Assignment Kubernetes / Docker

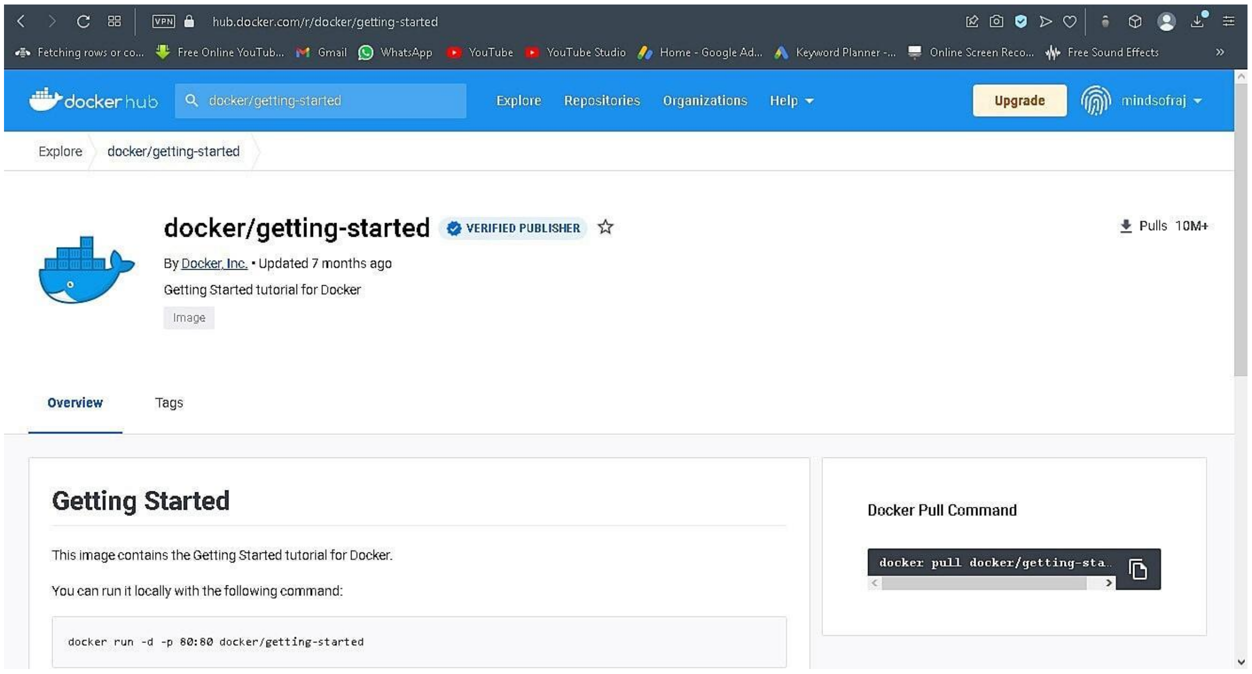
Team ID	PNT2022TMID40352
Project Name	Plasma Donor Application

Prerequisites :

- Download and Install the Docker Desktop for windows
- Login to the Docker Desktop

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

First We have to signup to the Docker Hub (https://hub.docker.com)
Search for the docker images



