

## Assignment 4

Assignment date	06 November 2022
Student name	Hari Prasath S
Student roll no	711719104030
Team ID	PNT2022TMID31583


### Question 1:

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

### Solution 1:


```
docker pull uifd/ui-for-docker
```

```
docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
```


 docker hub

Explore Repositories Organizations Help

Upgrade

 711719104

Explore uifd/ui-for-docker



**uifd/ui-for-docker** ☆

By [uifd](#) • Updated 6 years ago

A web interface for Docker, formerly known as DockerUI. Depreciated, use Portainer for new features.

Image

Pulls

Overview Tags

### UI For Docker

This repo is deprecated. Development continues at: [portainer/portainer](#)

chat on [gitter](#)

UI For Docker is a web interface for the Docker Remote API. The goal is to provide a pure client side implementation so it is effortless to connect and manage docker.

**Goals**

- Minimal dependencies - I really want to keep this project a pure html/js app.

**Docker Pull Command**

```
docker pull uifd/ui-for-docker
```

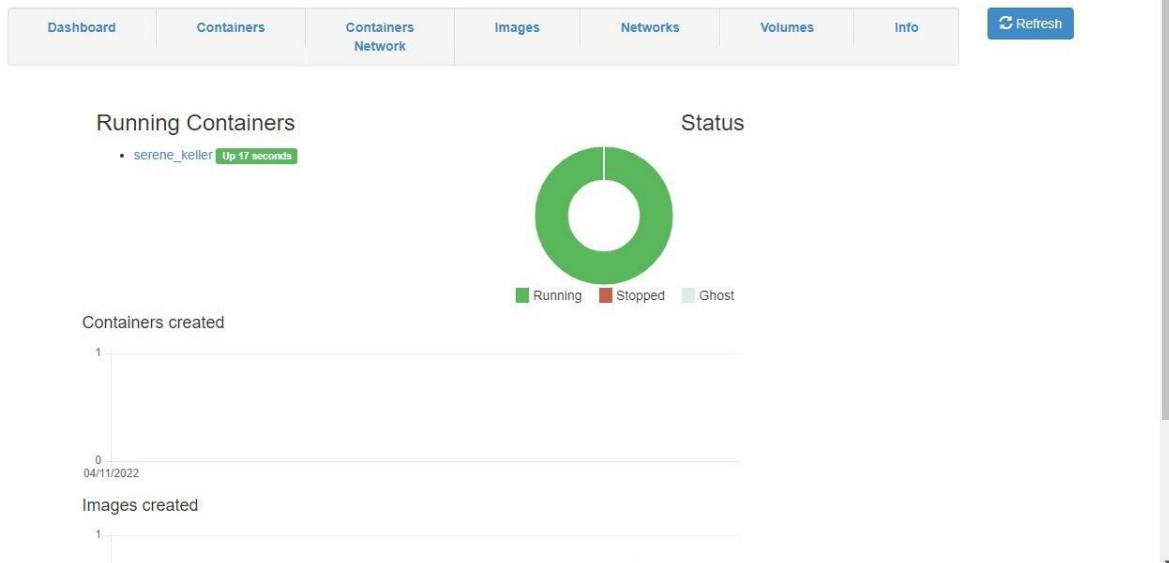
## Docker playground:

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a timer at 03:44:55, a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button. Below that, a list of instances shows '192.168.0.8 node1'. The main area displays the instance name 'cdoseje3\_cdosevm3tccg00aokbk0' and its IP '192.168.0.8'. There are buttons for 'OPEN PORT', 'DELETE', and 'EDITOR'. The terminal window shows the following commands and output:

```
[node1] (local) root@192.168.0.8 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    feb5d9fea6a5  13 months ago  13.3kB
[node1] (local) root@192.168.0.8 ~
$ docker run -d -p 9000:9000 --privileged -v/var/run/docker.sock hello-world
b56c7d9e14d6aab9e6606acc7e37f97619b33615b250bd9a9664ecfec9dec7f7
[node1] (local) root@192.168.0.8 ~
$ docker pull hello-world:latest
latest: Pulling from library/hello-world
Digest: sha256:faa03e786c97f07ef34423fccceec2398ec8a5759259f94d99078f264e9d7af
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
[node1] (local) root@192.168.0.8 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    feb5d9fea6a5  13 months ago  13.3kB
[node1] (local) root@192.168.0.8 ~
$
```

## Docker UI:

### UI For Docker



## Question 2:

Create a docker file for the job portal app or hello world app and deploy it in docker desktop app.

