

ASSIGNMENT – 4

Assignment Date	28 October 2022
Student Name	Surya T
Student Roll Number	921319104203
Maximum Marks	2 Mark

Question-1

Pull an image from Dockers hub and run it in Dockers playground.

SOLUTION:

STEP: 1

Login to Dockers hub and get an image

STEP: 2

- Open Dockers playground
- Login with Dockers
- Create new instance

STEP: 3

In the command prompt run the following:

```
$ docker pull hello-world
$ docker run hello-world
```

```

[rode1] (local) root@192.168.0.8 ~
$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:18a657d0cc1c7d0478a3fba8b7eb4918bbe25968d3e1b0e9ebfa71caddbc346
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
[rode1] (local) root@192.168.0.8 ~
$ docker run hello-world

Hello From Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (and54)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

[rode1] (local) root@192.168.0.8 ~
$

```

QUESTION 2:

Create a Dockers file and deploy it in Dockers desktop application SOLUTION:

STEP: 1

- Create a flask application
- Create a Dockerfile in the same folder

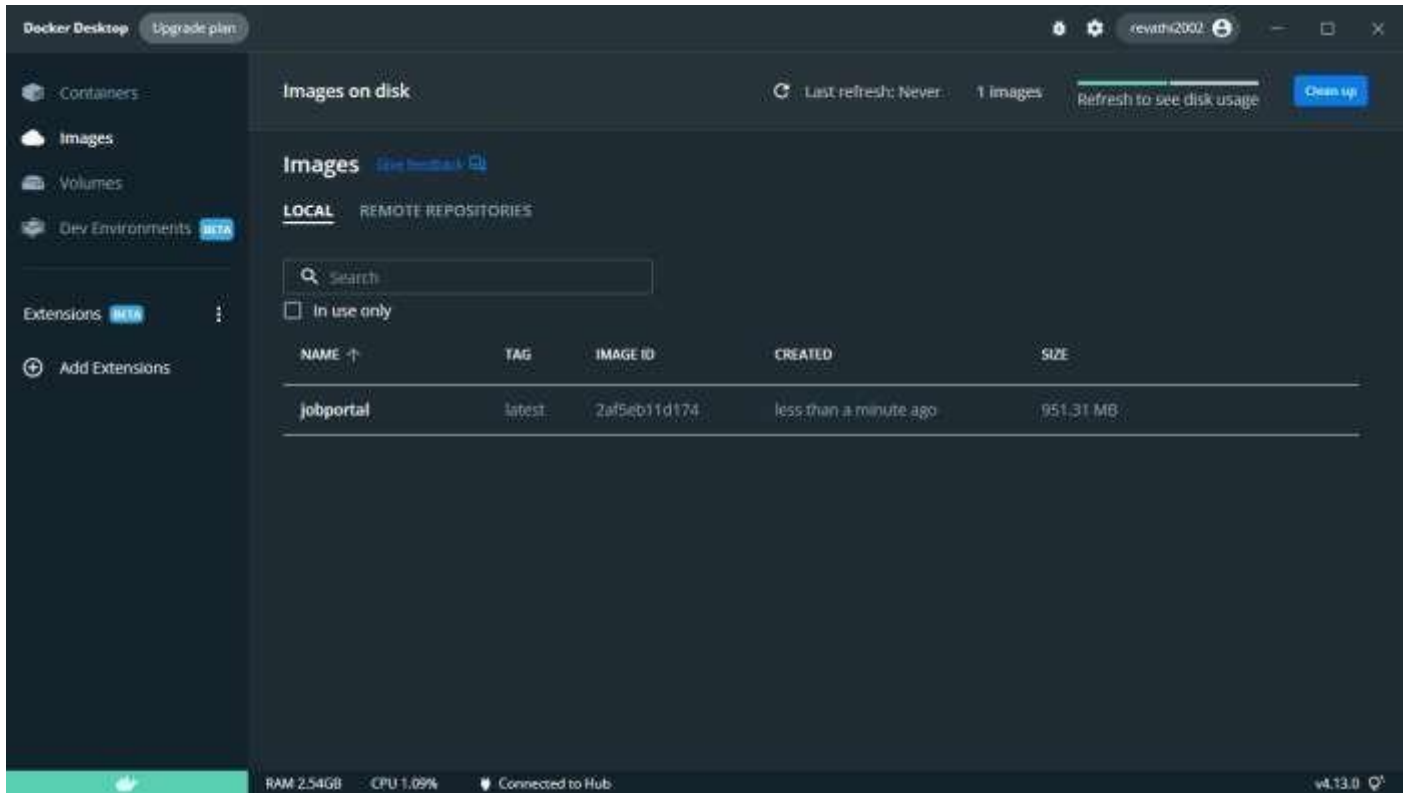
STEP: 2

Run the following commands to deploy it in docker desktop

```

$ docker build -t jobportal
$ docker image ls