docker/getting-started

By Docker, Inc. • Updated 7 months ago

Getting Started tutorial for Docker

Getting Started

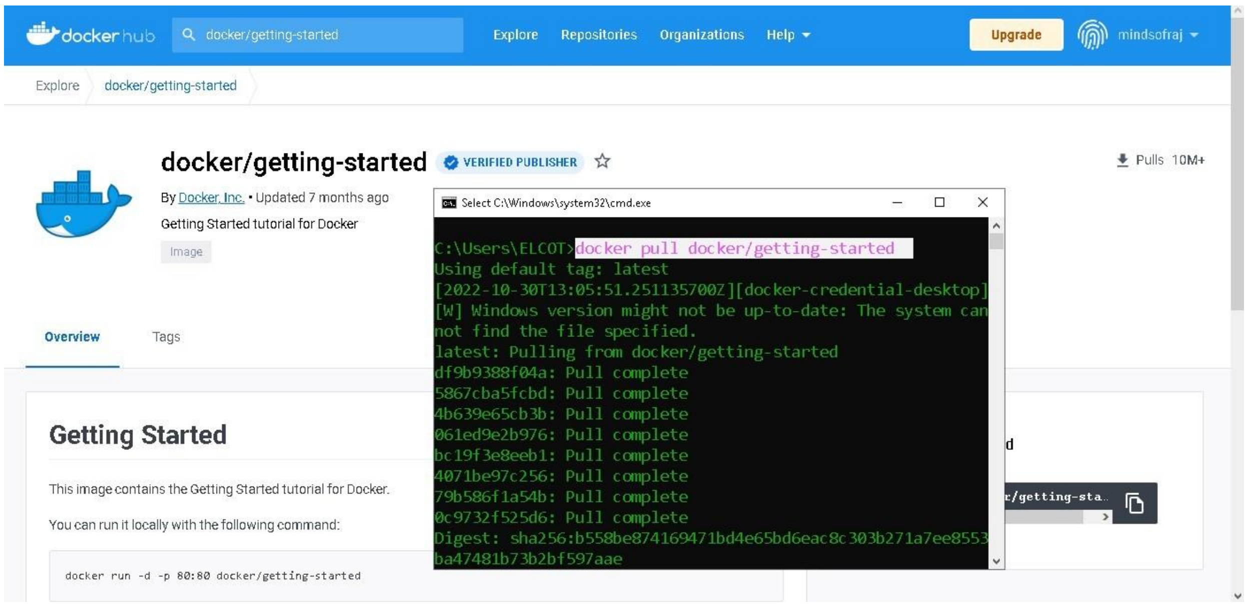
This image contains the Getting Started tutorial for Docker.

You can run it locally with the following command:

```
docker run -d -p 80:80 docker/getting-started
```

```
docker pull docker/getting-sta...
```

Run the pull command in command prompt



Getting Started

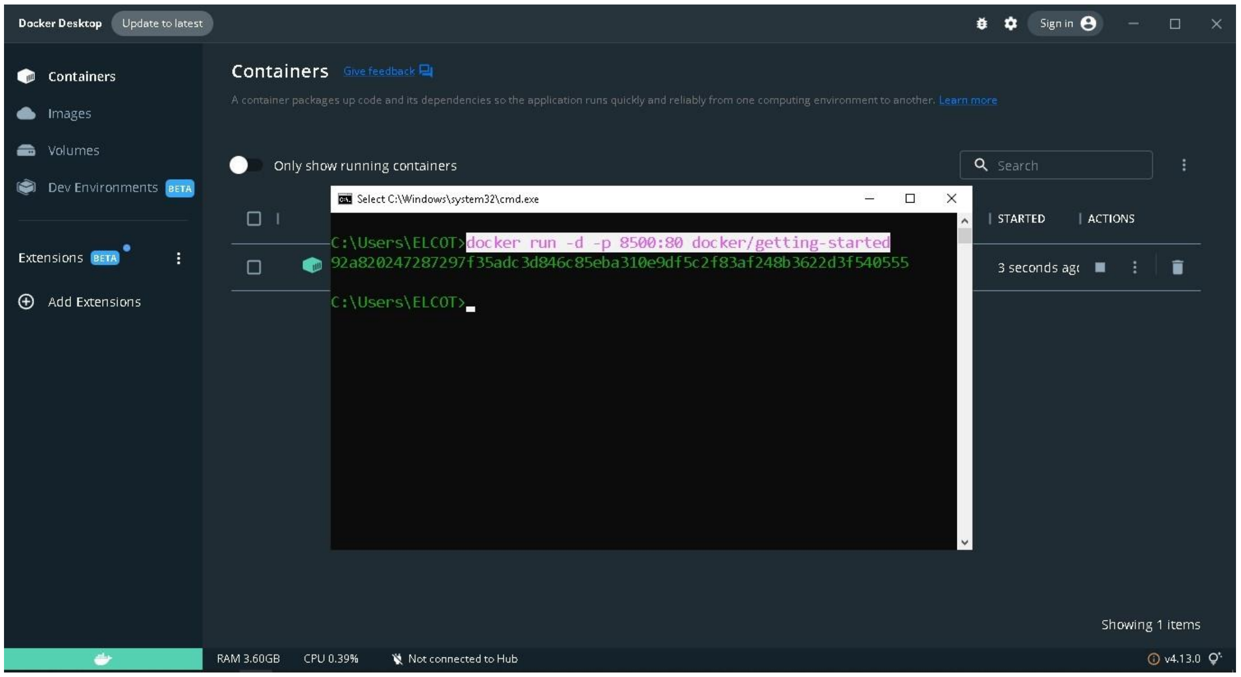
This image contains the Getting Started tutorial for Docker.