## Solution 2:

### DockerFile

```
1 FROM python:3.10.6
2 WORKDIR /app
3 COPY requirements.txt ./
4 RUN pip install -r requirements.txt
5 COPY . .
6 EXPOSE 5000
7 CMD ["python", ". /app.py"]
8
```

### Bulid Docker image

```
C:\Windows\System32\cmd.exe

E:\Study materials\Sem 7\IBM\Exercise\Assignment4>docker build -t hello-world .
[+] Building 160.4s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 194B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/python:3.8                    5.0s
[1/5] FROM docker.io/library/python:3.8@sha256:089d758211770a2dd03ecc4b10a8d851f6f77ef3f1e3f3620d8519190b8aa1d5 149.9s
=> => resolve docker.io/library/python:3.8@sha256:089d758211770a2dd03ecc4b10a8d851f6f77ef3f1e3f3620d8519190b8aa1d5 0.0s
=> => sha256:9a00972ffecdc8c17c25b21573681851f092a054f57ccd7eb43937e1a47114480 0.56kB / 0.56kB 0.0s
=> => sha256:17c9e6141fdb3387e5a1c07d4f9bba05ac1498e96029fa3ea55470d4504f7770 55.05MB / 55.05MB 65.2s
=> => sha256:4edcd8587e6c18412817019074f5e04a8ede4e2fc89d06af13df3f80d78a70d 10.88MB / 10.88MB 8.7s
=> => sha256:089d758211770a2dd03ecc4b10a8d851f6f77ef3f1e3f3620d8519190b8aa1d5 1.06kB / 1.06kB 0.0s
=> => sha256:254101fcf737ef09a912ce9ad7488801a01e0a35bffc5e7d6bb86d0b0e1c3f 2.22kB / 2.22kB 0.0s
=> => sha256:de4a4c6cae8801bb0b7377e10220a914da403bc93fa79663cbf2dcf1800b6f1 5.16MB / 5.16MB 18.3s
=> => sha256:a7969cfff7b46e6a91291fd70b19ecbe93c03ea4ded0d14042aebc4c0c4211a43 54.59MB / 54.59MB 47.5s
=> => sha256:74fbfde6af91271fb88f0a1716224dce5c0beead3609943792a9cb6ba4d6d3d 196.87MB / 196.87MB 133.3s
=> => sha256:16fe51aed899f36017fe42b598b1a622b29ebe8c3622e92e13df14578825eb37 6.29MB / 6.29MB 53.8s
=> => sha256:2b979a731384cf50dac8fd255d381b70028d67b69b45c1a2b6c3ea10b92636d4 17.39MB / 17.39MB 68.0s
=> => sha256:aa3c4359fdb43308669ae8ba78b2ebb713221ef3a3eca97f93590508f1506de1 234B / 234B 67.3s
=> => extracting sha256:17c9e6141fdb3387e5a1c07d4f9bba05ac1498e96029fa3ea55470d4504f7770 10.8s
=> => sha256:58700fbcfa0c82e5d24a9f76ba7748a194c4fdf7312a397800b4637f72ce91b6 2.89MB / 2.89MB 70.7s
=> => extracting sha256:de4a4c6cae8801bb0b7377e10220a914da403bc93fa79663cbf2dcf1800b6f1 1.3s
=> => extracting sha256:4edcd8587e6c18412817019074f5e04a8ede4e2fc89d06af13df3f80d78a70d 1.0s
=> => extracting sha256:a7969cfff7b46e6a91291fd70b19ecbe93c03ea4ded0d14042aebc4c0c4211a43 13.1s
=> => extracting sha256:74fbfde6af91271fb88f0a1716224dce5c0beead3609943792a9cb6ba4d6d3d 13.6s
=> => extracting sha256:16fe51aed899f36017fe42b598b1a622b29ebe8c3622e92e13df14578825eb37 0.4s
=> => extracting sha256:2b979a731384cf50dac8fd255d381b70028d67b69b45c1a2b6c3ea10b92636d4 1.1s
=> => extracting sha256:aa3c4359fdb43308669ae8ba78b2ebb713221ef3a3eca97f93590508f1506de1 0.0s
=> => extracting sha256:58700fbcfa0c82e5d24a9f76ba7748a194c4fdf7312a397800b4637f72ce91b6 0.4s
=> [internal] load build context                                                0.0s
=> => transferring context: 1.15kB                                              0.0s
=> [2/5] WORKDIR /app                                                         0.4s
=> [3/5] ADD . /app                                                            0.1s
=> [4/5] COPY requirements.txt /app                                           0.0s
=> [5/5] RUN python3 -m pip install -r requirements.txt                       3.0s
=> => exporting to image                                                         0.2s
=> => exporting layers                                                           0.2s
=> => writing image sha256:f68fcdce5bb665f00e8f47bc4d137a4f7e0533348402c5bfdad71121d7d43f63 0.0s
=> => naming to docker.io/library/hello-world                                0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

## Deploy it on Docker hub

**Docker Desktop** Upgrade plan

Containers Images Volumes Dev Environments **BETA**

Extensions **BETA** Add Extensions

**Images on disk** Last refresh: Never 1 images Refresh to see disk usage **Clean up**

**Images** [Give feedback](#)

**LOCAL** REMOTE REPOSITORIES

Search

☐ In use only

NAME ↑	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	f68fcdce5bb6	less than a minute ago	919.36 MB

RAM 3.66GB CPU 0.08% Connected to Hub v4.13.1

