

## ASSIGNMENT 4

Assignment Date	27 October 2022
Student Name	Mahjabeen A
Student Roll Number	2019103542
Maximum Marks	2 marks

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

Pulling alpine

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:33:48, a 'CLOSE SESSION' button, and a list of instances. The main area displays the instance name 'cdtooqe3\_cdttopam0qau0009mh23g' and its IP '192.168.0.18'. Below this, there's a terminal window showing the following commands and output:

```
[node1] (local) root@192.168.0.18 ~
$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
ca7dd9ec2225: Pull complete
Digest: sha256:b95359c2505145f16c6aa384f9cc74eeff78eb36d308ca4fd902eeeb0a0b161b
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
alpine        latest   bfe296a52501   9 days ago    5.54MB
[node1] (local) root@192.168.0.18 ~
$
```

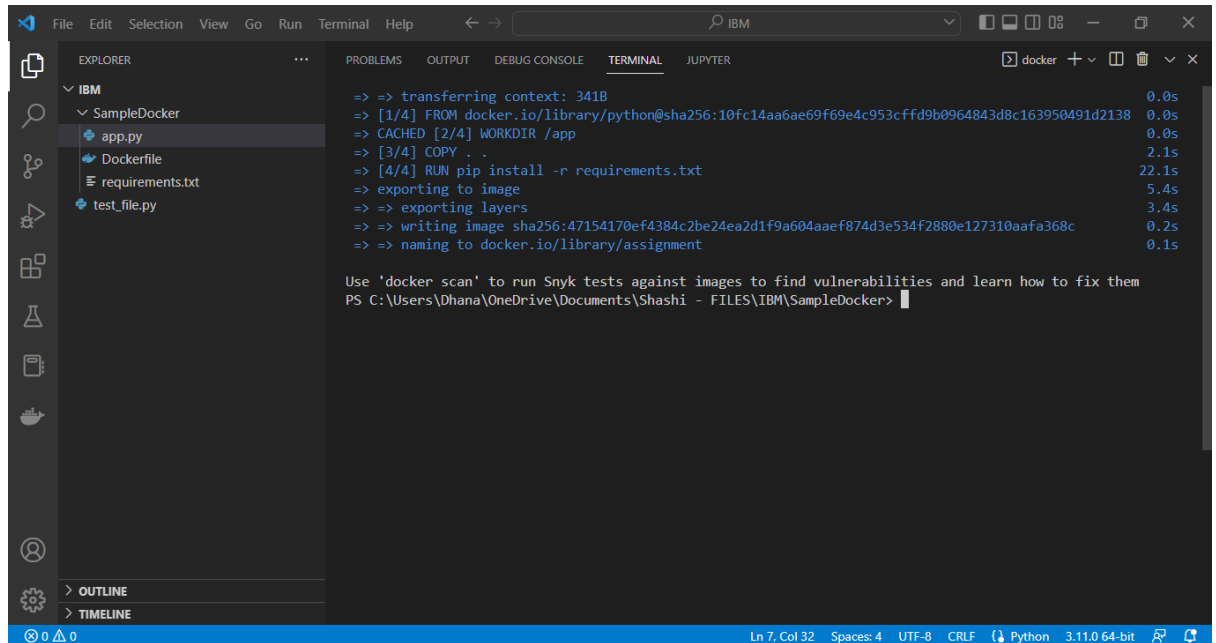
Running alpine on docker playground

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:19:32, a 'CLOSE SESSION' button, and a list of instances. The main area displays the instance name 'cdtooqe3\_cdttopam0qau0009mh23g' and its IP '192.168.0.18'. Below this, there's a terminal window showing the following commands and output:

```
[node1] (local) root@192.168.0.18 ~
$ docker run alpine echo "Hello from alpine"
Hello from alpine
[node1] (local) root@192.168.0.18 ~
$ docker run -it alpine /bin/sh
/ # uname -a
Linux 0f410d464c82 4.4.0-210-generic #242-Ubuntu SMP Fri Apr 16 09:57:56 UTC 2021 x86_64 Linux
/ # exit
[node1] (local) root@192.168.0.18 ~
$
```