You can run it locally with the following command:

```
docker run -d -p 80:80 docker/getting-started
```

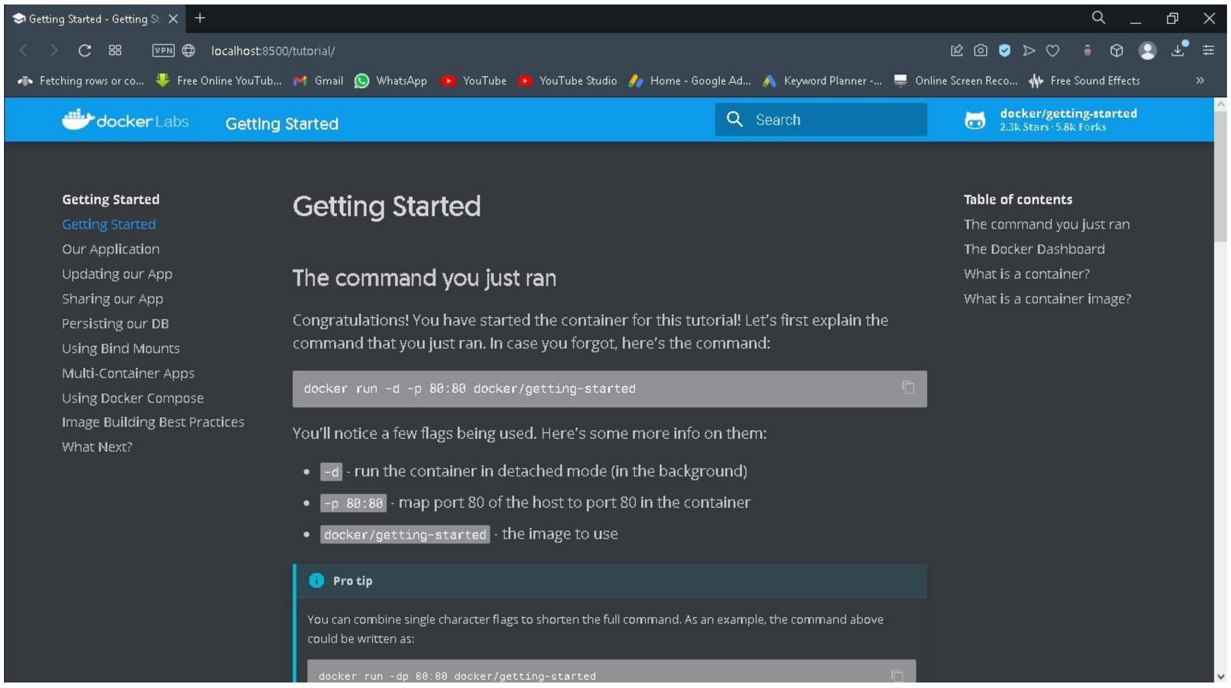
```
C:\Users\ELCOT>docker pull docker/getting-started
Using default tag: latest
[2022-10-30T13:05:51.251135700Z][docker-credential-desktop]
[W] Windows version might not be up-to-date: The system can
not find the file specified.
latest: Pulling from docker/getting-started
df9b9388f04a: Pull complete
5867cba5fcbd: Pull complete
4b639e65cb3b: Pull complete
061ed9e2b976: Pull complete
bc19f3e8eeb1: Pull complete
4071be97c256: Pull complete
79b586f1a54b: Pull complete
0c9732f525d6: Pull complete
Digest: sha256:b558be874169471bd4e65bd6eac8c303b271a7ee8553
ba47481b73b2bf597aae
```

After Successfully downloading the docker image run it using the command prompt in a desired port.



```
C:\Users\ELCOT>docker run -d -p 8500:80 docker/getting-started
92a820247287297f35adc3d846c85eba310e9df5c2f83af248b3622d3f540555

C:\Users\ELCOT>
```

Then open the localhost inside the browser on the given port *https://localhost:8500/*



Getting Started

The command you just ran

Congratulations! You have started the container for this tutorial! Let's first explain the command that you just ran. In case you forgot, here's the command:

```
docker run -d -p 80:80 docker/getting-started
```

You'll notice a few flags being used. Here's some more info on them:

- `-d` - run the container in detached mode (in the background)
- `-p 80:80` - map port 80 of the host to port 80 in the container
- `docker/getting-started` - the image to use

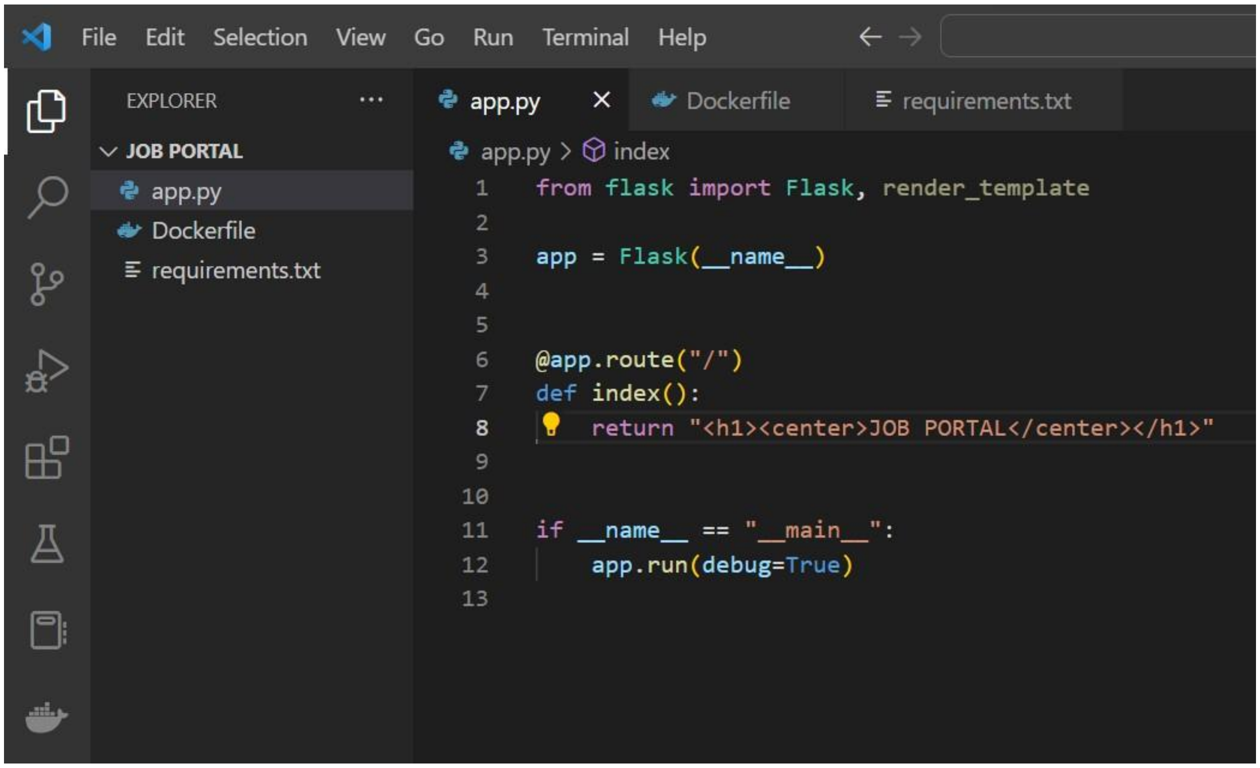
Pro tip

You can combine single character flags to shorten the full command. As an example, the command above could be written as:

```
docker run -dp 80:80 docker/getting-started
```

