

ASSIGNMENT – 4

DOCKER AND KUBERNETES

Date	04 November 2022
Team ID	PNT2022TMID11934
Project Name	Inventory Management System For Retailers

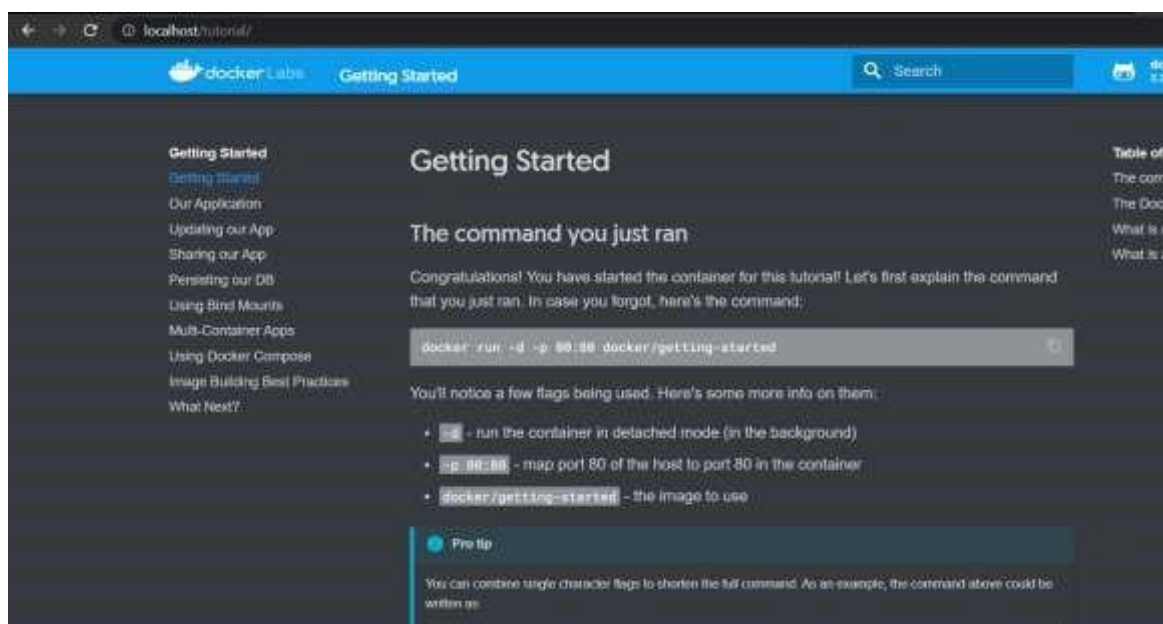
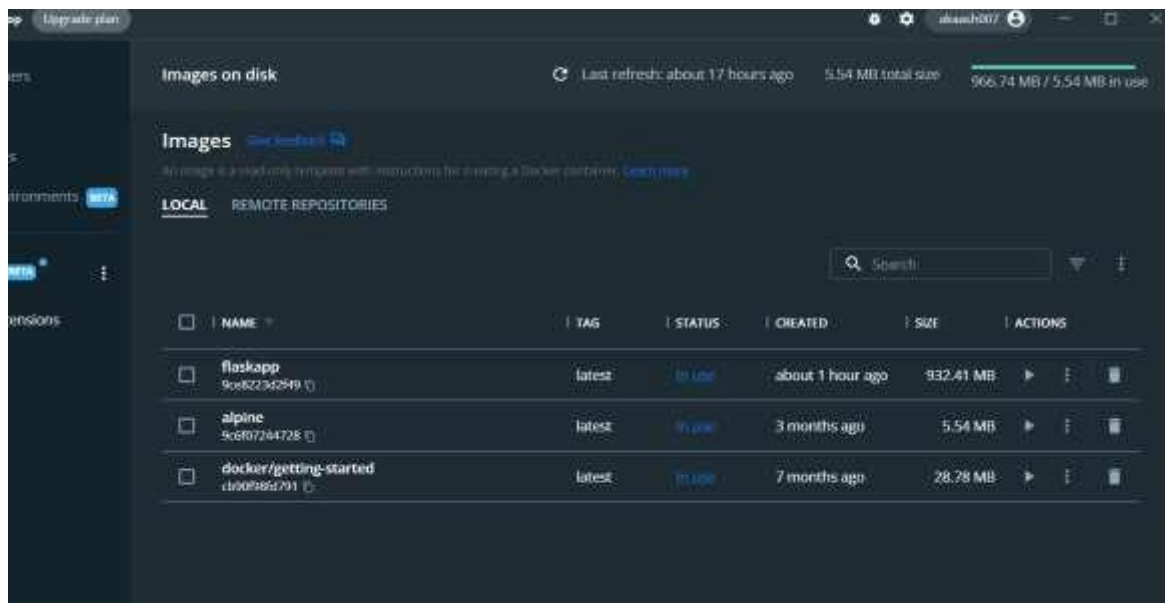
Question-1: pull an image from docker hub and run it in docker playground.