```

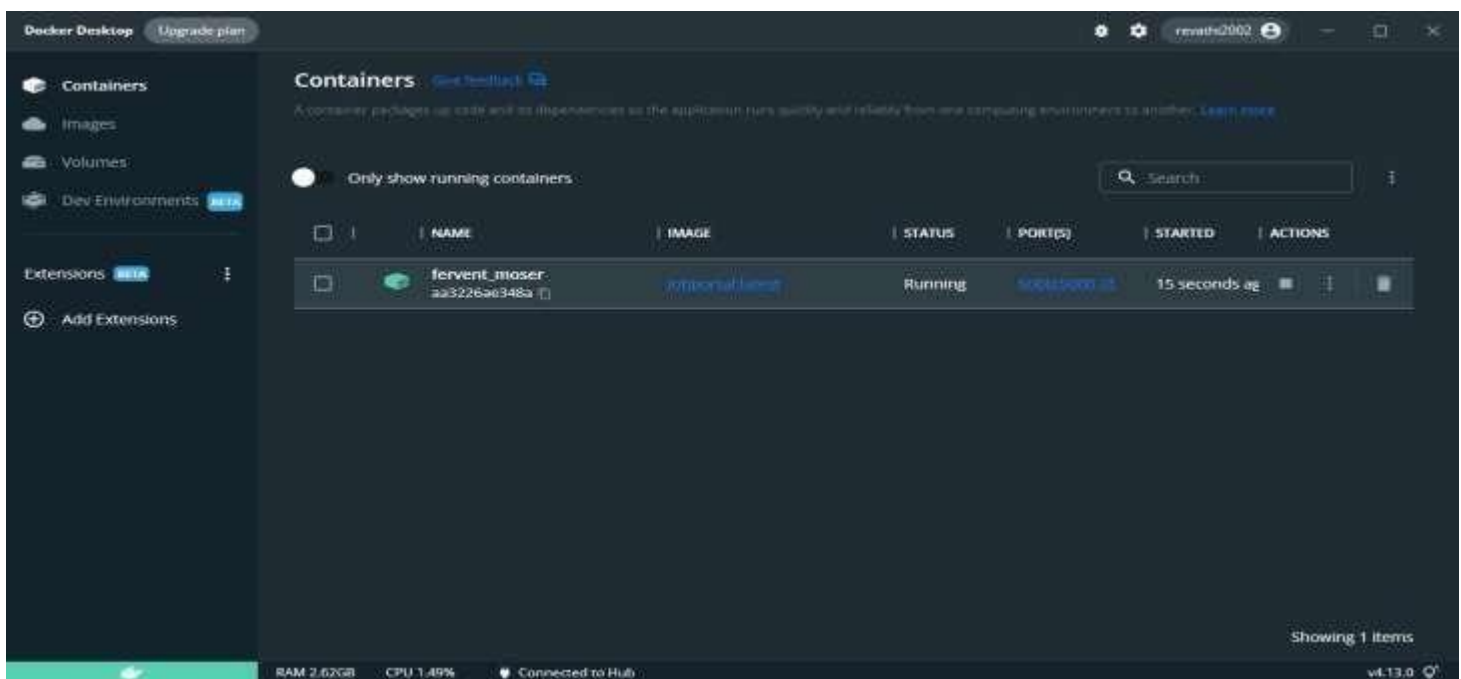


\$ **docker container run -p 5000:5000 jobportal**

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
PS D:\Ibse project\Assignments\Team Lead\Assignment 4\Docker_Desktop> docker container run -p 5000:5000 jobportal
* Serving Flask app "app"
* Debug mode: on
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://172.17.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 221-899-559
172.17.0.1 - - [25/Oct/2022 11:30:28] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [25/Oct/2022 11:30:29] "GET /static/css/style.css HTTP/1.1" 200 -
172.17.0.1 - - [25/Oct/2022 11:30:29] "GET /static/images/1.png HTTP/1.1" 200 -
172.17.0.1 - - [25/Oct/2022 11:30:30] "GET /favicon.ico HTTP/1.1" 404 -
PS D:\Ibse project\Assignments\Team Lead\Assignment 4\Docker_Desktop> docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
jobportal latest 2af5eb11d174 7 minutes ago 951MB
PS D:\Ibse project\Assignments\Team Lead\Assignment 4\Docker_Desktop> docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS D:\Ibse project\Assignments\Team Lead\Assignment 4\Docker_Desktop>
```



\$ docker container ls



QUESTION 3:

Create an IBM container registry and deploy hello-world-app or job-portal-app SOLUTION:

QUESTION 4:

```
PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> ibmcloud plugin install container-registry --c 'IBM Cloud'
Looking up 'container-registry' from repository 'IBM Cloud'...
Plug-in 'container-registry[cr] 1.8.2' found in repository 'IBM Cloud'
Attempting to download the binary file...
11.90 MB / 11.90 MB [-----] 100.00% 1m37s
12476416 bytes downloaded

Installing binary...
OK
Plug-in 'container-registry 1.8.2' was successfully installed into C:\Users\ELCOP\ibm\plugins\container-registry, the 'C:\Program Files\IBM\Cloud\bin\ibmcloud.exe' plugin show container-registry' to show its details.
PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> ibmcloud cr region-set global
FAILED
You are not logged in to IBM Cloud.
Log in by running the 'C:\Program Files\IBM\Cloud\bin\ibmcloud.exe login' command.

PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> ibmcloud login -d https://cloud.ibm.com
API endpoint: https://cloud.ibm.com

Email> revathi3007@gmail.com

Password>