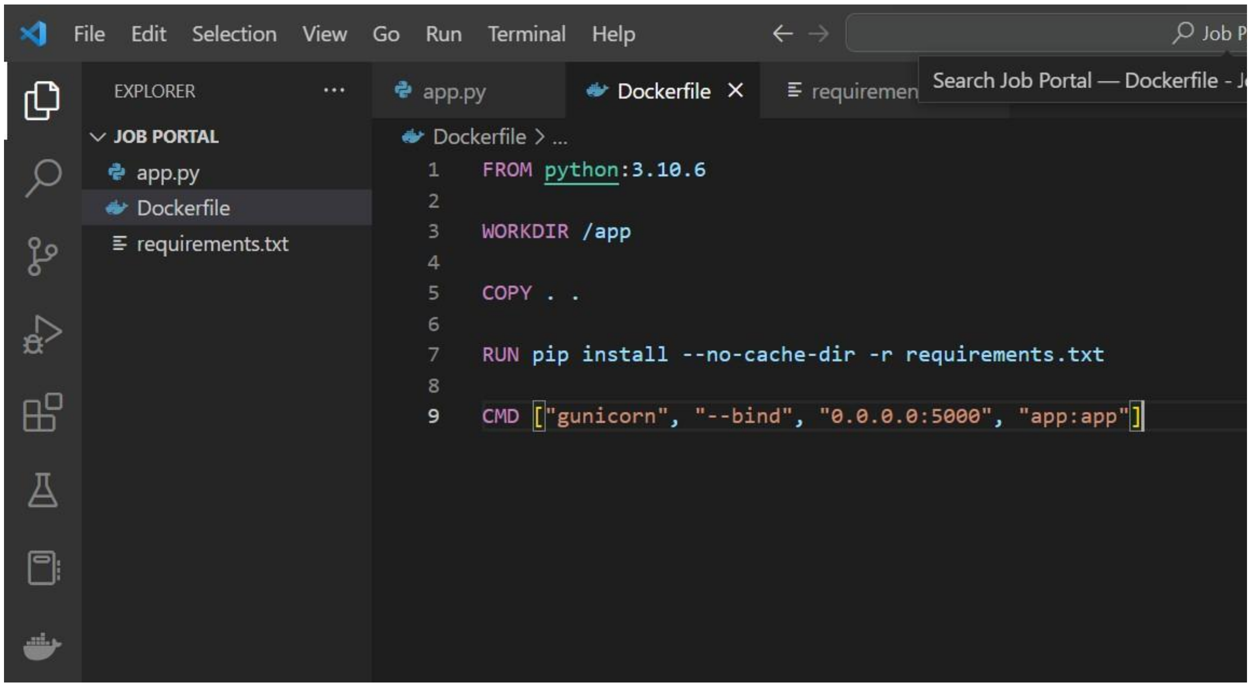
2. Create a docker file for the jobportal application and deploy it in Docker desktop application.

Create Job Portal Flask Application



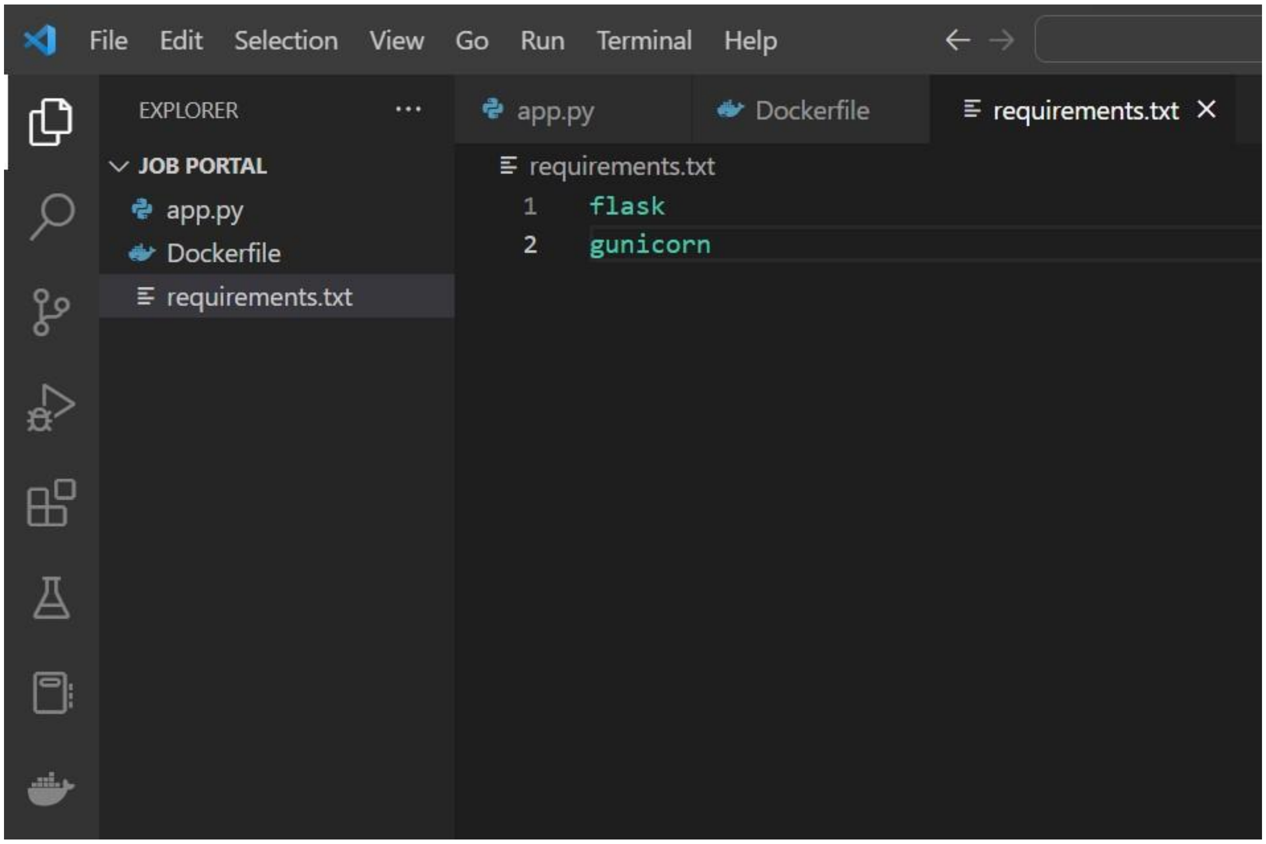
```python
from flask import Flask, render_template

app = Flask(__name__)


@app.route("/")
def index():
    return "<h1><center>JOB PORTAL</center></h1>"


if __name__ == "__main__":
    app.run(debug=True)
```