1) pull an image form docker hub

```
PowerShell
Loading personal and system profiles took 541ms.
+ assignment 4 git:(main) docker pull docker/getting-started
Using default tag: latest
latest: Pulling from docker/getting-started
df9b9388f04a: Pull complete
5867cba5fcbd: Pull complete
4b639e65cb3b: Pull complete
061ed9e2b976: Pull complete
bc19f3e8eeb1: Pull complete
4071be97c256: Pull complete
79b586f1a54b: Pull complete
0c9732f525d6: Pull complete
Digest: sha256:b558be874169471bd4e65bd6eac8c303b271a7ee8553ba47481b73b2bf597aae
Status: Downloaded newer image for docker/getting-started:latest
docker.io/docker/getting-started:latest
+ assignment 4 git:(main) |
```

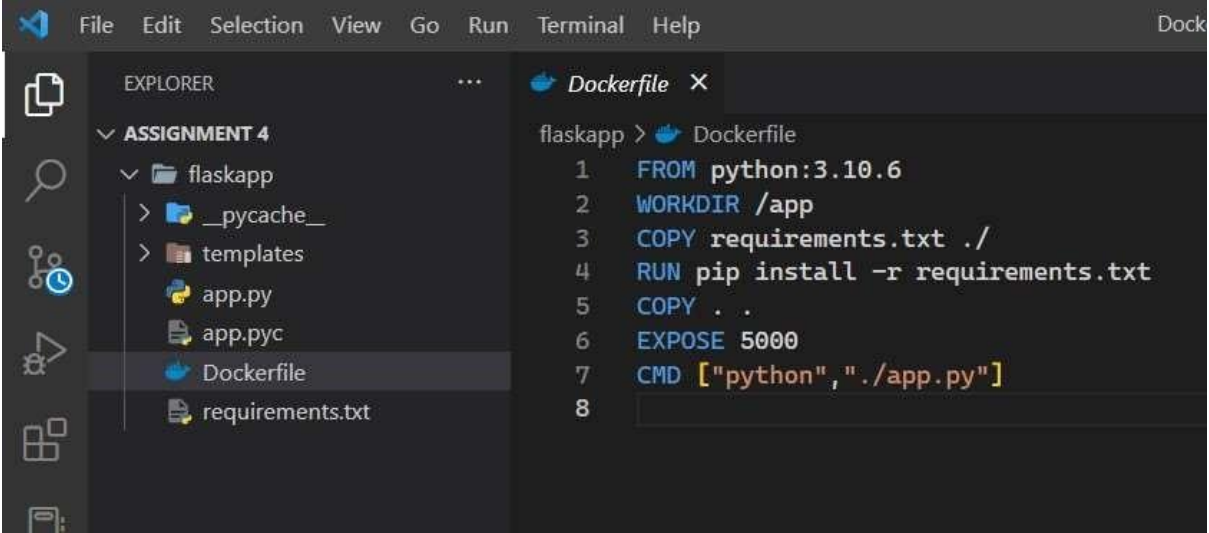
2)run it in docker playground

```
Digest: sha256:b558be874169471bd4e65bd6eac8c303b271a7ee8553ba47481b73b2bf597aae
Status: Downloaded newer image for docker/getting-started:latest
docker.io/docker/getting-started:latest
+ assignment 4 git:(main) docker run -d -p 80:80 docker/getting-started
ee6d34bd49e20106c8d3a3cc85bab0bde9c96a667bb3112bc896358efd6d2f68
+ assignment 4 git:(main) D|
```