Authenticating...
Credentials were rejected.
Code: B00D0602E, message: The credentials you entered for the user 'revathi3007@gmail.com' are incorrect.

Password>
Authenticating...
Credentials were rejected.
Code: B00D0602E, message: The credentials you entered for the user 'revathi3007@gmail.com' are incorrect.

PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> ibmcloud login -d https://cloud.ibm.com
API endpoint: https://cloud.ibm.com

Email> 90061918665@wartintara.co

Password>
PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> ibmcloud cr region-set global
The region is set to 'global', the registry is 'icr.io'.

No resource group is targeted, therefore, the default resource group for the account ('Default') is targeted.

Adding namespace 'jobportal' in resource group 'Default' for account Revathi R's account in registry icr.io...

The requested namespace is already in use in registry 'icr.io'.
Choose a different namespace.

No resource group is targeted, therefore, the default resource group for the account ('Default') is targeted.

Adding namespace 'jobspace1' in resource group 'Default' for account Revathi R's account in registry icr.io...

Successfully added namespace 'jobspace1'

OK
PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> ibmcloud cr login
Logging 'docker' in to 'icr.io'...
logged in to 'icr.io'.

OK
PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
20c29710123e: Pull complete
Digest: sha256:13a0578cc1c788578a3f8e8b75b918ba2596d33e8b9d6fa71caddbc36
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> docker tag hello-world icr.io/jobspace1/hello-world:v1
PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
jobportal            latest      2af5eb11d124   59 minutes ago 95.9B
hello-world          latest      f05d0fead65   13 months ago 13.3kB
icr.io/jobspace1/hello-world v1          f05d0fead65   13 months ago 13.3kB
PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> docker push icr.io/jobspace1/hello-world:v1
The push refers to repository [icr.io/jobspace1/hello-world]
a07e0b8a5ff: Pushed
v1: digest: sha256:f54a58bcaac5a25d79bae155d22b37be11d8bae778b9c4b72f13d8e4 size: 525
PS D:\Ibm project\Assignments\Team Lead\Assignment 4\IBM Container_Registry> ibmcloud cr image-list
Listing images...