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

### Build Docker Images from docker file

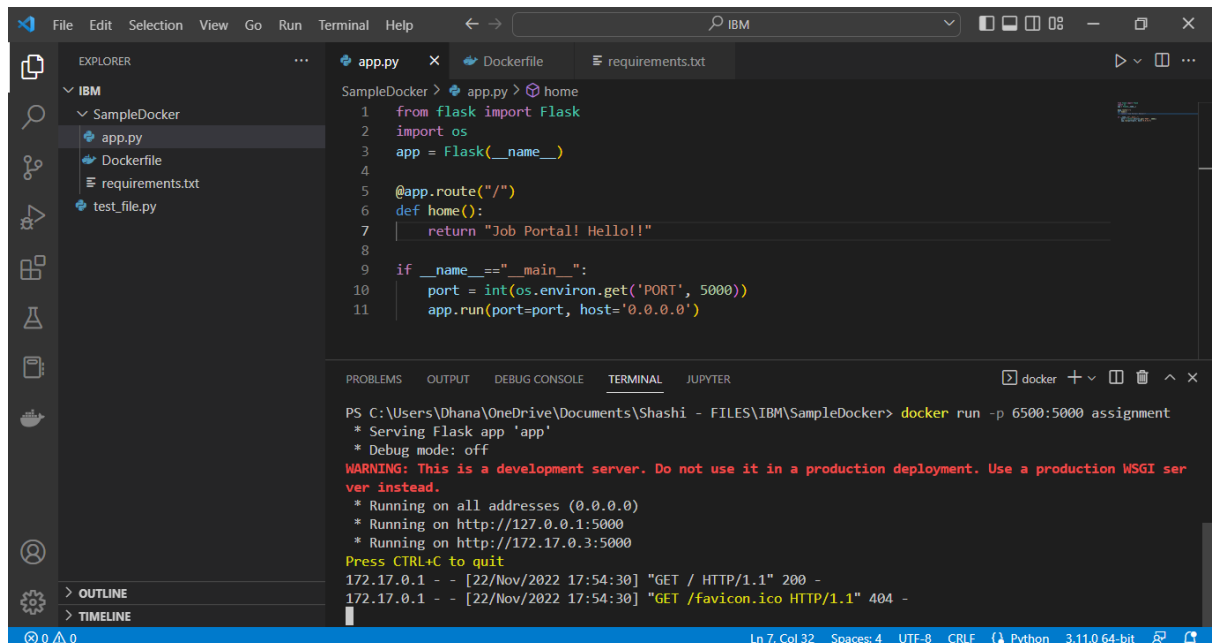


The screenshot shows the VS Code interface with the Explorer pane on the left displaying the file structure of a project named 'SampleDocker'. The files listed are 'app.py', 'Dockerfile', 'requirements.txt', and 'test\_file.py'. The Terminal pane on the right shows the output of the 'docker build' command. The build process includes transferring context, pulling the Python base image, copying the source files, installing dependencies, and exporting the image. The final output is a Docker image named 'assignment'.

```
=> => transferring context: 341B
=> [1/4] FROM docker.io/library/python@sha256:10fc14aa6ae69f69e4c953c ffd9b0964843d8c163950491d2138
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN pip install -r requirements.txt
=> exporting to image
=> exporting layers
=> writing image sha256:47154170ef4384c2be24ea2d1f9a604aaef874d3e534f2880e127310aafa368c
=> naming to docker.io/library/assignment

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
PS C:\Users\Dhana\OneDrive\Documents\Shashi - FILES\IBM\SampleDocker>
```

### Run container

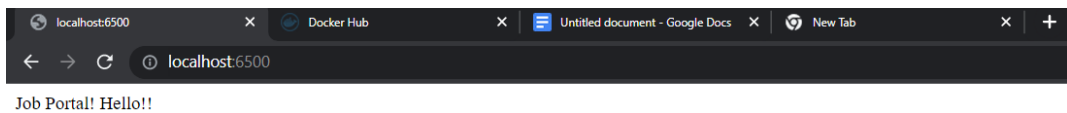


The screenshot shows the VS Code interface with the Explorer pane on the left displaying the file structure of a project named 'SampleDocker'. The files listed are 'app.py', 'Dockerfile', 'requirements.txt', and 'test\_file.py'. The Editor pane shows the code for 'app.py', which is a Flask application. The Terminal pane shows the output of the 'docker run' command, which starts the container and runs the application. The output shows the application running on port 5000 and responding to GET requests.

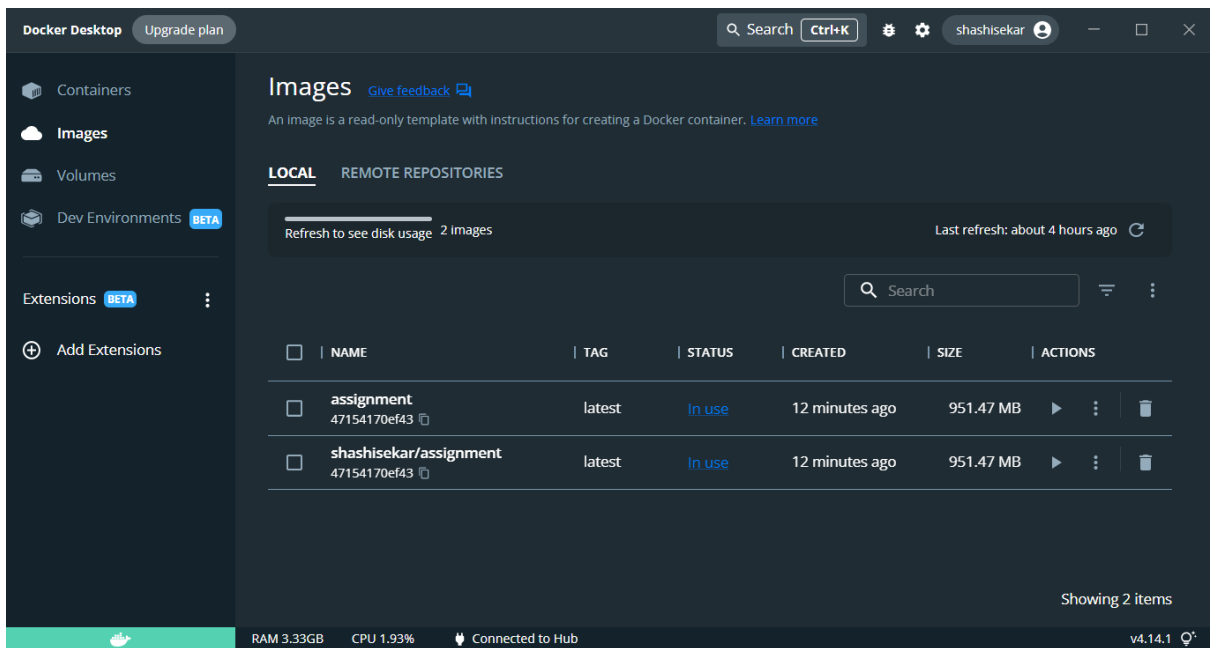
```
1 from flask import Flask
2 import os
3 app = Flask(__name__)
4
5 @app.route("/")
6 def home():
7     return "Job Portal! Hello!!"
8
9 if __name__ == "__main__":
10     port = int(os.environ.get('PORT', 5000))
11     app.run(port=port, host='0.0.0.0')
```

```
PS C:\Users\Dhana\OneDrive\Documents\Shashi - FILES\IBM\SampleDocker> docker run -p 6500:5000 assignment
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI ser
ver instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.3:5000
Press CTRL+C to quit
172.17.0.1 - - [22/Nov/2022 17:54:30] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [22/Nov/2022 17:54:30] "GET /favicon.ico HTTP/1.1" 404 -
```