```
C:\Windows\System32\cmd.exe
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
E:\Study materials\Sem 7\IBM\Exercise\Assignment4>docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest f68fcdce5bb6 5 minutes ago 919MB
E:\Study materials\Sem 7\IBM\Exercise\Assignment4>docker login
Authenticating with existing credentials...
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.docker.com/go/access-tokens/
E:\Study materials\Sem 7\IBM\Exercise\Assignment4>docker tag hello-world itsmona14/hello-world
E:\Study materials\Sem 7\IBM\Exercise\Assignment4>docker push itsmona14/hello-world
Using default tag: latest
The push refers to repository [docker.io/itsmona14/hello-world]
373eb5cf4cab: Pushed
1e505dc1de5e: Pushed
098c85cb75c5: Pushed
ded8299b8f1a: Pushed
1fe0699af9f7: Mounted from library/python
156568a71809: Mounted from library/python
5fca8a94d542: Mounted from library/python
6b183c62e3d7: Mounted from library/python
882fd36bfd35: Mounted from library/python
d1dec9917839: Mounted from library/python
d38adf39e1dd: Mounted from library/python
4ed121b04368: Mounted from library/python
d9d07d703dd5: Mounted from library/python
latest: digest: sha256:46ff91edc98aaa5d7fff51ba708b6498af3c4f64612d9a990bf437497555fd82 size: 3049
E:\Study materials\Sem 7\IBM\Exercise\Assignment4>
```

## Tested it using Docker playground

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a timer '03:09:45', a 'CLOSE SESSION' button, and a list of instances. The main area displays details for a container named 'cdi0ji60\_cdi18b63tccg00fmtsog' with IP '192.168.0.13'. It shows memory usage at 27.73% (1.083GiB / 3.906GiB) and CPU usage at 0.16%. An SSH button is available. Below the details, there's a terminal window showing the container's output, which includes the Docker command used to run the container and the Flask application's startup logs.

```
1bd231713cc1: Pull complete
59ebc78c27fb: Pull complete
72f61f026f6a: Pull complete
b8ba28eaa452: Pull complete
Digest: sha256:0036fe1456627bba779e865ba4793212e8332e6835b48c6b5814784adb70c46f
Status: Downloaded newer image for itsmona14/hello-world:latest
docker.io/itsmona14/hello-world:latest
(node1) (local) root@192.168.0.13 ~
$ docker run -p 5000:5000 itsmona14/hello-world
 * 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 server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
Press CTRL+C to quit
172.18.0.1 - - [03/Nov/2022 19:24:35] "GET / HTTP/1.1" 200 -
```

## Question 3:

Create an IBM container registry and deploy helloworld app or job portal app.

## Solution 3:

My image link: [au.icr.io/hello-world-app/hello-world](https://au.icr.io/hello-world-app/hello-world)

The screenshot shows a Windows Command Prompt window with the following commands and output:

```
C:\Users\Monashree>ibmcloud plugin install container-registry
Looking up 'container-registry' from repository 'IBM Cloud'...
Plug-in 'container-registry[cr] 1.0.2' found in repository 'IBM Cloud'
Attempting to download the binary file...
11.90 MiB / 11.90 MiB [=====] 100.00% 5s
12476416 bytes downloaded
Installing binary...
OK
Plug-in 'container-registry 1.0.2' was successfully installed into C:\Users\Monashree\bluemix\plugins\container-registry. Use 'ibmcloud plugin show container-registry' to show its details.