Repository          Tag         Digest          Namespace        Created        Size        Security status
icr.io/jobspace1/hello-world v1          f54a58bcaac     jobspace1        1 year ago     2.5 KB      -
```

IBM Cloud

Search resources and products...

Catalog

Manage

Revathi R's Account

Container Registry

Quick start

Namespaces

Repositories

Images

Trash

Settings

Repositories

Location

Global

Search

Create

Name	Image count	Namespace	Last updated
hello-world icr.io/jobspace1/hello-world	1	jobspace1	397 days ago
Digest	Manifest type	Tags	Created
f54a58bcaac	Docker	v1	397 days ago
Size	Security status		
2 KB	Unsupported OS		

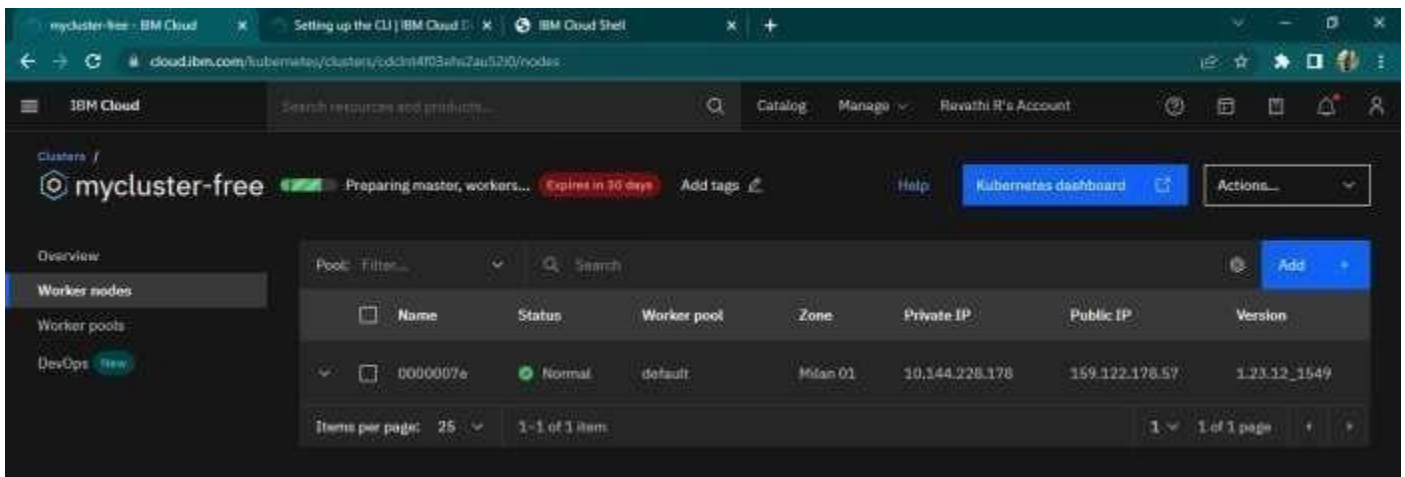
Items per page: 25 1-1 of 1 item

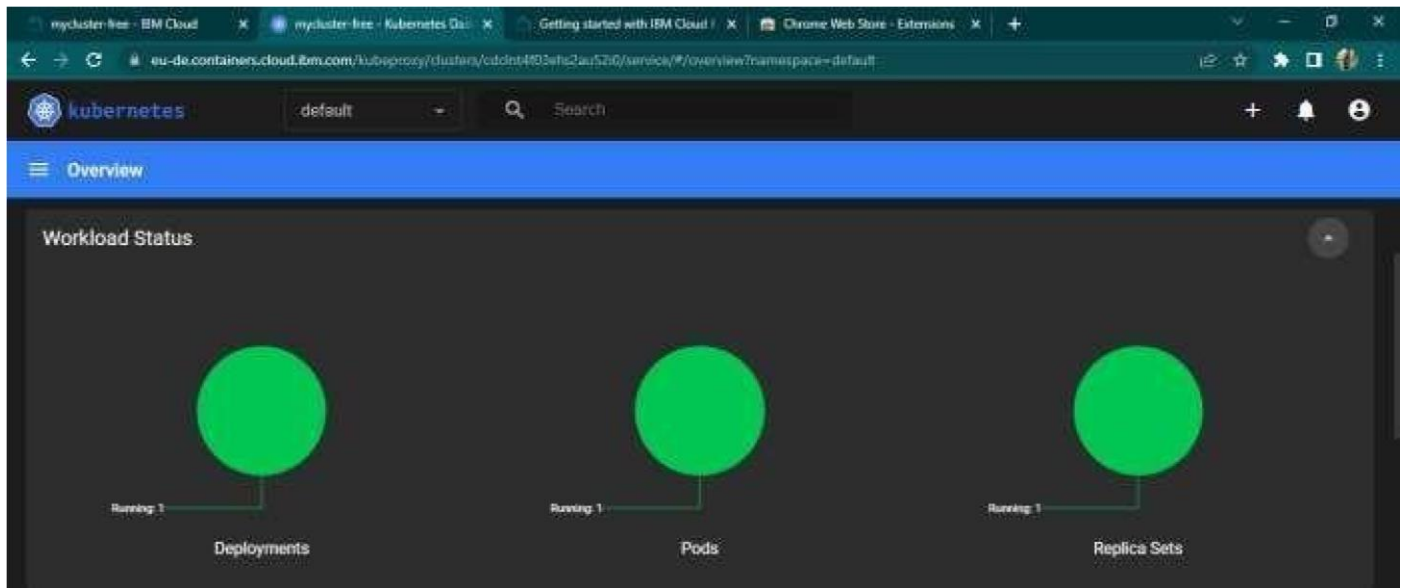
1 1 of 1 page

Create a Kubernetes cluster in IBM cloud and deploy hello-world-image or job-portal-image and also expose the same app to run in node-port.

SOLUTION:

1. Select your cluster from the cluster list to open the details for your cluster.
2. Click Kubernetes dashboard.
3. From the menu bar, click the Create new resource icon (+).
4. Select the Create from form tab.
 - a) Enter a name for your app, i.e **hello-world**.
 - b) Enter **websphere-liberty** for your container image.
 - c) Enter the number of pods for your app deployment, such as 1.
 - d) Leave the Service drop-down menu set to None.
5. Click Deploy. During the deployment, the cluster downloads the websphere-liberty container image from Docker Hub and deploys the app in your cluster.
6. Create a node port so that your app can be accessed by other users internally or externally. Because your cluster is a free cluster, you can only expose an app with a node port, not a load balancer or Ingress.
 - a) Click the Create new resource icon (+).
 - b) Copy the node port YAML from GitHub.
 - c) In the Create from input box, paste the node port YAML that you copied in the previous step.