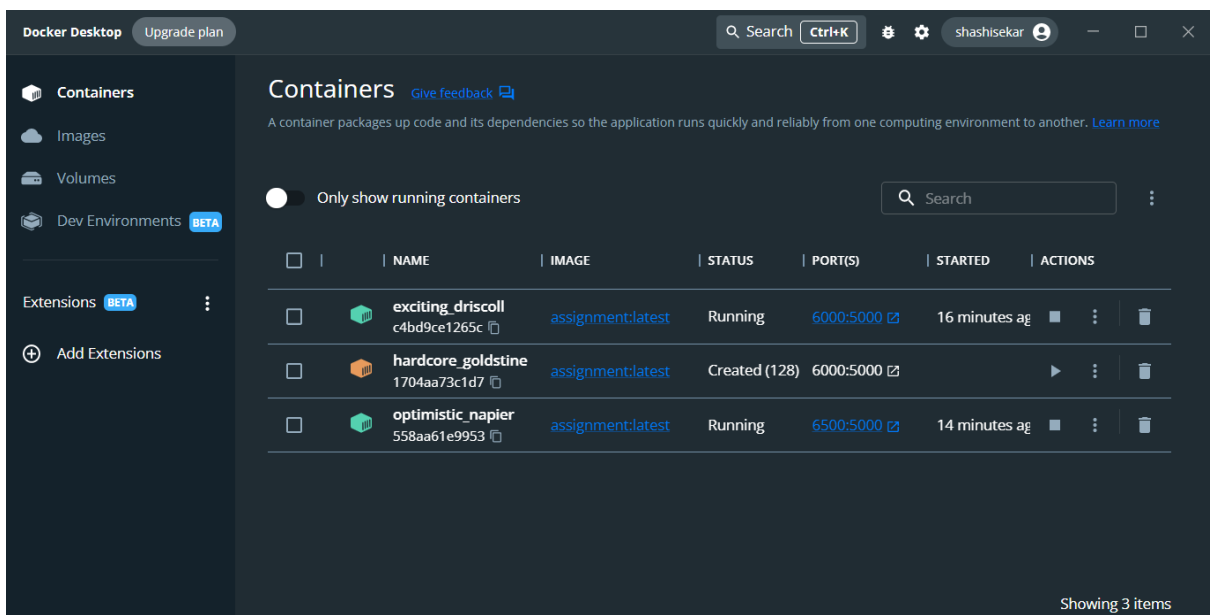
## Viewing the Output at PORT 6500 local host



