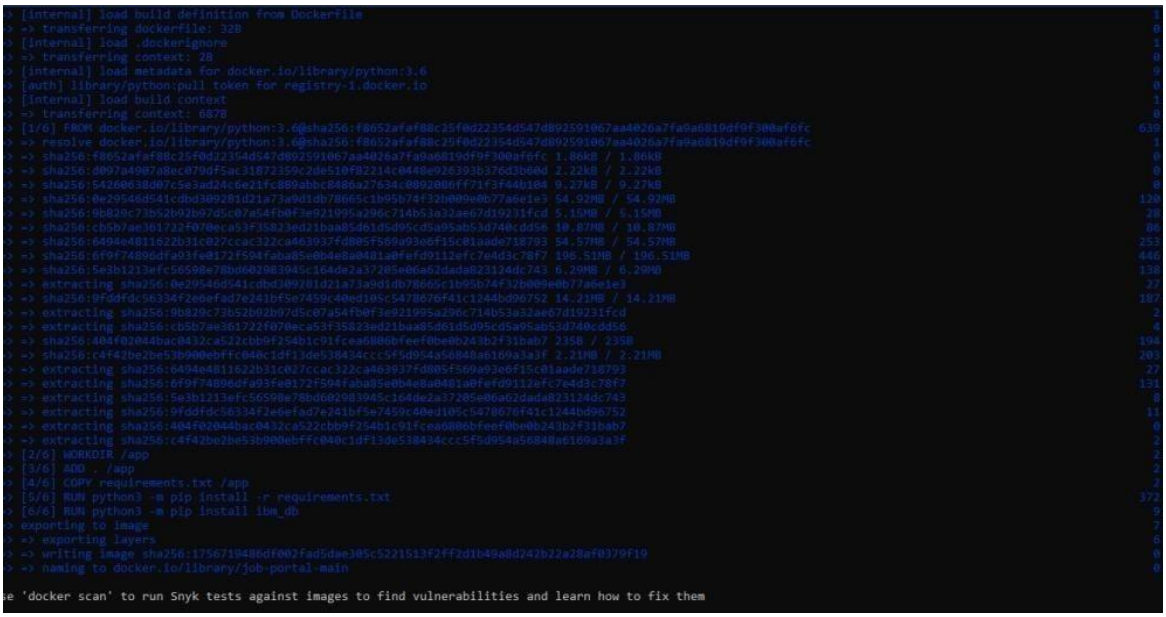
```
C:\Users\jayap>docker pull jayaprakash2019/docker101tutorial
Using default tag: latest
latest: Pulling from jayaprakash2019/docker101tutorial
Digest: sha256:ec5151b5ed5ba01931614d029ba15e99e319d514534fb93528b9d358b75e8b0f
Status: Image is up to date for jayaprakash2019/docker101tutorial:latest
docker.io/jayaprakash2019/docker101tutorial:latest
```

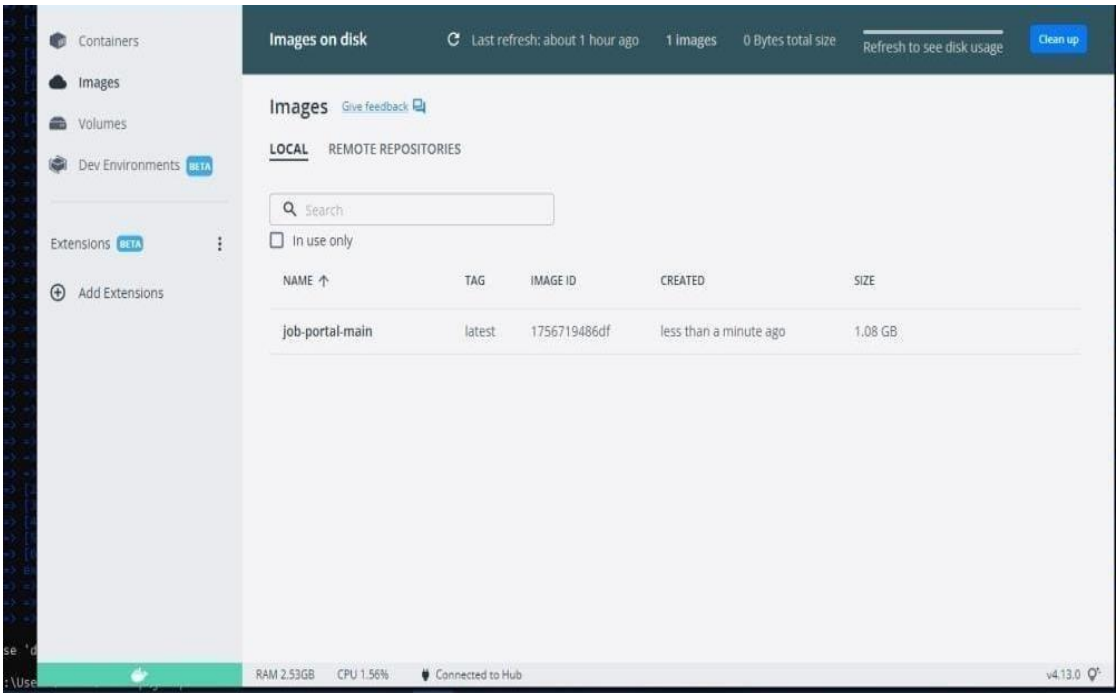
STEP-5 :Run the docker container

```
C:\Users\jayap>docker run -p 8081:8081 jayaprakash2019/docker101tutorial
```