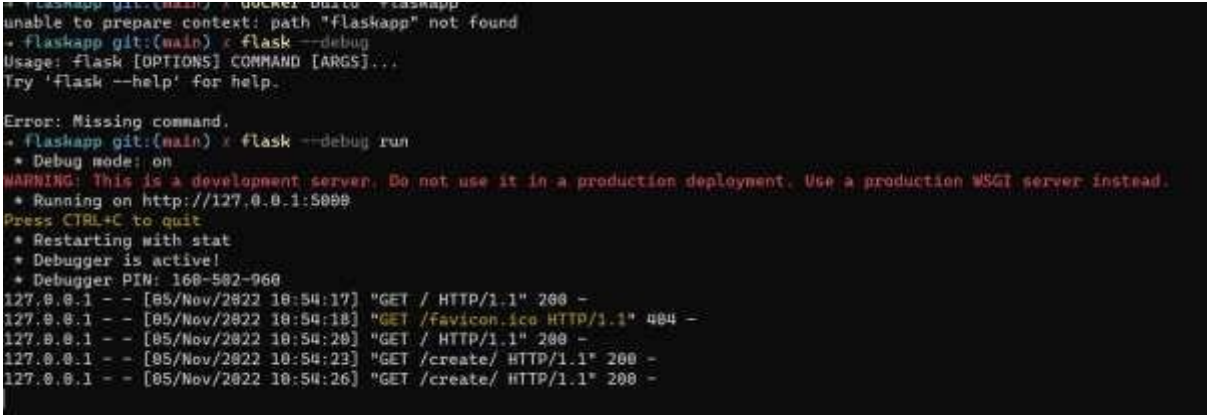
Question-2: Create a docker file for the job portal application and deploy it in docker application.

1)Creating a docker file for the job portal application



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project structure under 'ASSIGNMENT 4' with a subfolder 'flaskapp' containing files like 'app.py', 'app.pyc', 'Dockerfile', and 'requirements.txt'. The 'Dockerfile' file is selected. On the right, the Dockerfile editor shows the following content:

```
flaskapp > 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
```



The screenshot shows a terminal window with the following output:

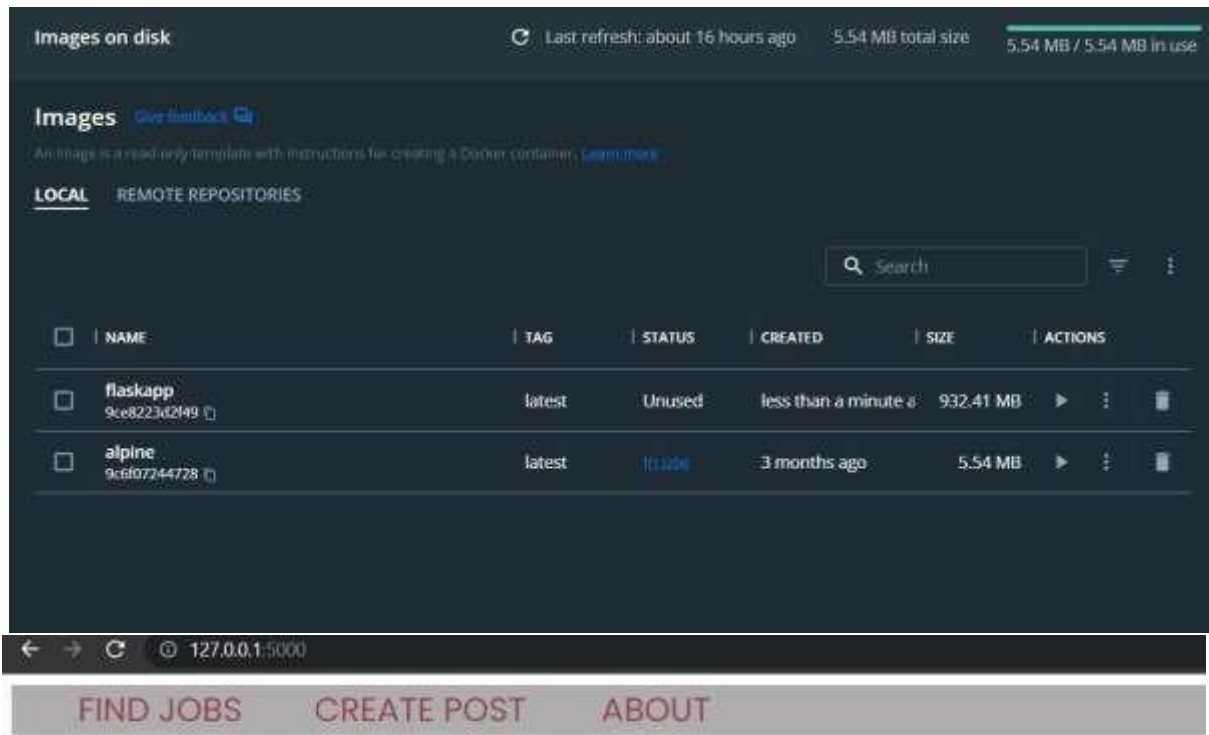
```
flaskapp git:(main) x docker build -f Dockerfile -t flaskapp .
unable to prepare context: path "flaskapp" not found
flaskapp git:(main) x flask --debug
Usage: flask [OPTIONS] COMMAND [ARGS]...
Try 'flask --help' for help.

Error: Missing command.
flaskapp git:(main) x flask --debug run
* 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 http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 160-502-960
127.0.0.1 - - [05/Nov/2022 10:54:17] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [05/Nov/2022 10:54:18] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [05/Nov/2022 10:54:20] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [05/Nov/2022 10:54:23] "GET /create/ HTTP/1.1" 200 -
127.0.0.1 - - [05/Nov/2022 10:54:26] "GET /create/ HTTP/1.1" 200 -
```