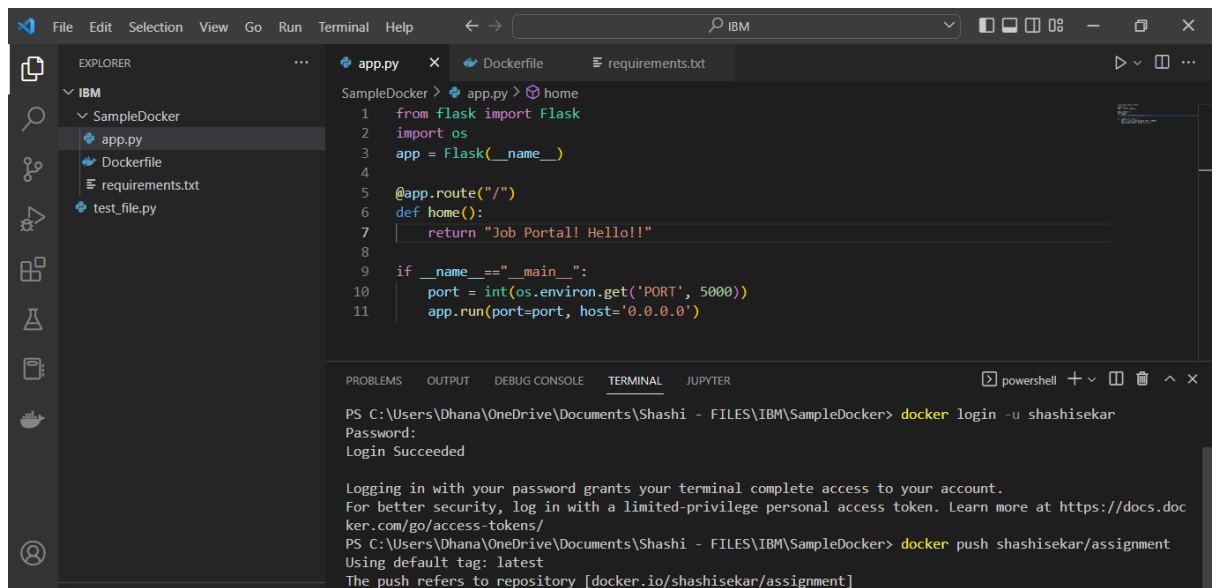
## Docker Images



## Docker Containers



## Deploying Docker images on Docker Hub



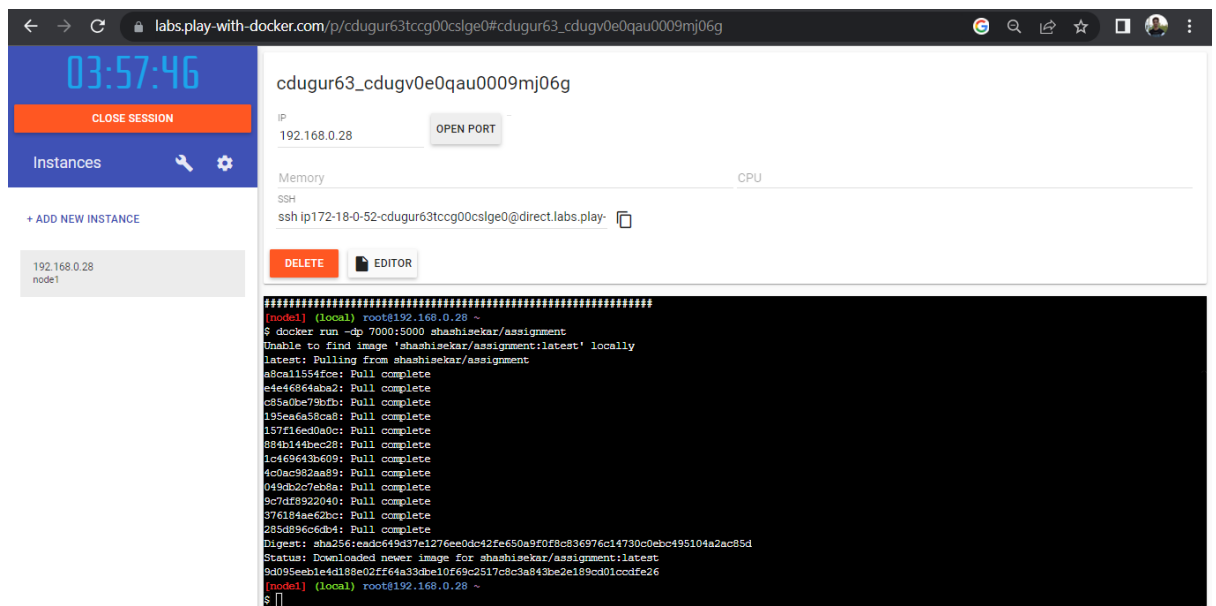
The screenshot shows a Visual Studio Code editor with a project named 'SampleDocker'. The Explorer sidebar on the left lists files: 'app.py', 'Dockerfile', 'requirements.txt', and 'test\_file.py'. The main editor window displays the 'app.py' file, which is a Flask application. The code defines a route for '/' that returns 'Job Portal! Hello!!'. Below the code editor, the TERMINAL panel shows the execution of Docker commands. The user logs in to Docker Hub with the username 'shashisekar' and then pushes the 'assignment' image to the repository 'shashisekar/assignment'.

```
SampleDocker > app.py > home
1 from flask import Flask
2 import os
3 app = Flask(__name__)
4
5 @app.route("/")
6 def home():
7     return "Job Portal! Hello!!"
8
9 if __name__ == "__main__":
10     port = int(os.environ.get('PORT', 5000))
11     app.run(port=port, host='0.0.0.0')
```

```
PS C:\Users\Dhana\OneDrive\Documents\Shashi - FILES\IBM\SampleDocker> docker login -u shashisekar
Password:
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.doc
ker.com/go/access-tokens/
PS C:\Users\Dhana\OneDrive\Documents\Shashi - FILES\IBM\SampleDocker> docker push shashisekar/assignment
Using default tag: latest
The push refers to repository [docker.io/shashisekar/assignment]
```

