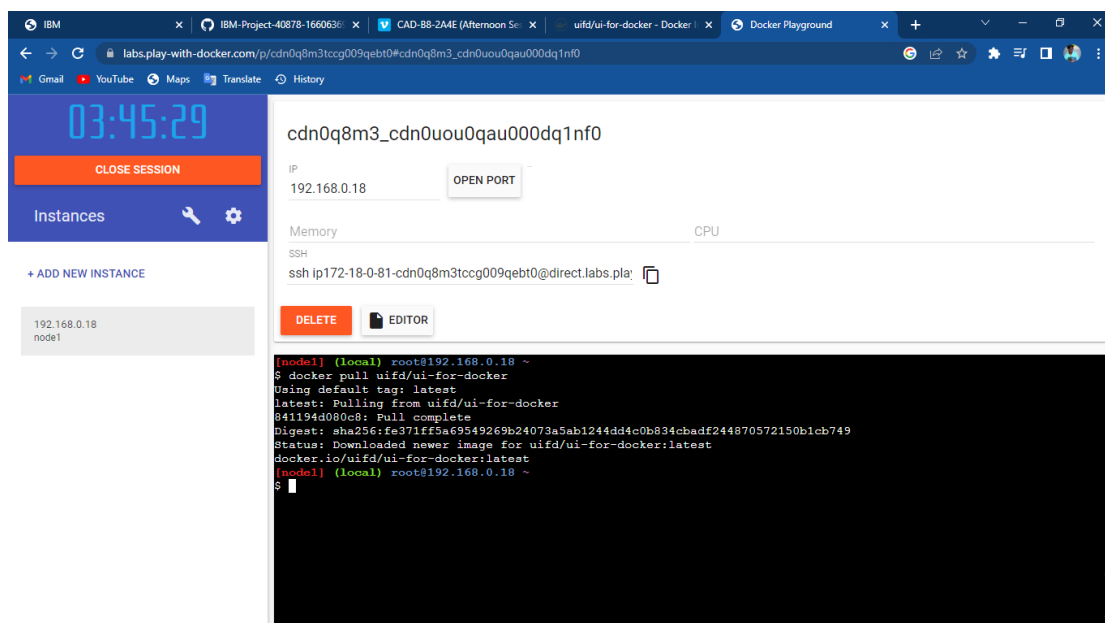
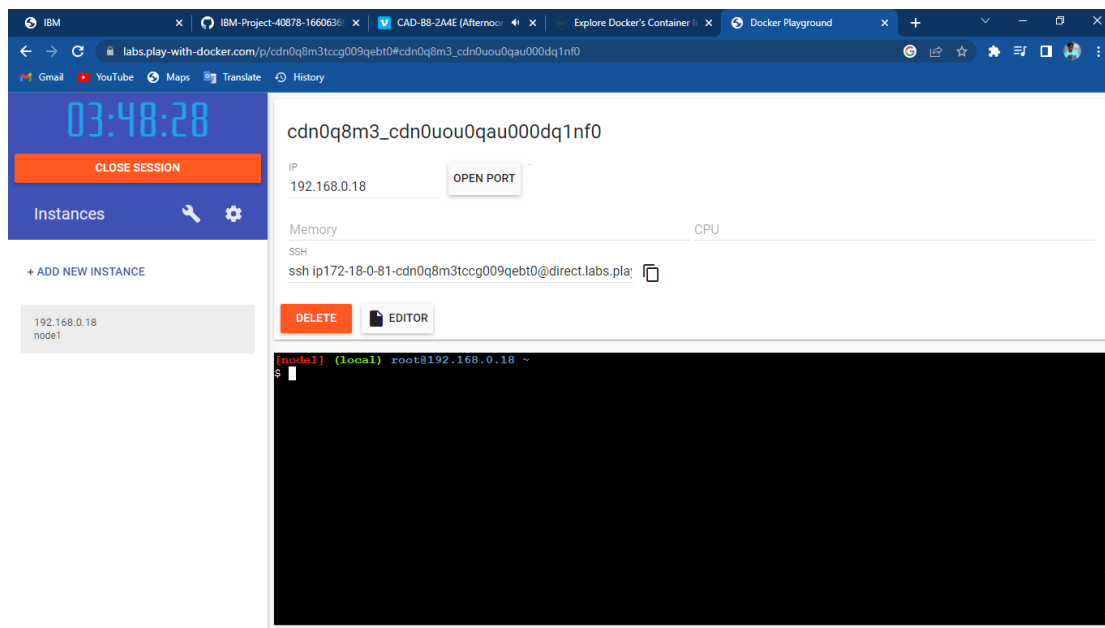


Assignment - 4

Kubernetes/Docker

Assignment Date	19 November 2022
Student Name	A.Shanthini
Student Roll Number	421319104025
Maximum Marks	2 Marks

1. Pull an image from docker hub and run it on docker playground.



To run a pulled image in docker playground, copy the run command of the pulled image.

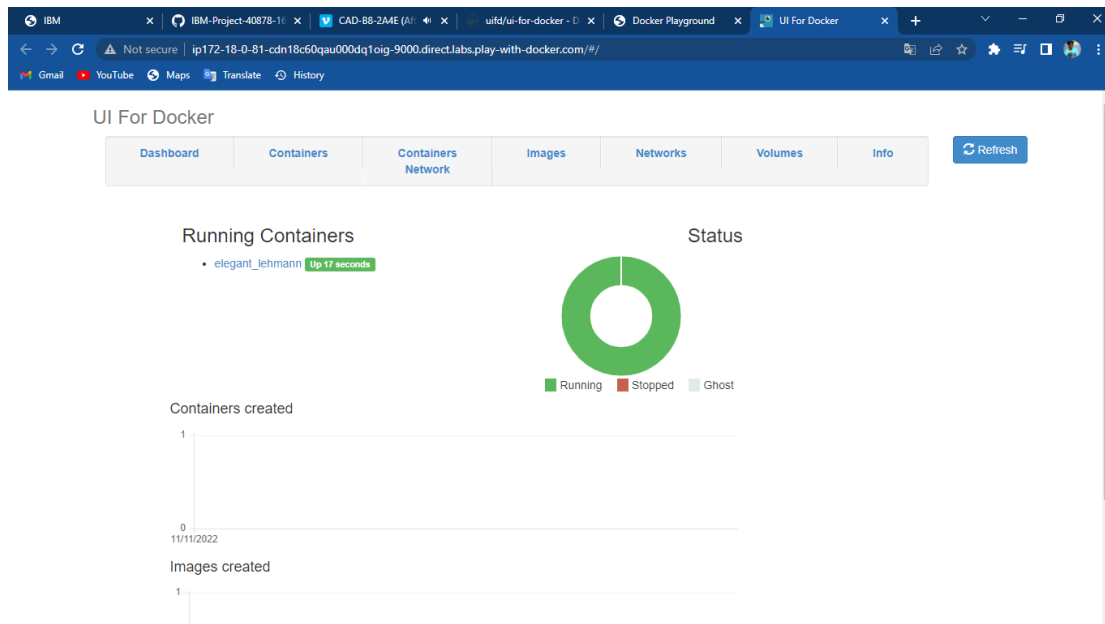
The screenshot shows the Docker Hub page for the repository `uifd/ui-for-docker`. The page includes a description, goals, a quickstart section, and a section for specifying the socket to connect to the Docker daemon. The quickstart section contains the following steps:

- Run `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://<dockerd host ip>:9000`

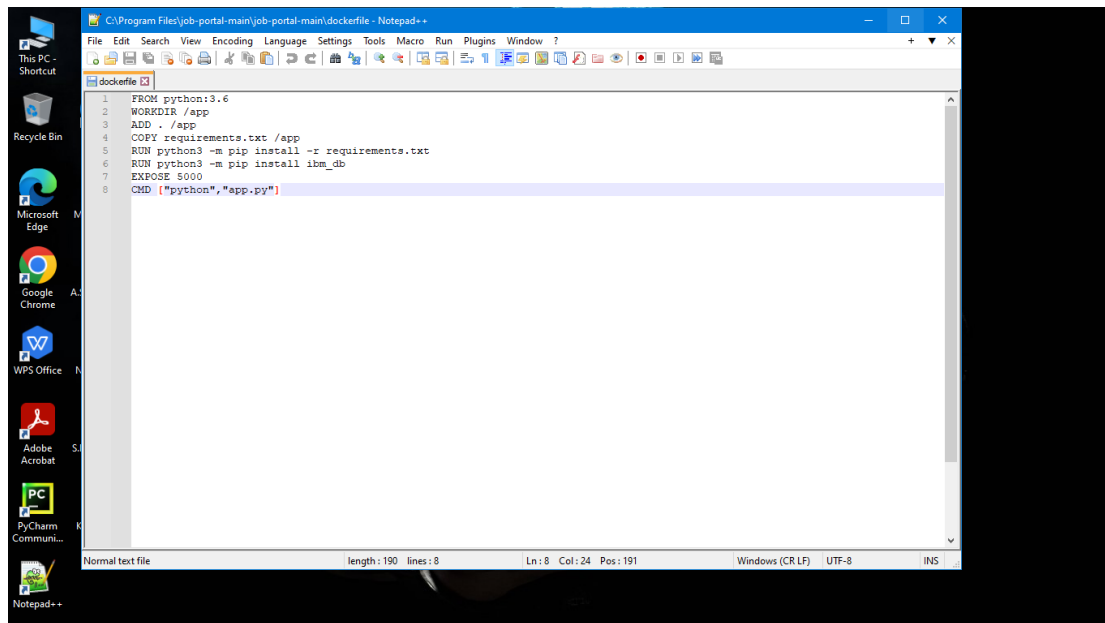
The 'Specify socket to connect to Docker daemon' section explains that by default, UI For Docker connects to the Docker daemon with `/var/run/docker.sock`. It also mentions that you can use the `-H` flag to change this socket.

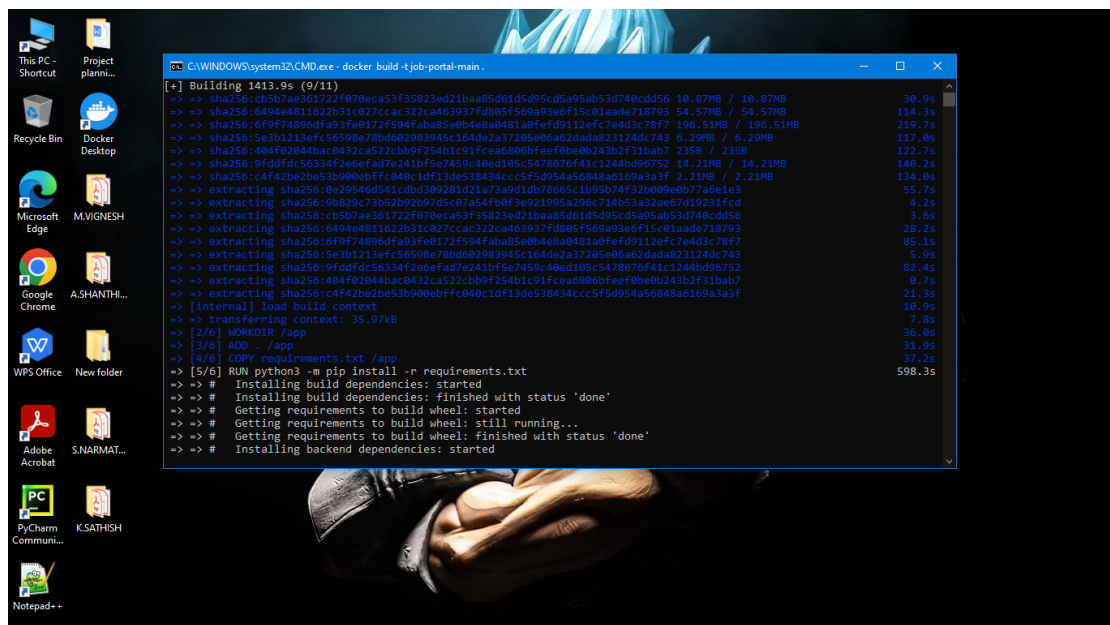
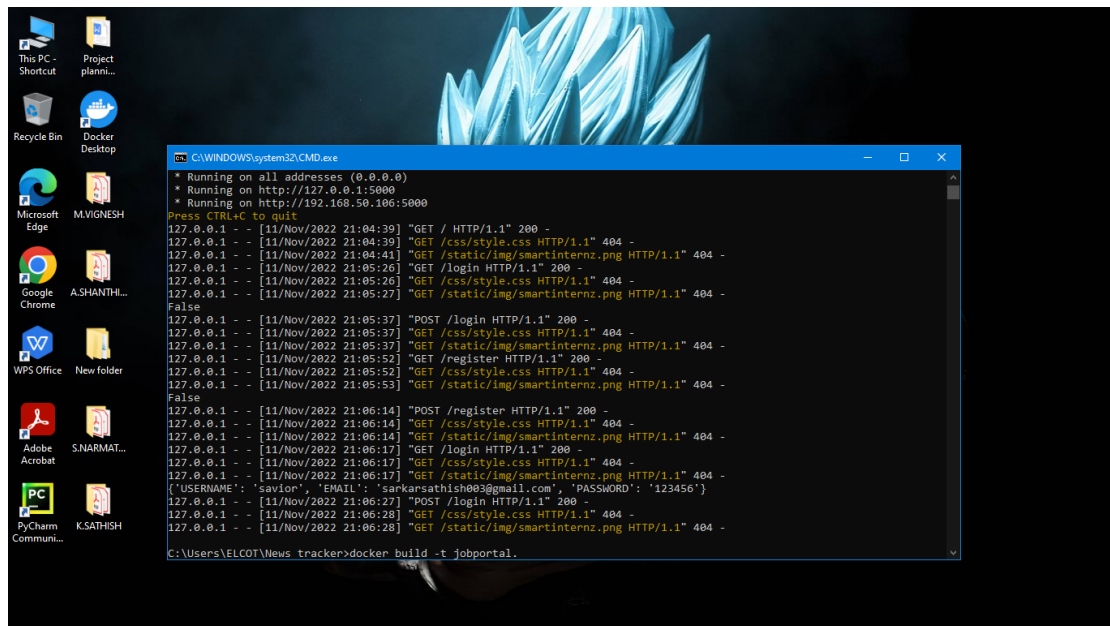
The screenshot shows the Docker Playground interface. On the left, there is a sidebar with a clock showing 03:37:25, a 'CLOSE SESSION' button, and a list of instances. The main area displays the details of a running instance named `cdn0q8m3_cdn0uou0qau000dq1nf0`. The instance's IP is `192.168.0.18`. The 'SSH' field shows the command `ssh ip172-18-0-81-cdn0q8m3tccg009qebt0@direct.labs.pla`. Below this, there is a terminal window showing the following commands and output:

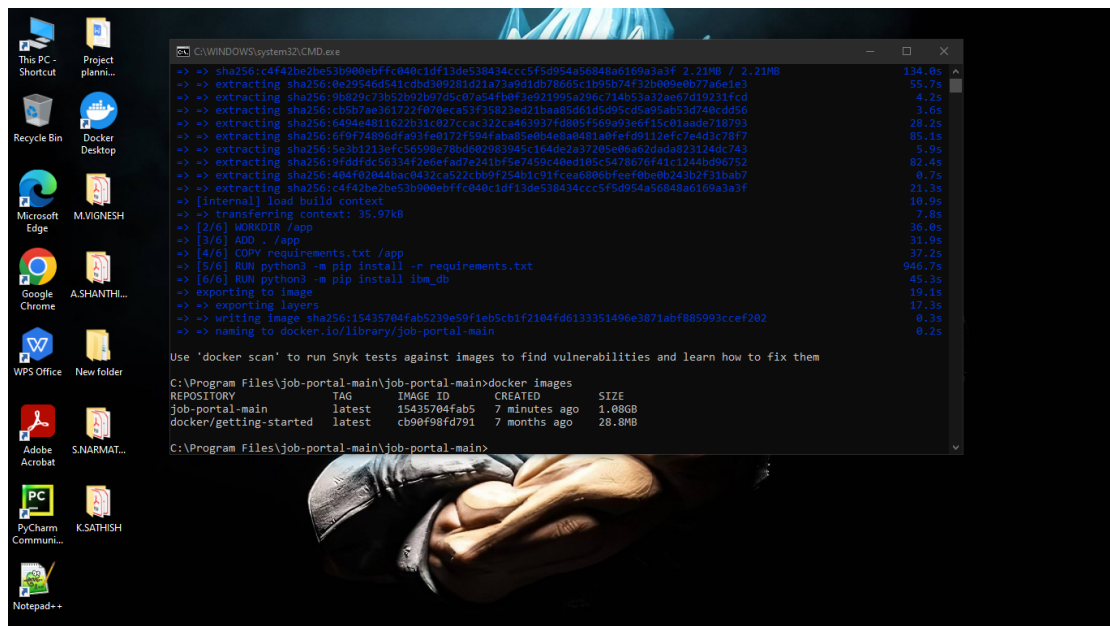
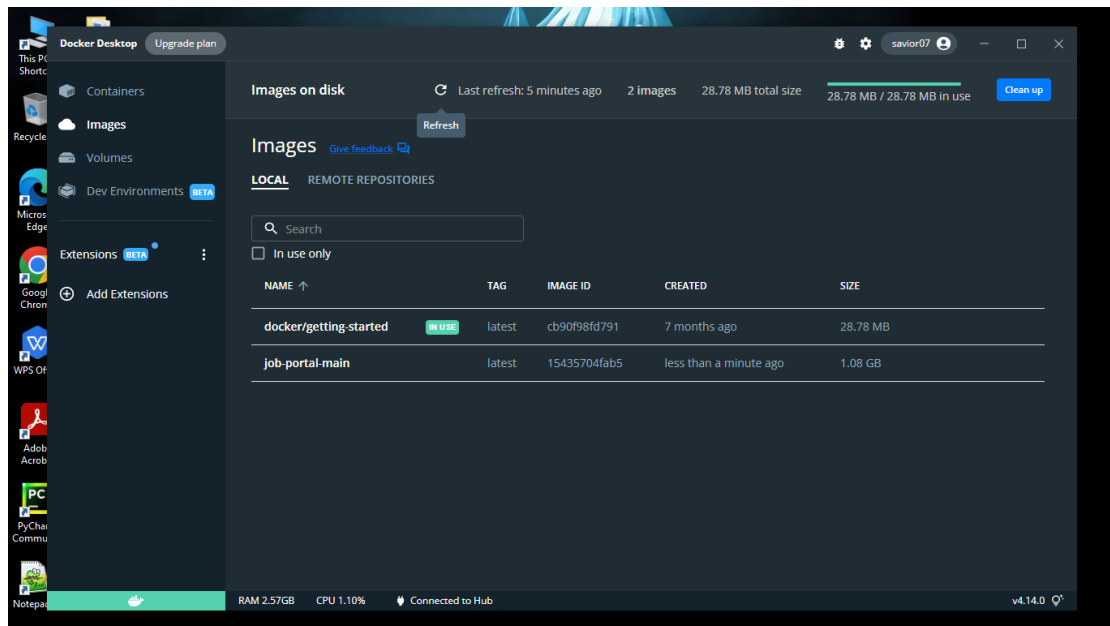
```
[node1] (local) root@192.168.0.18 ~
$ docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
841194d080c8: Pull complete
Digest: sha256:fe371ff5a69549269b24079a5ab1244dd4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[Run] docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
Run: docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
bash: Run:: command not found
[udocker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
a1a76af493904262dc7c61b3a2eb94c2faa32dd2b643189ff1cb1a49e23f8496k1/var/run/docker.sock uifd/ui-for-docker
[node1] (local) root@192.168.0.18 ~
```



