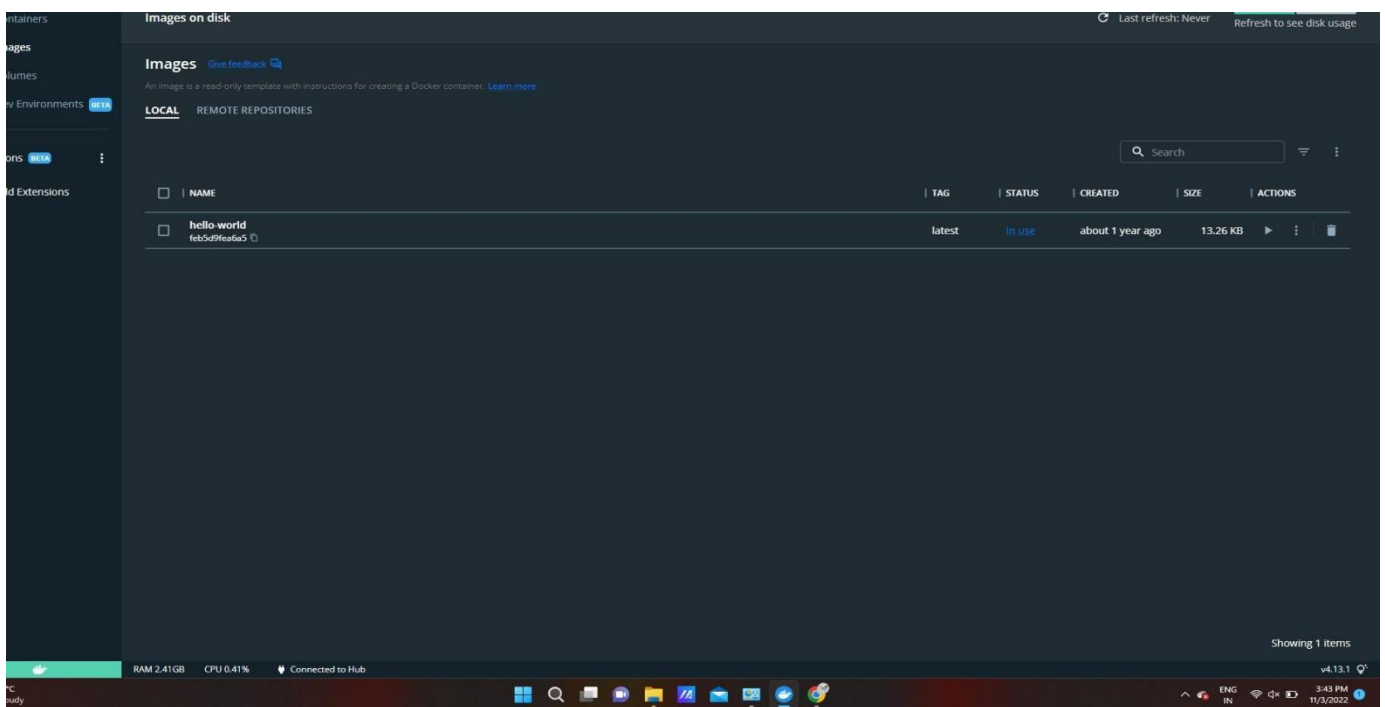


Assignment -4

Assignment Date	25 October 2022
Student Name	S.Mahalakshmi
Student Roll Number	110819205010
Maximum Marks	2 Marks