## Using Docker Play ground to pull the image and run it

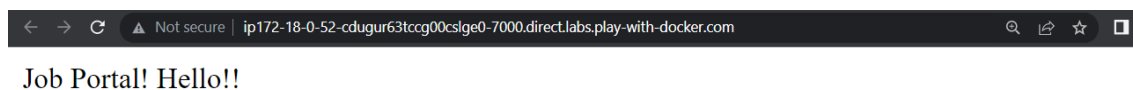


The screenshot shows the Docker Playground interface in a web browser. The URL is 'labs.play-with-docker.com/p/cdugur63tccg00cslge0?cdugur63\_cdugv0e0qau0009mj06g'. The interface displays a timer at 03:57:46 and a 'CLOSE SESSION' button. Below the timer, there are 'Instances' and 'ADD NEW INSTANCE' options. A list of instances shows one instance named 'node1' with IP '192.168.0.28'. The main panel shows the details of the selected instance, including its IP, memory, CPU, and SSH access. The terminal window shows the execution of the 'docker run' command to pull and run the 'shashisekar/assignment' image. The output shows the image being pulled from Docker Hub and the container running successfully, displaying 'Job Portal! Hello!!'.

```
cdugur63_cdugv0e0qau0009mj06g
IP: 192.168.0.28
Memory: CPU:
SSH: ssh ip172-18-0-52-cdugur63tccg00cslge0@direct.labs.play-
DELETE EDITOR

#####
[root@1] (local) root@192.168.0.28 ~
$ docker run -dp 7000:5000 shashisekar/assignment
Unable to find image 'shashisekar/assignment:latest' locally
latest: Pulling from shashisekar/assignment
a8ca11554f0e: Pull complete
e4e46864aba2: Pull complete
c85a0be79bfb: Pull complete
195ea6a58ca8: Pull complete
157f16e0a00c: Pull complete
884b144bec2b: Pull complete
1c469643b609: Pull complete
4c0ac982aa89: Pull complete
049db2c7eb8a: Pull complete
8c7df8922040: Pull complete
376184ae62bc: Pull complete
285d896c6db4: Pull complete
Digest: sha256:readc049d37e1276ec0dc42fe650a9f0f8e836976c14730c0ebc495104a2ac85d
Status: Downloaded newer image for shashisekar/assignment:latest
9a095ee1e4d18e02ff64a33db10f66c2517c8c3a943be2e189cd01ccdf2e6
[root@1] (local) root@192.168.0.28 ~
$
```

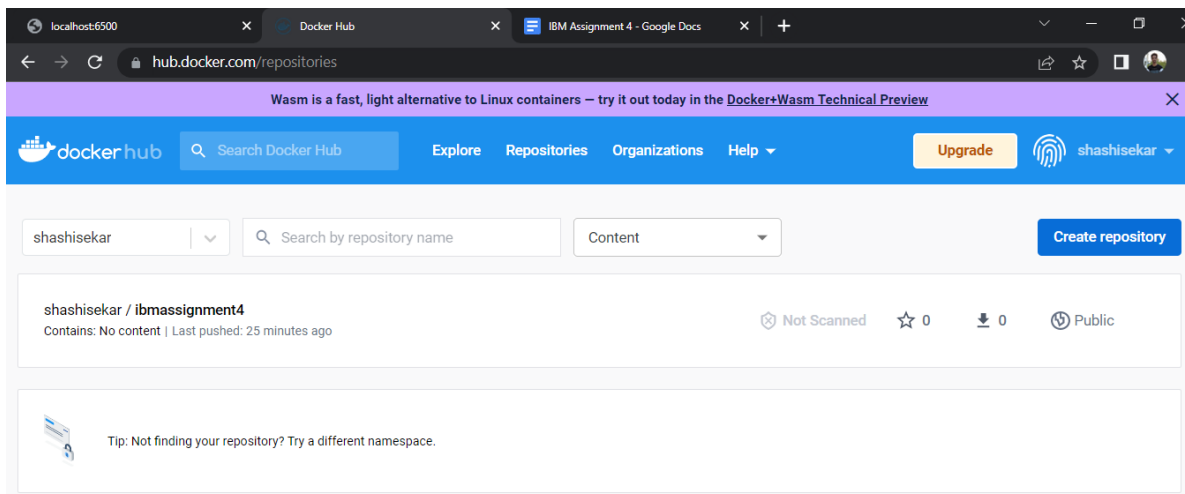
## Testing the Output on PORT 7000



The screenshot shows a web browser window with the address bar displaying 'ip172-18-0-52-cdugur63tccg00cslge0-7000.direct.labs.play-with-docker.com'. The page content shows the output of the application: 'Job Portal! Hello!!'.

```
Job Portal! Hello!!
```

## Docker Hub Repository



### 3. Create an IBM container registry and deploy helloworld app on jobportalapp

#### Login to IBM Cloud

```
PS C:\Users\Mahjabeen\Desktop\Flask_Prac\CLOUD> ibmcloud login -a https://cloud.ibm.com
API endpoint: https://cloud.ibm.com
```

```
Email> 2019103609@smartinternz.com
```

```
Password>
```

```
Authenticating...
```

```
OK
```

```
Targeted account Shashi D's Account (3dd395deaa66432a8f1a01f81faa00c6)
```

```
Select a region (or press enter to skip):
```