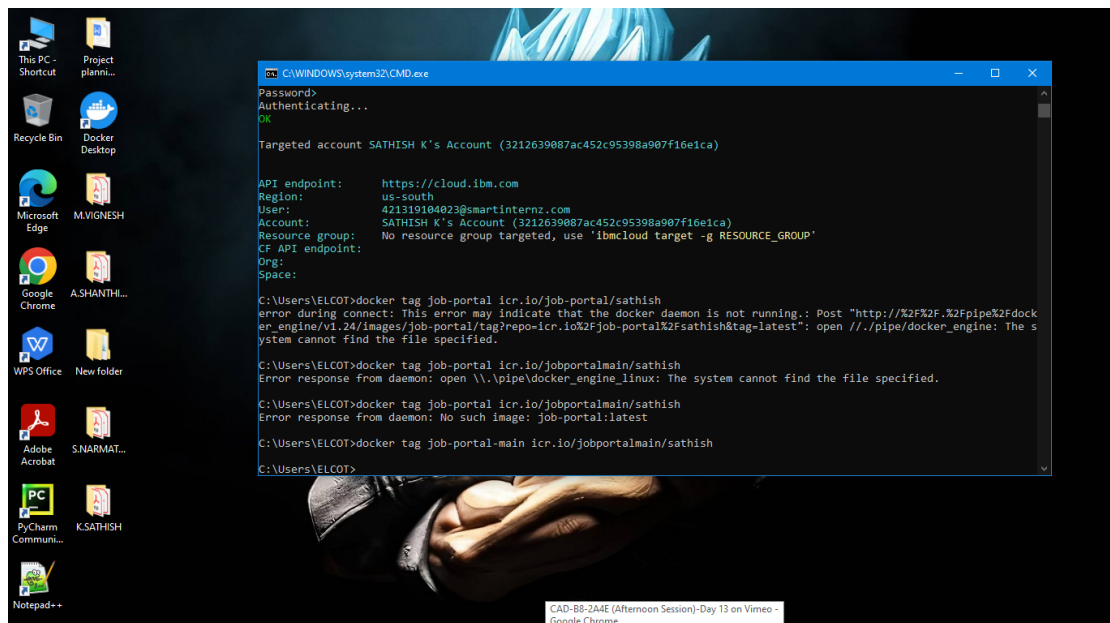
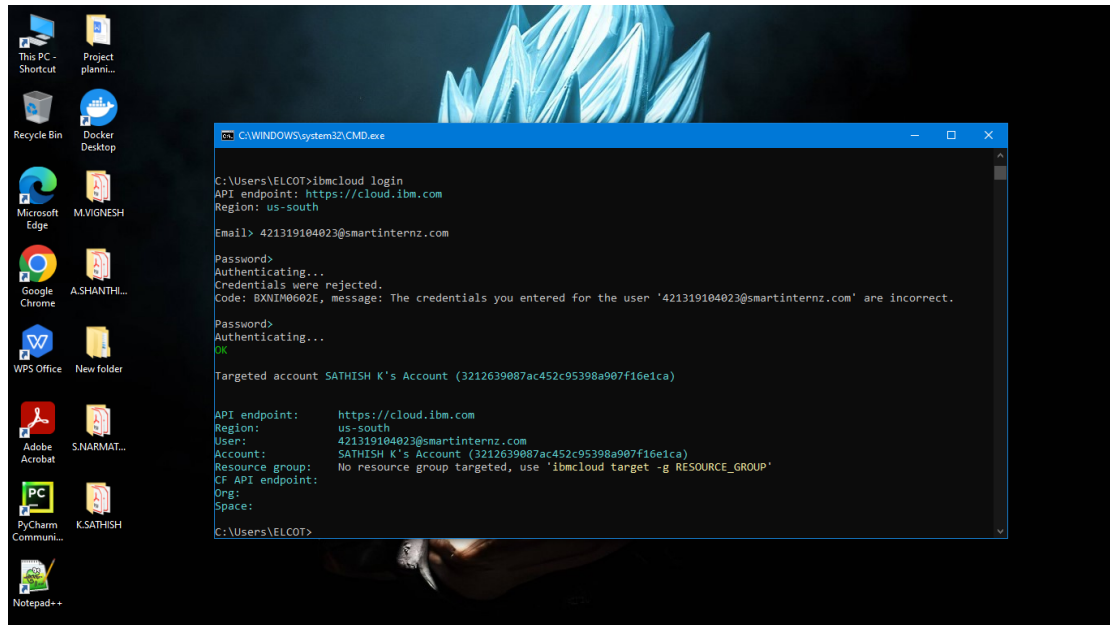
2. Create a docker file for the jobportal application and deploy it in Docker desktop application.