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

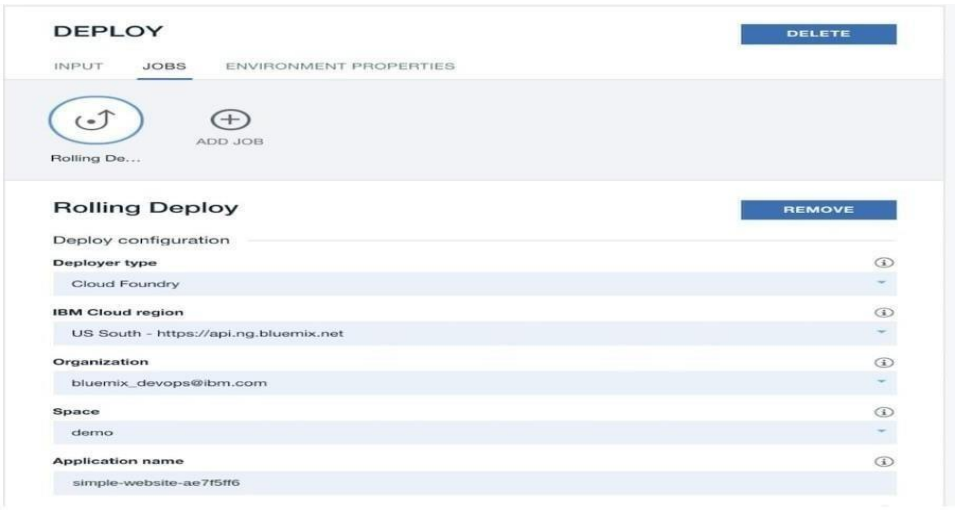




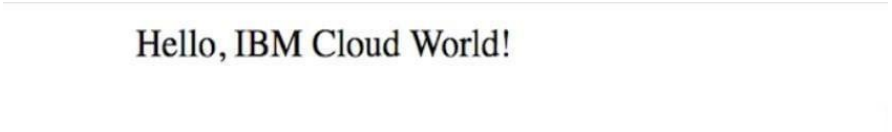
3. CREATE A IBM CONTAINER REGISTRY AND DEPLOY HELLOWORLD APP OR JOBPORTALAPP.

Solution:

```
        <html>
    <body>
        Hello,IBMCloudWorld!
    </body>
</html>---
application
s:
- buildpack: https://github.com/cloudfoundry/staticfile-
  buildpack.githost:simple-website-${random}
  name: simple-website-
  ${random}memory:64M
  stack:cflinuxfs2
```



```
{
  "ServiceId": "com.ibm.cloudoe.orion.client.deploy",
  "Params": {
    "Target": {
      "Url": "https://api.ng.bluemix.net",
      "Org": "bluemix_devops@ibm.com",
      "Space": "demo"
    },
    "Name": "simple-website-ae7f5ff6",
    "Instrumentation": {}
  },
  "Path": "manifest.yml",
  "Type": "Cloud Foundry"
}
```



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:

ibmcloudtarget-g<resource_group_name>ibmcloudcrnishanthc-add <your_nishanthc>ibmcloudresource service-instance-create example-postgresql databases-for-postgresql standard us- southibmcloudks cluster-service-bind mycluster default example-postgresqlgit clone -b node git@github.com:IBM-Cloud/clouddatabases-helloworld-kubernetes-examples.gitspec:

replicas:3name:cloudpostgres-nodejs-app

image:"registry.<region>.bluemix.net/<namespace>/icdpg"#Editme

imagePullPolicy: AlwaysibmcloudcrregionYou are targeting region 'us-south', the registry is'registry.ng.bluemix.net'.ibmcloudcr build -t registry.ng.bluemix.net/<namespace>/icdpg .ibmcloudcrimages

env:

- name:

BINDINGvalue

From:

```
            secretKeyRef:
              name: <postgres-secret-name> # Edit
              mekey:binding
apiVersion:
v1kind:
Servicemeta
data:
  name: cloudpostgres-
  servicelabels:
    run: clouddb-
demospec:
  type:
  NodePortsele
  ctor:
    run: clouddb-
  demoports:
  -
    protocol:TC
    Pport:8080
    nodePort:30081
```