1. au-syd
2. in-che
3. jp-osa
4. jp-tok
5. kr-seo
6. eu-de
7. eu-gb
8. ca-tor

```
PS C:\Users\Mahjabeen\Desktop\Flask_Prac\CLOUD> ibmcloud plugin install container-registry -v 0.1.584
Looking up 'container-registry' from repository 'IBM Cloud'...
Plug-in 'container-registry[cr] 0.1.584' found in repository 'IBM Cloud'
Attempting to download the binary file...
 11.53 MiB / 11.53 MiB [=====] 100.00% 5s
12090368 bytes downloaded
Installing binary...
OK
Plug-in 'container-registry 0.1.584' was successfully installed into C:\Users\Mahjabeen\.bluemix\plugins\container-registry. Use 'C:\Program Files\IBM\Cloud\bin\ib
mcloud.exe plugin show container-registry' to show its details.
PS C:\Users\Mahjabeen\Desktop\Flask_Prac\CLOUD> ibmcloud cr region-set global

Warning: plug-in 'container-registry 0.1.584' may have compatibility issue with current version of IBM Cloud CLI.
Use 'C:\Program Files\IBM\Cloud\bin\ibmcloud.exe plugin update container-registry' to upgrade the plug-in.
Use 'C:\Program Files\IBM\Cloud\bin\ibmcloud.exe config --check-version=false' to disable update check.

The region is set to 'global', the registry is 'icr.io'.
OK
```

## Adding Namespace

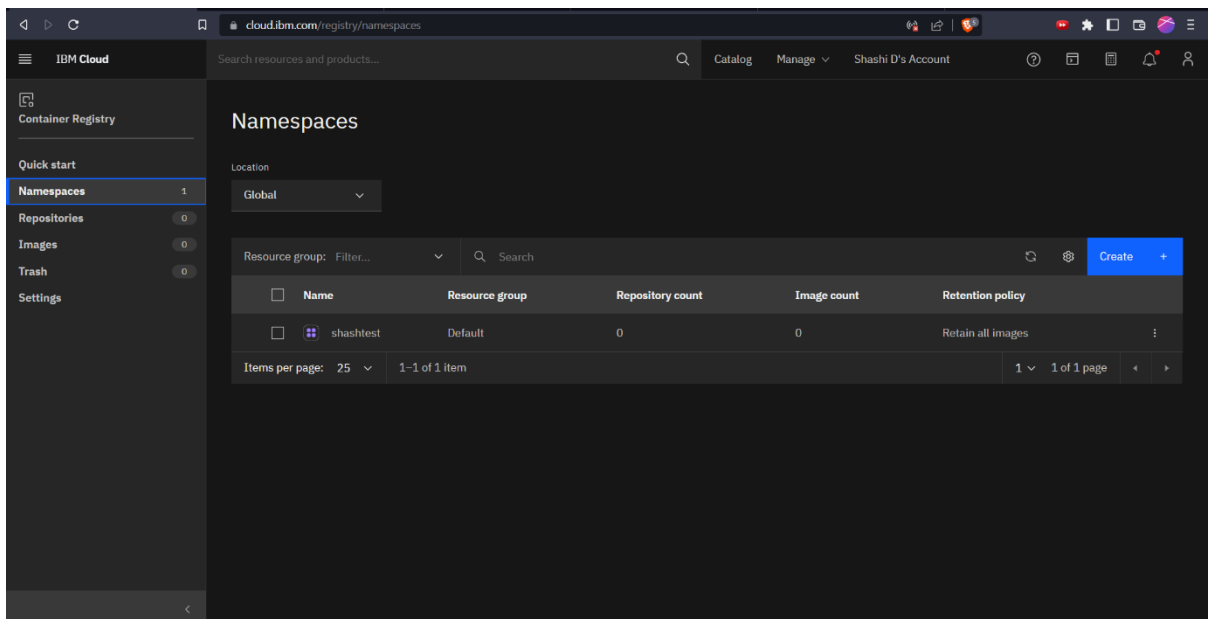
```
PS C:\Users\Mahjabeen\Desktop\Flask_Prac\CLOUD> ibmcloud cr namespace-add shashtest
No resource group is targeted. Therefore, the default resource group for the account ('Default') is targeted.

Adding namespace 'shashtest' in resource group 'Default' for account Shashi D's Account in registry icr.io...

Successfully added namespace 'shashtest'

OK
```

## Namespaces on Container Registry



The screenshot shows the IBM Cloud Container Registry interface. On the left, there's a sidebar with 'Namespaces' selected. The main area is titled 'Namespaces' and shows a table with one namespace named 'shashtest' in the 'Default' resource group. The table has columns for Name, Resource group, Repository count, Image count, and Retention policy. The 'shashtest' namespace has 0 repositories and 0 images, with a retention policy of 'Retain all images'.