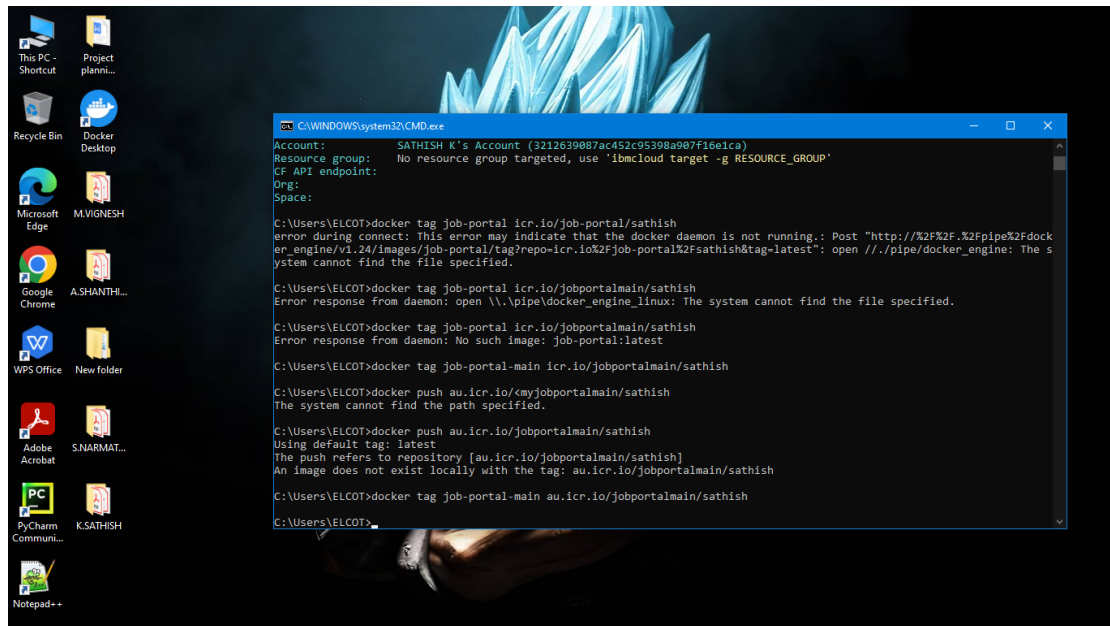




3. Create a IBM container registry and deploy helloworld app or jobportal app.



Pushing the image to repository



The screenshot shows a Windows desktop with a terminal window titled "C:\WINDOWS\system32\CMD.exe". The terminal displays the following commands and output:

```
Account: SATHISH K's Account (3212639087ac452c95398a087f16e1ca)
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:
Org:
Space:

C:\Users\ELCOT>docker tag job-portal icr.io/job-portal/sathish
error during connect: This error may indicate that the docker daemon is not running.: Post "http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.24/images/job-portal/tag?repo=icr.io%2Fjob-portal%2Fsathish&tag=latest": open //./pipe/docker_engine: The system cannot find the file specified.

C:\Users\ELCOT>docker tag job-portal icr.io/jobportalmain/sathish
Error response from daemon: open \\.\pipe\docker_engine_linux: The system cannot find the file specified.

C:\Users\ELCOT>docker tag job-portal icr.io/jobportalmain/sathish
Error response from daemon: No such image: job-portal:latest

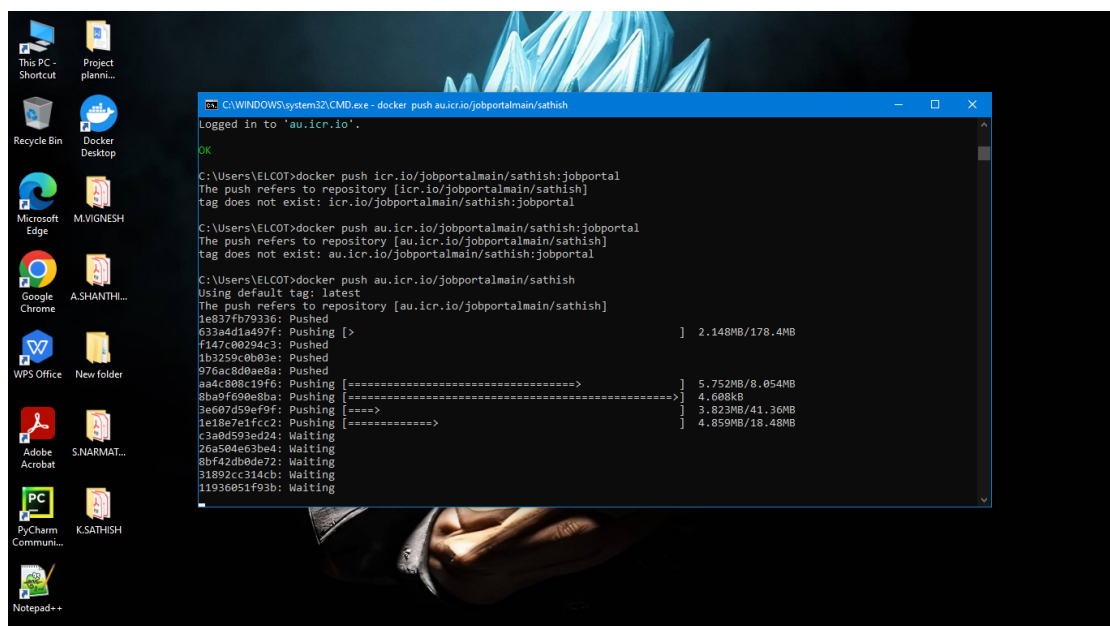
C:\Users\ELCOT>docker tag job-portal-main icr.io/jobportalmain/sathish

C:\Users\ELCOT>docker push au.icr.io/myjobportalmain/sathish
The system cannot find the path specified.

C:\Users\ELCOT>docker push au.icr.io/jobportalmain/sathish
Using default tag: latest
The push refers to repository [au.icr.io/jobportalmain/sathish]
An image does not exist locally with the tag: au.icr.io/jobportalmain/sathish

C:\Users\ELCOT>docker tag job-portal-main au.icr.io/jobportalmain/sathish

C:\Users\ELCOT>
```



The screenshot shows a Windows desktop with a terminal window titled "C:\WINDOWS\system32\CMD.exe - docker push au.icr.io/jobportalmain/sathish". The terminal displays the following commands and output:

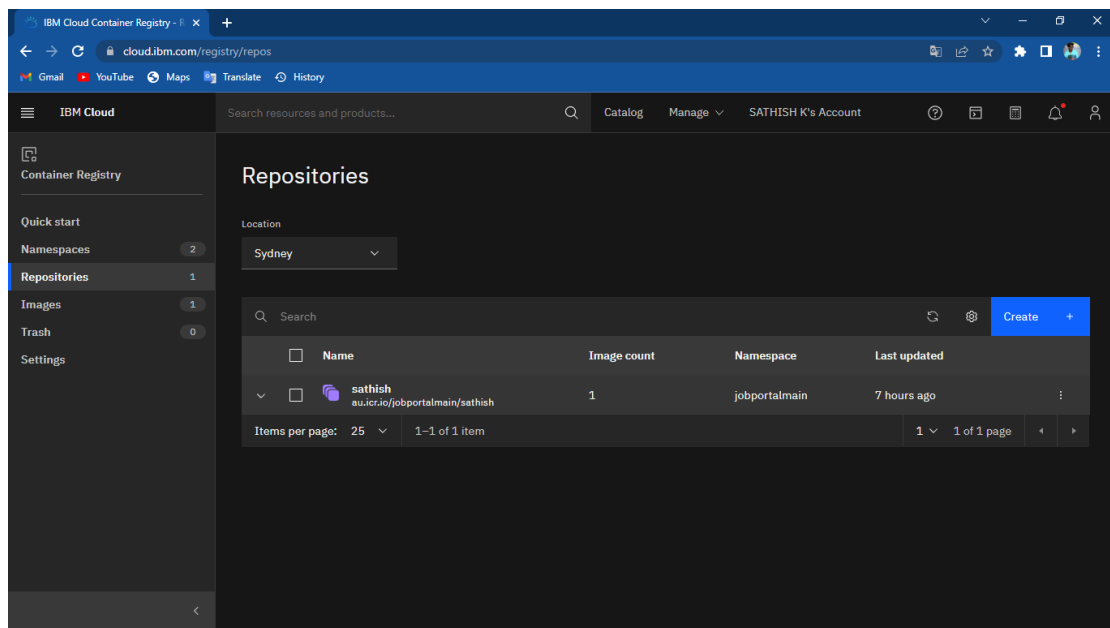
```
Logged in to 'au.icr.io'.
OK

C:\Users\ELCOT>docker push icr.io/jobportalmain/sathish:jobportal
tag does not exist: icr.io/jobportalmain/sathish:jobportal

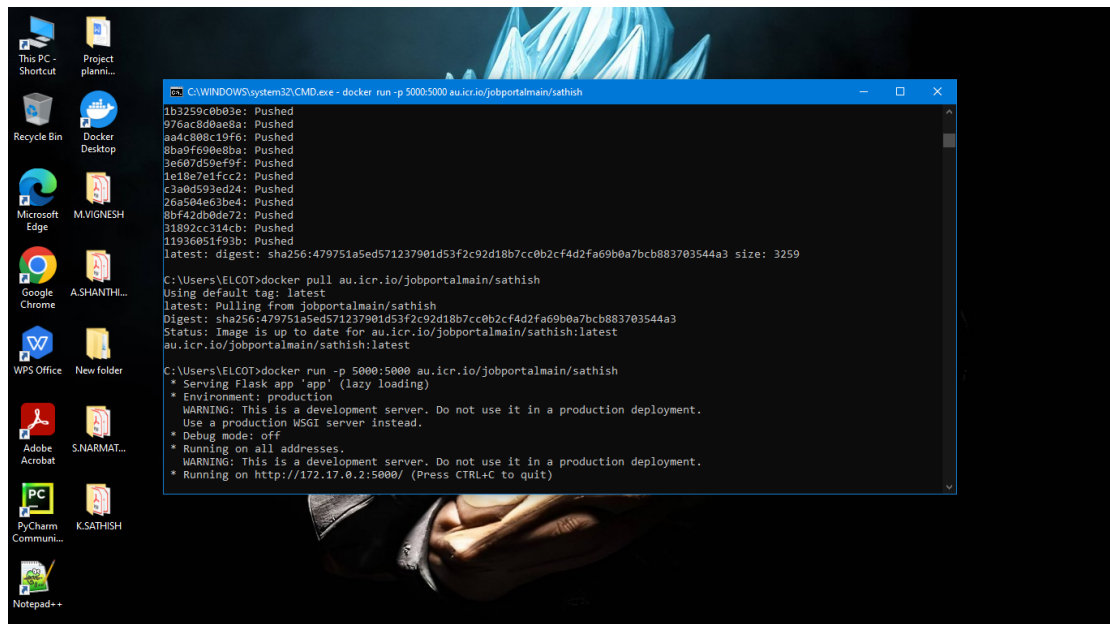
C:\Users\ELCOT>docker push au.icr.io/jobportalmain/sathish:jobportal
The push refers to repository [au.icr.io/jobportalmain/sathish]
tag does not exist: au.icr.io/jobportalmain/sathish:jobportal

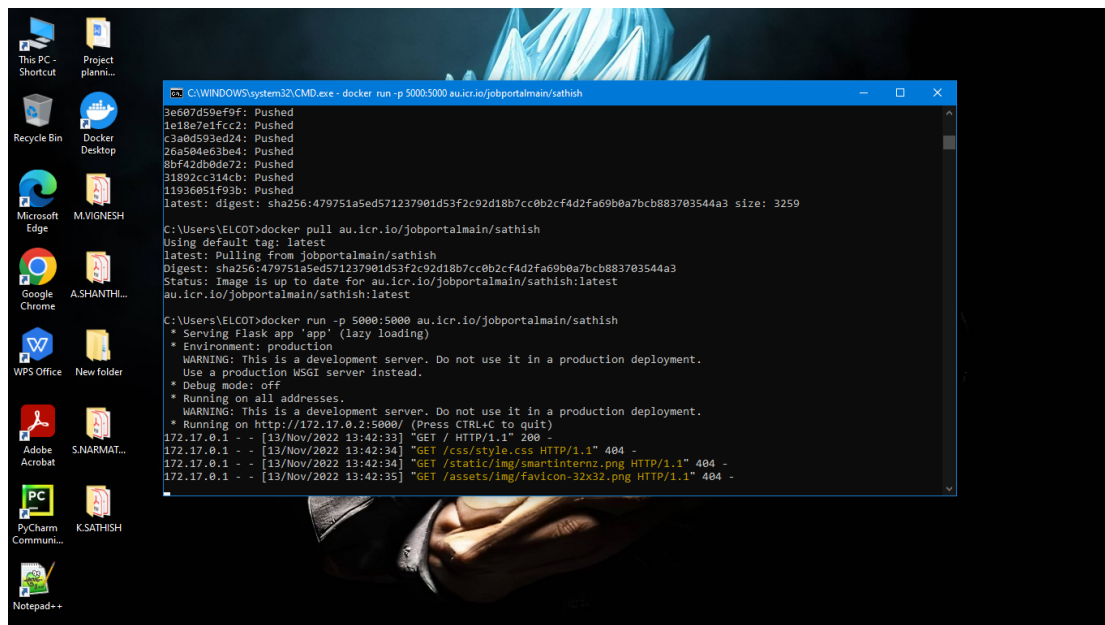
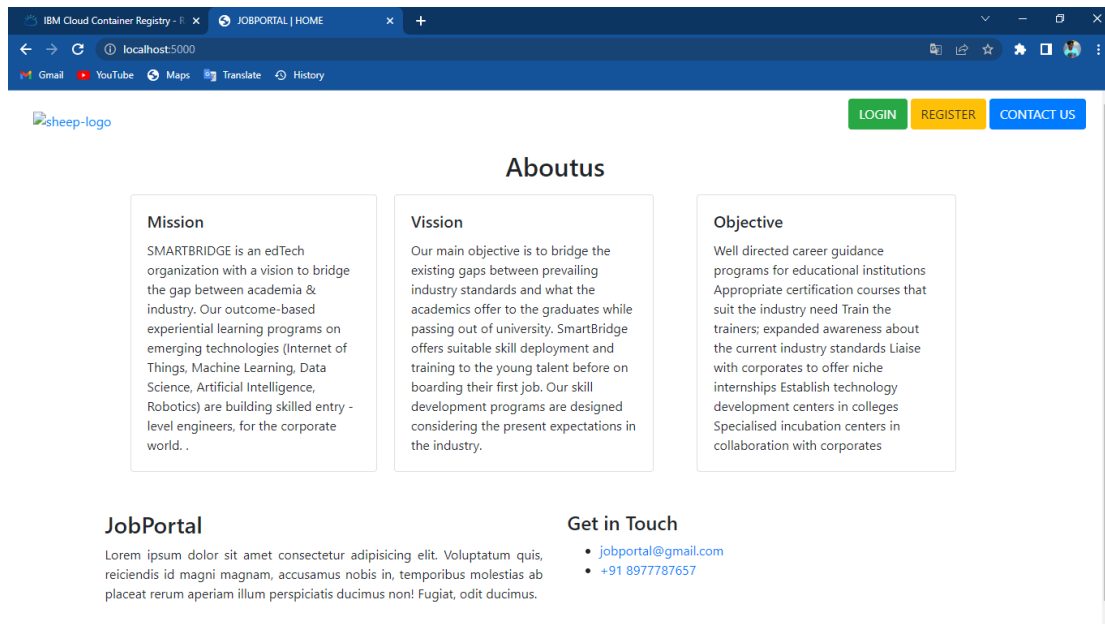
C:\Users\ELCOT>docker push au.icr.io/jobportalmain/sathish
Using default tag: latest
The push refers to repository [au.icr.io/jobportalmain/sathish]

1e037f079336: Pushed
633a4d1a497f: Pushed [>] 2.148MB/178.4MB
f147c00294c3: Pushed
1b3259c0b03e: Pushed
976ac8d0ae8a: Pushed
aa4c00c19f6: Pushing [=====] 5.752MB/8.054MB
8ba9f690ebba: Pushing [=====] 4.608KB
3e607d59ef9f: Pushing [=====] 3.823MB/41.36MB
1e18e7e1fcc2: Pushing [=====] 4.859MB/18.48MB
c3a0d593ed24: Waiting
26a504e63be4: Waiting
8bf42db0de72: Waiting
31892cc314cb: Waiting
11936051f93b: Waiting
```