2) deploy it in docker application

```
PowerShell
- Flaskapp git:(main) # docker build -t flaskapp .
[+] Building 200.2s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 179B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.10.6
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load build context
=> => transferring context: 2.56kB
== [1/8] FROM docker.io/library/python:3.10.6@sha256:748efdfb7e4aac9a8422bd8c33a83e8979a240cd36477cbb3eb / 180.5s
=> => resolve docker.io/library/python:3.10.6@sha256:748efdfb7e4aac9a8422bd8c33a83e8979a240cd36477cbb3eb 0.0s
=> => sha256:d23a86300b1821681f4946d7777bba5cd15612140d8e7d11189370048cfc90 8.53kB / 8.13kB
=> => sha256:8a99d13e55e7a9ef17ff21776f37f655c7e1706971f8704a99280968d81f8e4 5.15MB / 5.15MB
=> => sha256:fa9c7528c80216139e8e07bfc642a7762e7b1dca155ab85540a41508830e57d0 18.80MB / 18.80MB
=> => sha256:708cfd8f7e4aac9a8422bd8c33a83e8979a240cd36477cbb3eb 2.35kB / 2.35kB
=> => sha256:8d1f943ceaaef3b3ce95df3c8926e7958836b040b780176bfc9c56d5f37aaf13fca 2.22kB / 2.22kB
=> => sha256:1871565cc8cf8c365c9661d3fbc164e71d81f1b6438c6179565428f99a9da2e 55.81MB / 55.81MB
=> => sha256:53ad872f9cd18cf8eb93b182b2e755e11acc0ef68babe9bf1043c80de1001a 54.50MB / 54.50MB
=> => sha256:d86983117833b7183794f1781e4083d2cfac611c7908c853b8e2a156e6448a 196.79MB / 196.79MB
=> => extracting sha256:1871565cc8cf8c365c9661d3fbc164e71d81f1b6438c6179565428f99a9da2e 2.3s
=> => sha256:d86983117833b7183794f1781e4083d2cfac611c7908c853b8e2a156e6448a 6.29MB / 6.29MB
=> => extracting sha256:fa9c7528c80216139e8e07bfc642a7762e7b1dca155ab85540a41508830e57d0 8.2s
=> => extracting sha256:53ad872f9cd18cf8eb93b182b2e755e11acc0ef68babe9bf1043c80de1001a 3.0s
=> => sha256:c71af6c37d59adc04c54d3c1483884dfe2b356bb184f8d57ea22c6a8e1d385a5 18.82MB / 18.82MB
=> => sha256:884a18b3c784353e88cb5fcd12fbaee1c87048fc345f8e435e8a335413da49b 234B / 234B
=> => sha256:4314b3fa8293d19ddc1c1559093aaa88f21601a7c85a11c6da8c8dc484fb6ed3c 3.84MB / 3.84MB
=> => extracting sha256:d86983117833b7183794f1781e4083d2cfac611c7908c853b8e2a156e6448a 7.0s
=> => extracting sha256:d86983117833b7183794f1781e4083d2cfac611c7908c853b8e2a156e6448a 8.1s
=> => extracting sha256:fa9c7528c80216139e8e07bfc642a7762e7b1dca155ab85540a41508830e57d0 0.3s
=> => extracting sha256:53ad872f9cd18cf8eb93b182b2e755e11acc0ef68babe9bf1043c80de1001a 3.0s
=> => sha256:c71af6c37d59adc04c54d3c1483884dfe2b356bb184f8d57ea22c6a8e1d385a5 18.82MB / 18.82MB
=> => sha256:884a18b3c784353e88cb5fcd12fbaee1c87048fc345f8e435e8a335413da49b 234B / 234B
=> => sha256:4314b3fa8293d19ddc1c1559093aaa88f21601a7c85a11c6da8c8dc484fb6ed3c 3.84MB / 3.84MB
=> => extracting sha256:d86983117833b7183794f1781e4083d2cfac611c7908c853b8e2a156e6448a 7.0s
=> => extracting sha256:d86983117833b7183794f1781e4083d2cfac611c7908c853b8e2a156e6448a 8.1s
=> => extracting sha256:fa9c7528c80216139e8e07bfc642a7762e7b1dca155ab85540a41508830e57d0 0.3s
=> => extracting sha256:53ad872f9cd18cf8eb93b182b2e755e11acc0ef68babe9bf1043c80de1001a 3.0s
=> => sha256:c71af6c37d59adc04c54d3c1483884dfe2b356bb184f8d57ea22c6a8e1d385a5 18.82MB / 18.82MB
=> => sha256:884a18b3c784353e88cb5fcd12fbaee1c87048fc345f8e435e8a335413da49b 234B / 234B
=> => sha256:4314b3fa8293d19ddc1c1559093aaa88f21601a7c85a11c6da8c8dc484fb6ed3c 3.84MB / 3.84MB
=> [2/8] WORKDIR /app
=> [3/8] COPY requirements.txt ./
=> [4/8] RUN pip install -r requirements.txt
=> [5/8] COPY
=> exporting to image
=> exporting layers
=> writing image sha256:9ee2f2d2f49cc12aa777e8b82e565fcc1f49aa344e67fe97b53ba57189311e
=> naming to docker.io/library/flaskapp
0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
# flaskapp git:(main) |
```