 - d) Click Upload. The node port service is created.
7. From the menu, click Services, and note the TCP endpoint port of your liberty service in the node port range 30000 - 32767, i.e **liberty:30357 TCP**.
8. From the menu, click Pods, and note the Node that your pod runs on, such as 10.xxx.xx.xxx.
9. Return to the IBM Cloud clusters console, select your cluster, and click the Worker Nodes tab. Find the Public IP of the worker node that matches the private IP of the node that the pod runs on.
10. In a tab in your browser, form the URL of your app by combining http:// with the public IP and TCP port that you previously retrieved i.e. **http:// 159.122.178.57: 30357**. The Welcome to Liberty page is displayed. Great job! You just deployed your first app in your Kubernetes cluster.





The screenshot shows the Kubernetes Overview page with detailed workload status. The 'Deployments' section shows one deployment named 'hello-world' with 1/1 pods. The 'Pods' section shows one pod named 'hello-world-57bc9fd44s-wrcs5' running on node '10.144.228.178' with 0 restarts, 9.00m CPU usage, and 229.73MB memory usage. The 'Replica Sets' section shows one replica set named 'hello-world-57bc9fd44s' with 1/1 pods.

Name	Images	Labels	Pods	Created ↑
hello-world	Show all	Show all	1 / 1	4 minutes ago

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
hello-world-57bc9fd44s-wrcs5	Show all	Show all	10.144.228.178	Running	0	9.00m	229.73MB	4 minutes ago

Name	Images	Labels	Pods	Created ↑
hello-world-57bc9fd44s	Show all	Show all	1 / 1	4 minutes ago

The screenshot shows the Kubernetes Overview page with the 'Services' section. The table lists two services: 'liberty' and 'kubernetes'.

Name	Labels	Type	Cluster IP	Internal Endpoints	External Endpoints	Created ↑
liberty	Show all	NodePort	172.21.160.240	liberty:9080 TCP liberty:30357 TCP	-	25 minutes ago
kubernetes	Show all	ClusterIP	172.21.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	a day ago