Name	Resource group	Repository count	Image count	Retention policy
shashtest	Default	0	0	Retain all images

```
PS C:\Users\Mahjabeen\Desktop\Flask_Prac\CLOUD> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
PS C:\Users\Mahjabeen\Desktop\Flask_Prac\CLOUD> ibmcloud cr login
Logging 'docker' in to 'icr.io'...
Logged in to 'icr.io'.
```

OK

```
PS C:\Users\Mahjabeen\Desktop\Flask_Prac\CLOUD> docker tag shashisekar/assignment:latest icr.io/shashtest/shashrepo:1
```

## Pushing the image into the repository

```
PS C:\Users\Mahjabeen\Desktop\Flask_Prac\CLOUD> docker push icr.io/shashtest/shashrepo:1
The push refers to repository [icr.io/shashtest/shashrepo]
326164d3388f: Layer already exists
566c18a65fa9: Pushed
a2d41df22a3b: Layer already exists
345c9e42b8e4: Pushed
24bf8dd8c4a6: Layer already exists
18bbb218c890: Pushed
e6e9854ca999: Layer already exists
397a239a053b: Pushed
89c3244a87b2: Pushed
80231db1194c: Pushed
f1c1f2298584: Pushed
ccba29d69370: Pushed
1: digest: sha256:eadc649d37e1276ee0dc42fe650a9f0f8c836976c14730c0ebc495104a2ac85d size: 2843
```

## Image List

```
PS C:\Users\Mahjabeen\Desktop\Flask_Prac\CLOUD> ibmcloud cr image-list
Listing images...

Repository          Tag    Digest          Namespace    Created      Size    Security status
icr.io/shashtest/shashrepo  1      eadc649d37e1    shashtest    20 hours ago  363 MB  3 Issues

OK
```

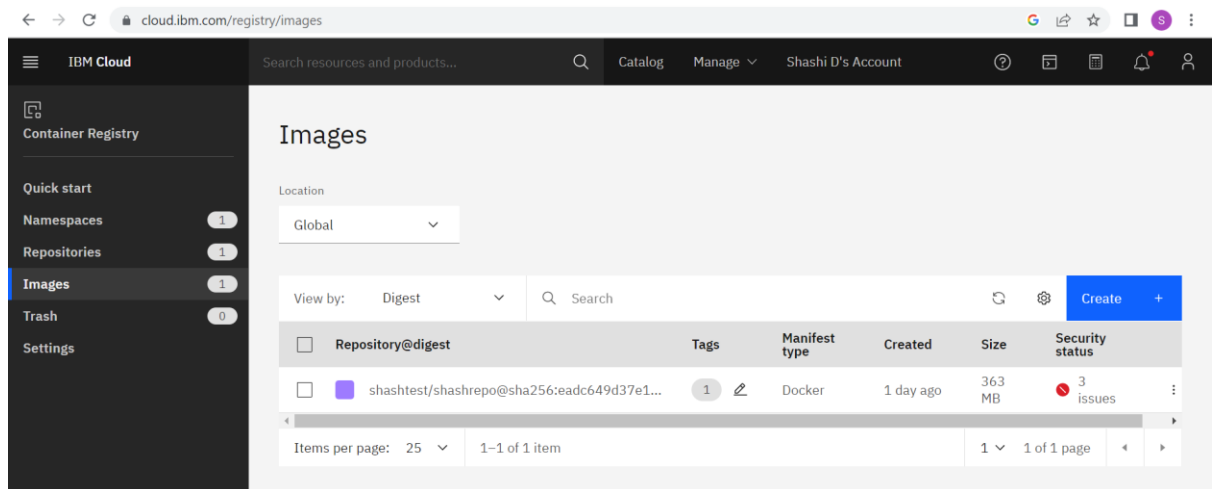
## Repositories - shashrepo

The screenshot shows the IBM Cloud Container Registry web interface. The left sidebar contains navigation links: Container Registry, Quick start, Namespaces (1), Repositories (1), Images (1), Trash (0), and Settings. The main content area is titled 'Repositories' and shows a list of repositories. The 'shashrepo' repository is listed with the following details:

Name	Image count	Namespace	Last updated
shashrepo icr.io/shashtest/shashrepo	1	shashtest	1 day ago