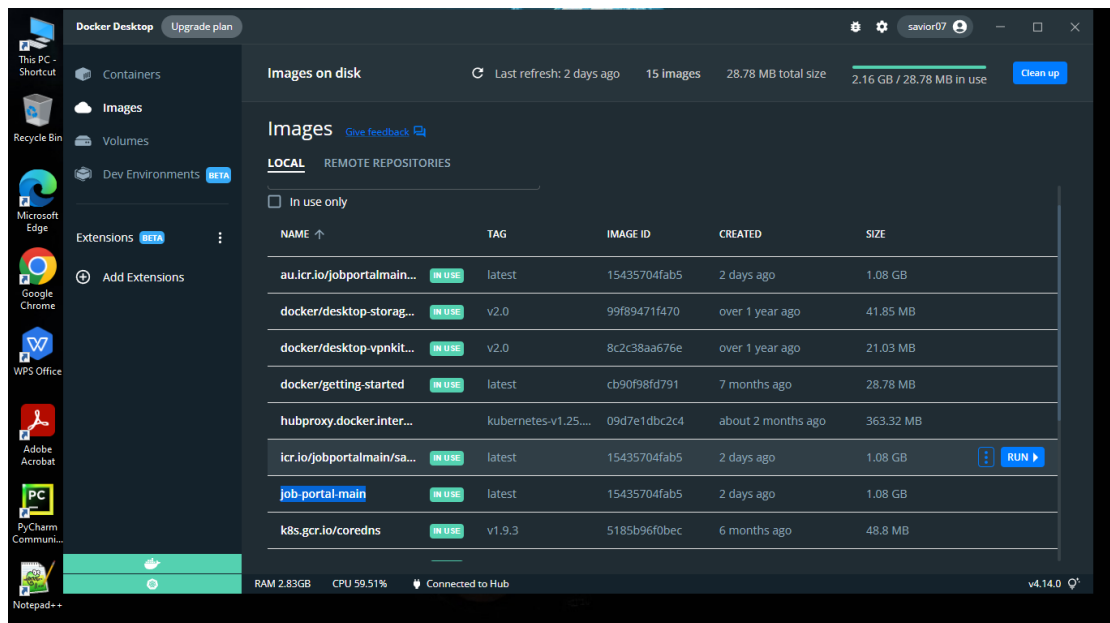
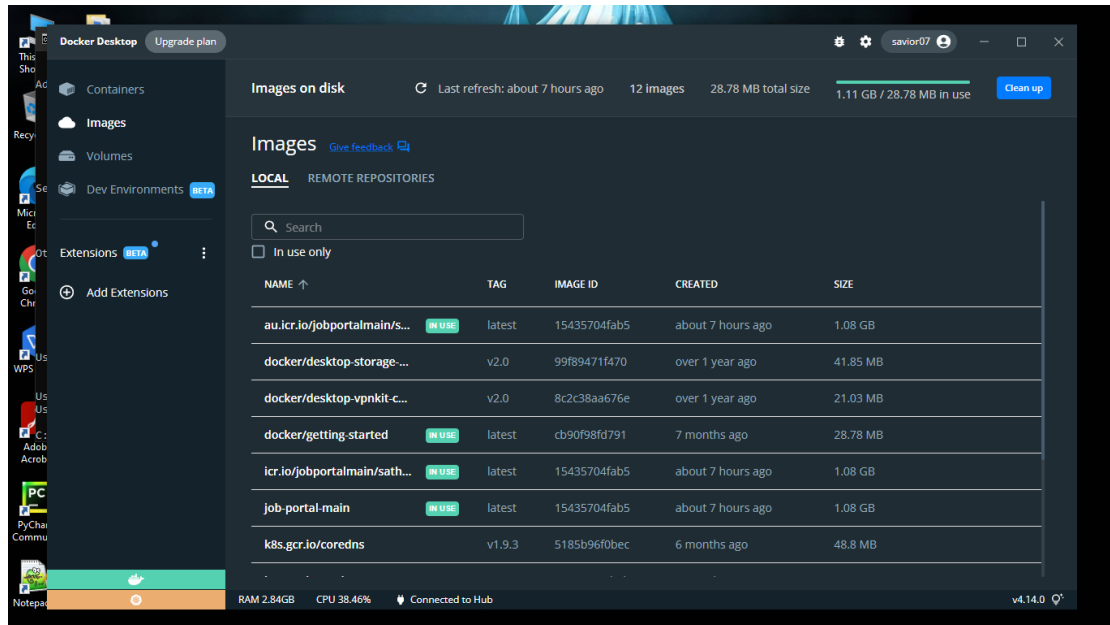