Blog Page

Messages

SDE Job in OHO

perks: unlimited snacks and drinks

Message Two

Message Two Content

Question-3: Create a IBM container registry and deploy hello world app or jobportalapp

1) create a IBM container registry


```

+ ~ git:(main) # ibmcloud
NAME:
  C:\Program Files\IBM\Cloud\bin\ibmcloud.exe - A command line tool to interact with IBM Cloud
  Find more information at: https://ibm.biz/cli-docs

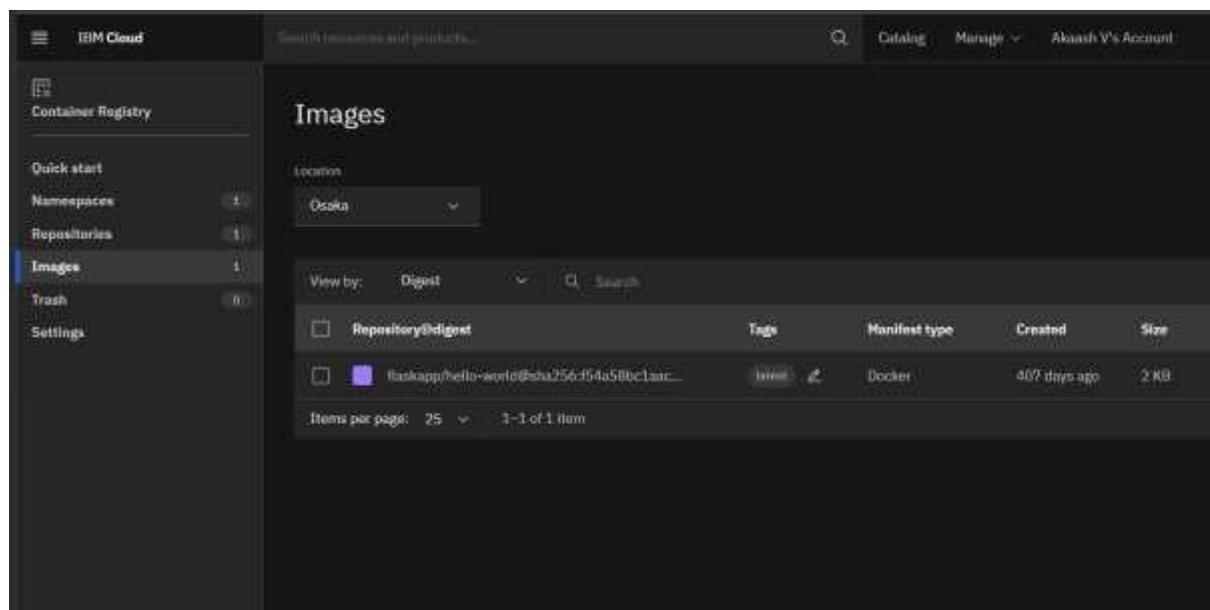
USAGE:
  [environment variables] C:\Program Files\IBM\Cloud\bin\ibmcloud.exe [global options] command [arguments.
  ptions]

VERSION:
  2.12.1+b8488a1-2022-10-31T15:08:10+00:00

COMMANDS:
  account      Manage accounts, users, orgs and spaces
  api          Set or view target API endpoint
  billing      Retrieve usage and billing information
  catalog      Manage catalog
  cf           Run Cloud Foundry CLI with IBM Cloud CLI context
  config       Write default values to the config
  cr           Manage IBM Cloud Container Registry content and configuration.
  dev         Create, develop, deploy, and monitor applications
  enterprise   Manage enterprise, account groups and accounts.
  iam         Manage identities and access to resources
  login       Log user in
  logout      Log user out
  plugin      Manage plug-ins and plug-in repositories
  regions     List all the regions

```

2)deploy hello world or jobportal



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 note port

1) Creating a Kubernetes cluster in IBM cloud