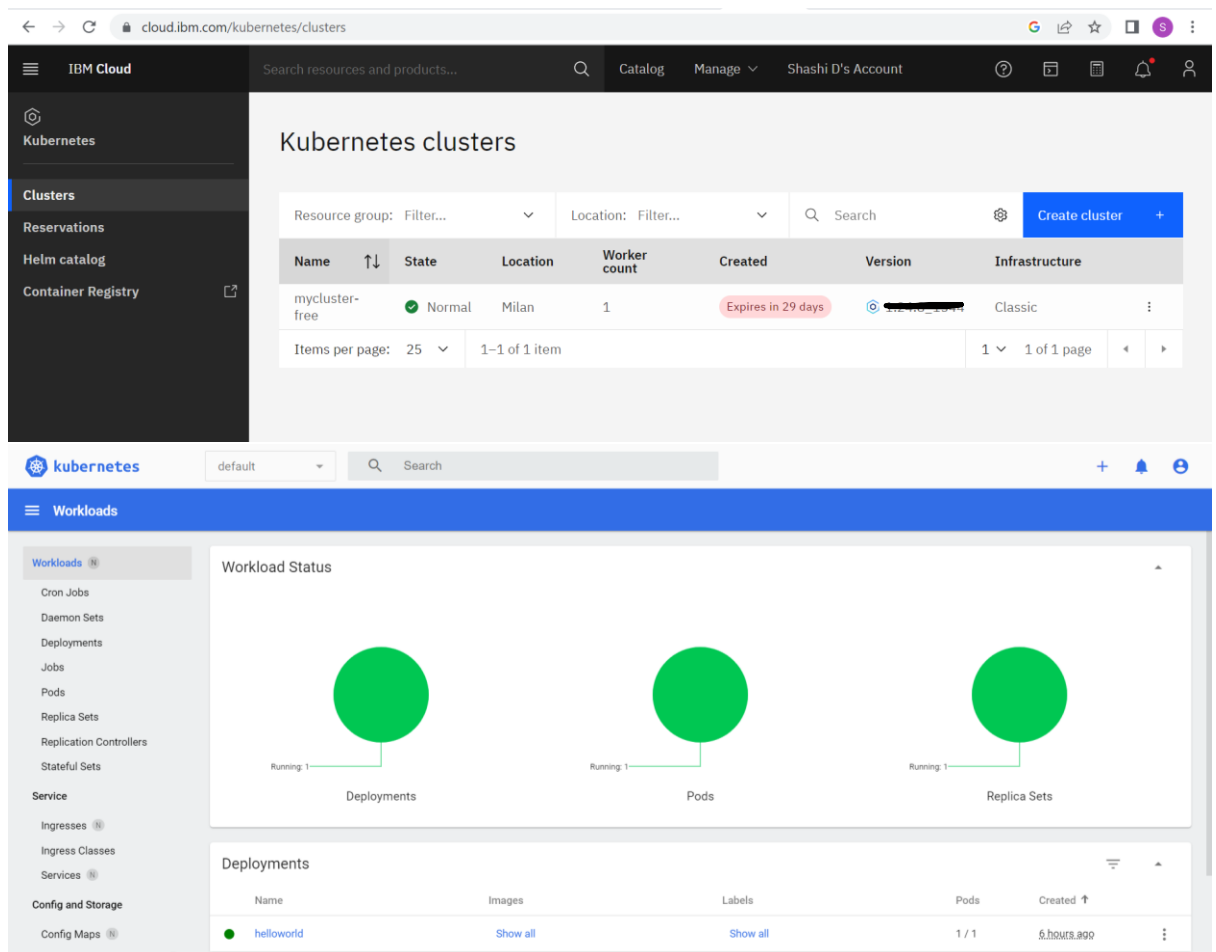
At the bottom of the repository list, there is a pagination bar showing 'Items per page: 25' and '1-1 of 1 item'.

## Images



4. Create a Kubernetes cluster in IBM cloud and deploy the same app to run in nodeport.

## Kubernetes Cluster





eu-de.containers.cloud.ibm.com/kubeproxy/clusters/cduhomff0h4a61rjae60/service/#/workloads?namespace=default

kubernetes default Search

Workloads

Workloads N

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

Service

- Ingresses N
- Ingress Classes
- Services N

Config and Storage

- Config Maps N
- Persistent Volume Claims N
- Secrets N

Running: 1

Deployments

Running: 1

Pods

Running: 1

Replica Sets

Deployments

Name	Images	Labels	Pods	Created ↑
helloworld	Show all	Show all	1 / 1	6 hours ago

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
helloworld-8b8cf6bd9-2c2z9	Show all	Show all	10.144.222.48	Running	0	1.00m	21.34Mi	6 hours ago

Replica Sets

Name	Images	Labels	Pods	Created ↑
helloworld-8b8cf6bd9	Show all	Show all	1 / 1	6 hours ago

## Deployments

eu-de.containers.cloud.ibm.com/kubeproxy/clusters/cduhomff0h4a61rjae60/service/#/deployment?namespace=default

kubernetes default Search

Workloads > Deployments

Workloads N

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

Service

- Ingresses N
- Ingress Classes
- Services N

CPU Usage

Memory Usage

Deployments

Name	Images	Labels	Pods	Created ↑
helloworld	Show all	Show all	1 / 1	7 hours ago

## POD

eu-de.containers.cloud.ibm.com/kubeproxy/clusters/cduhomff0h4a61rjae60/service/#/pod/default/helloworld-8b8cf6bd9-2c2z9?namespace=default

kubernetes default Search

Workloads > Pods > helloworld-8b8cf6bd9-2c2z9

Workloads N

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

Service

- Ingresses N
- Ingress Classes
- Services N

Config and Storage

- Config Maps N
- Persistent Volume Claims N

CPU Usage

Memory Usage

Metadata

Name	Namespace	Created	Age	UID
helloworld-8b8cf6bd9-2c2z9	default	Nov 23, 2022	7 hours ago	73202ef3-89c7-4974-bf10-bfa3e940ab56

Labels

k8s-app: helloworld pod-template-hash: 8b8cf6bd9

Annotations

cni.projectcalico.org/containerID cni.projectcalico.org/podIP: 172.30.55.75/32 cni.projectcalico.org/podIPs: 172.30.55.75/32 kubernetes.io/psp: ibm-privileged-psp