Pulling the image from IBM container and running it.





4. Create a Kubernetes Cluster in IBM cloud and deploy hello world image or job portal image.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ELCOT>kubectl
kubectl controls the Kubernetes cluster manager.

Find more information at: https://kubernetes.io/docs/reference/kubectl/

Basic Commands (Beginner):
  create      Create a resource from a file or from stdin
  expose      Take a replication controller, service, deployment or pod and expose it as a new Kubernetes service
  run         Run a particular image on the cluster
  set         Set specific features on objects

Basic Commands (Intermediate):
  explain     Get documentation for a resource
  get         Display one or many resources
  edit        Edit a resource on the server
  delete      Delete resources by file names, stdin, resources and names, or by resources and label selector

Deploy Commands:
  rollout     Manage the rollout of a resource
  scale       Set a new size for a deployment, replica set, or replication controller
  autoscale   Auto-scale a deployment, replica set, stateful set, or replication controller

Cluster Management Commands:
  certificate Modify certificate resources.
  cluster-info Display cluster information
  top          Display resource (CPU/memory) usage
  cordon       Mark node as unschedulable
  uncordon     Mark node as schedulable
  drain        Drain node in preparation for maintenance
  taint        Update the taints on one or more nodes

Troubleshooting and Debugging Commands:
  describe    Show details of a specific resource or group of resources
  logs        Print the logs for a container in a pod
  attach      Attach to a running container
  exec        Execute a command in a container
  port-forward Forward one or more local ports to a pod
  proxy       Run a proxy to the Kubernetes API server
  cp          Copy files and directories to and from containers
  auth        Inspect authorization
  debug       Create debugging sessions for troubleshooting workloads and nodes

Advanced Commands:
```

```
C:\WINDOWS\system32\cmd.exe - docker login
kubectl [flags] [options]

Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).

C:\Users\ELCOT>cd C:\Program Files\job-portal-main\job-portal-main
C:\Program Files\job-portal-main\job-portal-main>kubectl create -f deployment.yaml
error: the path "deployment.yaml" does not exist

C:\Program Files\job-portal-main\job-portal-main>kubectl create -f deployment.yaml
error: the path "deployment.yaml" does not exist

C:\Program Files\job-portal-main\job-portal-main>docker login
Authenticating with existing credentials...
Login Succeeded
```

```
C:\WINDOWS\system32\CMD.exe
--template='':
  Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format
  is golang templates [http://golang.org/pkg/text/template/#pkg-overview].

--validate='strict':
  Must be one of: strict (or true), warn, ignore (or false).          "true" or "strict" will use a schema to validate
  the input and fail the request if invalid. It will perform server side validation if ServerSideFieldValidation
  is enabled on the api-server, but will fall back to less reliable client-side validation if not.          "warn" will
  warn about unknown or duplicate fields without blocking the request if server-side field validation is enabled
  on the API server, and behave as "ignore" otherwise.          "false" or "ignore" will not perform any schema
  validation, silently dropping any unknown or duplicate fields.

--windows-line-endings=true:
  Only relevant if --edit=true. Defaults to the line ending native to your platform.

Usage:
  kubectl create -f FILENAME [options]

Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).

C:\Program Files\job-portal-main\job-portal-main>kubectl create -f service.yaml
error: the path "service.yaml" does not exist

C:\Program Files\job-portal-main\job-portal-main>kubectl create -f deployment.yaml
deployment.apps/flask-node-deployment created

C:\Program Files\job-portal-main\job-portal-main>
```

```
C:\WINDOWS\system32\CMD.exe
--template='':
  Template string or path to template file to use when -o=go-template, -o=go-template-file. The template format
  is golang templates [http://golang.org/pkg/text/template/#pkg-overview].

--validate='strict':
  Must be one of: strict (or true), warn, ignore (or false).          "true" or "strict" will use a schema to validate
  the input and fail the request if invalid. It will perform server side validation if ServerSideFieldValidation
  is enabled on the api-server, but will fall back to less reliable client-side validation if not.          "warn" will
  warn about unknown or duplicate fields without blocking the request if server-side field validation is enabled
  on the API server, and behave as "ignore" otherwise.          "false" or "ignore" will not perform any schema
  validation, silently dropping any unknown or duplicate fields.

--windows-line-endings=true:
  Only relevant if --edit=true. Defaults to the line ending native to your platform.

Usage:
  kubectl create -f FILENAME [options]

Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).

C:\Program Files\job-portal-main\job-portal-main>kubectl create -f service.yaml
error: the path "service.yaml" does not exist

C:\Program Files\job-portal-main\job-portal-main>kubectl create -f deployment.yaml
deployment.apps/flask-node-deployment created

C:\Program Files\job-portal-main\job-portal-main>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
flask-node-deployment-8489d6db57-pmjg4  0/1     ContainerCreating   0           3m13s

C:\Program Files\job-portal-main\job-portal-main>
```