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



03:51:19

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.8
node1

cdhp1je0_cdhp5bu3tccg00fmt5b0

IP
192.168.0.8
OPEN PORT

Memory
CPU

SSH
ssh ip172-18-0-32-cdhp1je0qau0008f971g@direct.labs.play

DELETE EDITOR

```
#
# The FWD team.
#####
[node1] (local) root@192.168.0.8 ~
$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:e18f0a77aefabe047a671ab3ec3eed05414477c951ab1a6f352a06974245fe7
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
[node1] (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.
   (amd64)
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/

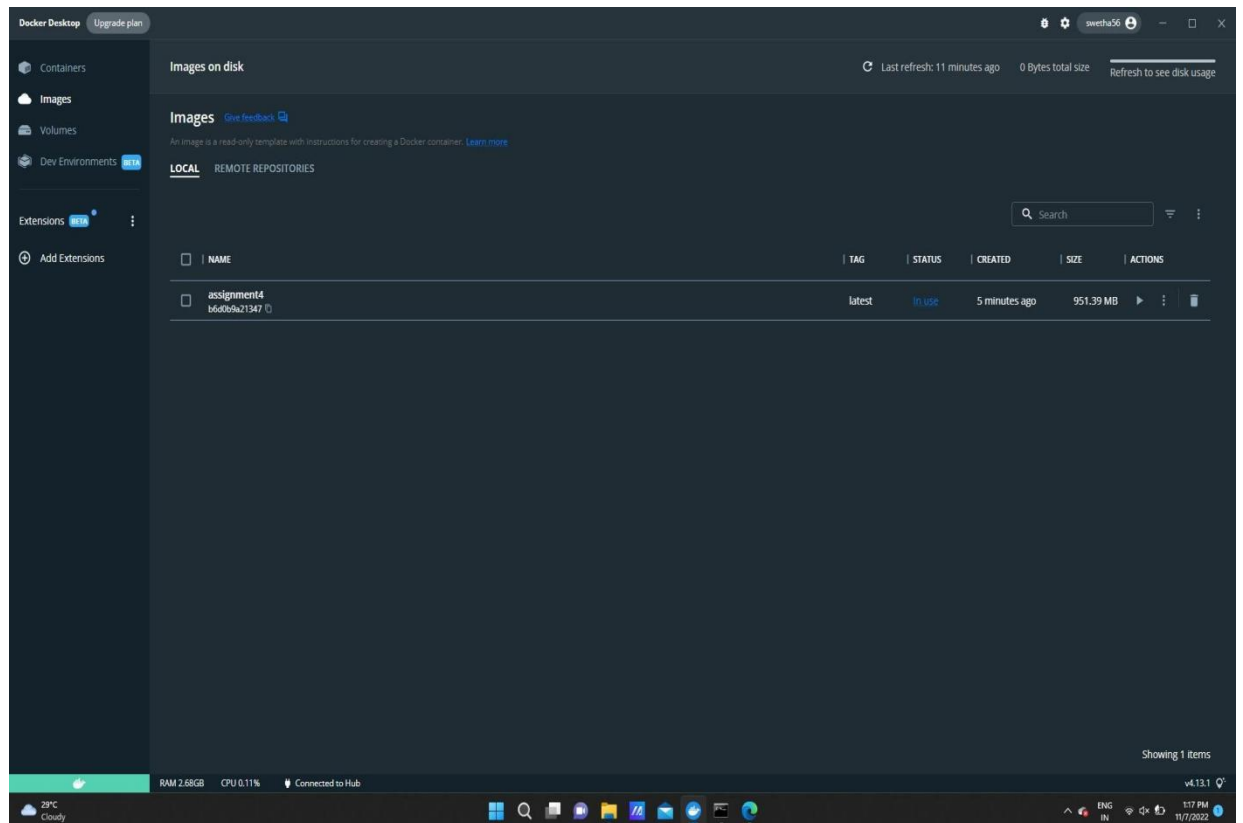
[node1] (local) root@192.168.0.8 ~
$
```