C:\Users\Monashree>ibmcloud login -a https://cloud.ibm.com
API endpoint: https://cloud.ibm.com

Email> 2019115055@smartinternz.com

Password>
Authenticating...
OK

Targeted account Monashree K's Account (302198646cc145ea8bc880cfb8a0d15d)

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
9. us-south
10. us-east
11. br-sao
Enter a number> 9
882fd36bfd35: Pushing [=====] 110.5MB/529MB
d1dec9917839: Pushing [=====] 79.9MB/152MB

API endpoint: https://cloud.ibm.com
d9d07d703dd5: Pushing [=====] 67.45MB/124.1MB
d1dec9917839: Pushing [=====] 69.67MB/152MB
```

```

C:\Windows\System32\cmd.exe - docker run -p 5000:5000 au.icr.io/hello-world-app/hello-world

E:\Study materials\Sem 7\IBM\Exercise\Assignment4>docker tag hello-world au.icr.io/hello-world-app/hello-world

E:\Study materials\Sem 7\IBM\Exercise\Assignment4>docker push au.icr.io/hello-world-app/hello-world
Using default tag: latest
The push refers to repository [au.icr.io/hello-world-app/hello-world]
492bcd5cc009: Pushed
006e0938fc5e: Pushed
40b28ce8724f: Pushed
402de03c8533: Pushed
f5d161bbe139: Pushed
1569e0d95ca6: Pushed
09e08da15d0c: Pushed
6b183c62e3d7: Mounted from hello-world-app/hello-world-app
882fd36bf35: Mounted from hello-world-app/hello-world-app
d1dec9917839: Mounted from hello-world-app/hello-world-app
d38adf39e1dd: Mounted from hello-world-app/hello-world-app
4ed121b04368: Mounted from hello-world-app/hello-world-app
d9d07d703dd5: Mounted from hello-world-app/hello-world-app
latest: digest: sha256:0036fe1456627bba779e865ba4793212e8332e6835b48c6b5814784adb70c46f size: 3049

E:\Study materials\Sem 7\IBM\Exercise\Assignment4>ibmcloud cr image-list
Listing images...

Repository          Tag      Digest          Namespace      Created      Size      Security status
au.icr.io/hello-world-app/hello-world  latest  0036fe145662    hello-world-app  12 minutes ago  350 MB    -

OK

E:\Study materials\Sem 7\IBM\Exercise\Assignment4>docker run -p 5000:5000 au.icr.io/hello-world-app/hello-world
* 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 server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
172.17.0.1 - - [03/Nov/2022 19:35:58] "GET / HTTP/1.1" 200 -

```

Container Registry

- Quick start
- Namespaces 1
- Repositories 1**
- Images 1
- Trash 1
- Settings

## Repositories

Location: Sydney

Search 🔍 🔗 🔖 Create +

	Name	Image count	Namespace	Last updated
📄	<div>hello-world</div> <div>au.icr.io/hello-world-app/hello-world</div>	1	hello-world-app	15 minutes ago

Items per page: 25 1-1 of 1 item

1 1 of 1 page

#### Question 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 4:

```
apiVersion: v1
kind: Service
metadata:
  name: hello-world-deployment
spec:
  ports:
    - port: 5000
      targetPort: 5000
  selector:
    app: hello-world
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-world-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello-world
  template:
    metadata:
      labels:
        app: hello-world
    spec:
      containers:
        - name: hello-world
          image: au.icr.io/hello-world-app/hello-world
          imagePullPolicy: Always
          ports:
            - containerPort: 5000
```

The screenshot shows the IBM Cloud Kubernetes Dashboard for a cluster named 'mycluster-free'. The cluster is in a 'Normal' state and expires in 29 days. The dashboard provides an overview of the cluster's status and details.

**Overview:**

- Node status:** 1 of 1 nodes are in a 'Normal' state.
- Add on status:** 0 of 0 add-ons are installed.
- Master status:** The master node is in a 'Normal' state.
- Ingress status:** The ingress controller is in an 'Unknown' state.

**Details:**

- Cluster ID:** cd14j3cfe0a6mchav0k1g
- Version:** 1.24.7\_1542
- Infrastructure:** Cioctic
- Zones:** Milan 01
- Created:** 04/11/2022, 01:12
- Resource group:** Default
- Image security enforcement:** Enable