29°C
Cloudy

ENG
IN

3:37 PM
11/3/2022

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



```
C:\Users\ASUS\OneDrive\Desktop\assignment4>docker build -t assignment4 .
[+] Building 293.9s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring Dockerfile: 184B
=> [internal] load .dockerignore
=> transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.11.0
=> [internal] load build context
=> transferring context: 368B
=> [1/5] FROM docker.io/library/python:3.11.0@sha256:fc80bada71c07cec7e2d2440c6fba13763897ba66f2aaf6267db43e89f0f
=> resolve docker.io/library/python:3.11.0@sha256:fc80bada71c07cec7e2d2440c6fba13763897ba66f2aaf6267db43e89f0f
=> sha256:08cd1fb0dc67527a560cd0f5e4ad9d7040117b06160204426281d641cac1 8.52kB / 8.52kB
=> sha256:17c9e141fdb387e5a1c07d4f9b6a5ac149b96429f3a3e5476045047770 55.05kB / 55.05kB
=> sha256:4edcd85876ec18412817019074f5e04a8ede42fc80b6af136f3f80d78a70d 16.80kB / 16.80kB
=> sha256:fc80bada71c07cec7e2d2440c6fba13763897ba66f2aaf6267db43e89f0f 1.14kB / 2.14kB
=> sha256:de4dc6caea8081b0b7377e10220a914da403c93fa79663cf2dcf100b06f1 5.10kB / 5.10kB
=> sha256:c43926c605d221f0a40da1e719de3143072fc0b0804c1b1e679f90c7fca3 2.22kB / 2.22kB
=> sha256:a796cfff046e6a91291f076b19e0e93c03e4a4de0014842aeb4c0c4211a43 54.59kB / 54.59kB
=> sha256:74fbfde6af91271fb08fba1716224dce5cbe4ad3609943792a9c0ba4d6d3d 196.87kB / 196.87kB
=> sha256:16fe15aeb099f36017fe42b590b1a622b29ebek3622e9e13df1457825eb37 6.29kB / 6.29kB
=> sha256:e9ee507b0b0e40992b092cc5f7c04f9a4e7bcca990ba5e20a44e450591f0 23.23kB / 23.23kB
=> sha256:40dbb46d211d6f0a8db2b30a5ade7a7ed1a724fff4280433ee990818c081e 2348 / 2348
=> extracting sha256:17c9e141fdb387e5a1c07d4f9b6a5ac149b96429f3a3e5476045047770
=> sha256:30b93c4e049c23740f1b1ab279ache785703a167f6812aa31eafec93c2c923 3.80kB / 3.80kB
=> extracting sha256:de4dc6caea8081b0b7377e10220a914da403c93fa79663cf2dcf100b06f1
=> extracting sha256:4edcd85876ec18412817019074f5e04a8ede42fc80b6af136f3f80d78a70d
=> extracting sha256:a796cfff046e6a91291f076b19e0e93c03e4a4de0014842aeb4c0c4211a43
=> extracting sha256:74fbfde6af91271fb08fba1716224dce5cbe4ad3609943792a9c0ba4d6d3d
=> extracting sha256:16fe15aeb099f36017fe42b590b1a622b29ebek3622e9e13df1457825eb37
=> extracting sha256:e9ee507b0b0e40992b092cc5f7c04f9a4e7bcca990ba5e20a44e450591f0
=> extracting sha256:40dbb46d211d6f0a8db2b30a5ade7a7ed1a724fff4280433ee990818c081e
=> extracting sha256:30b93c4e049c23740f1b1ab279ache785703a167f6812aa31eafec93c2c923
=> [2/5] WORKDIR /app
=> [3/5] COPY requirements.txt ./
=> [4/5] RUN pip install -r requirements.txt
=> [5/5] COPY . .
=> exporting to image
=> exporting layers
=> writing image sha256:b0d0b6a13475abdbf81901c750b18d6a3a72af51ed0000531710eb29c0d00
=> naming to docker.io/library/assignment4

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

C:\Users\ASUS\OneDrive\Desktop\assignment4>
C:\Users\ASUS\OneDrive\Desktop\assignment4>docker run -p 5000 assignment4 .
docker: Error response from daemon: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: exec: ".": executable file not found in $PATH: unknown.

C:\Users\ASUS\OneDrive\Desktop\assignment4>docker run -p 5000 assignment4
* 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://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 110-845-361
172.17.0.1 - - [07/Nov/2022 07:46:54] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [07/Nov/2022 07:47:04] "GET /favicon.ico HTTP/1.1" 404 -

C:\Users\ASUS\OneDrive\Desktop\assignment4>
```