Create a Dockerfile



```dockerfile
FROM python:3.10.6

WORKDIR /app

COPY . .

RUN pip install --no-cache-dir -r requirements.txt

CMD ["gunicorn", "--bind", "0.0.0.0:5000", "app:app"]
```

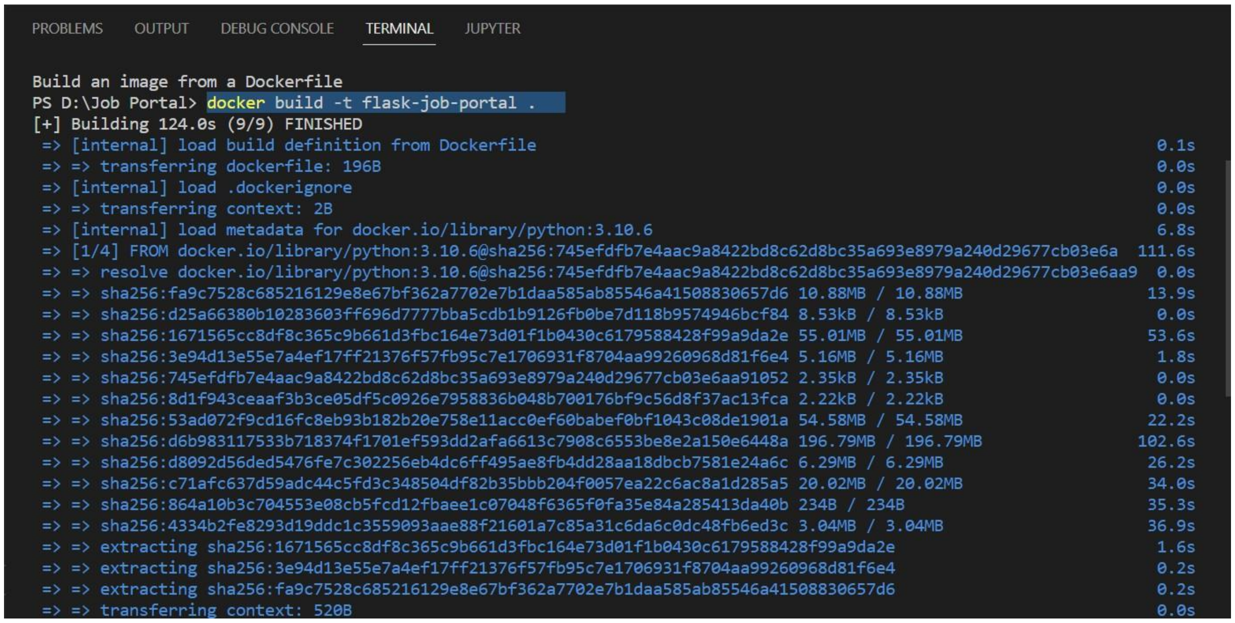
Create Requirements.txt File



```
flask
gunicorn
```

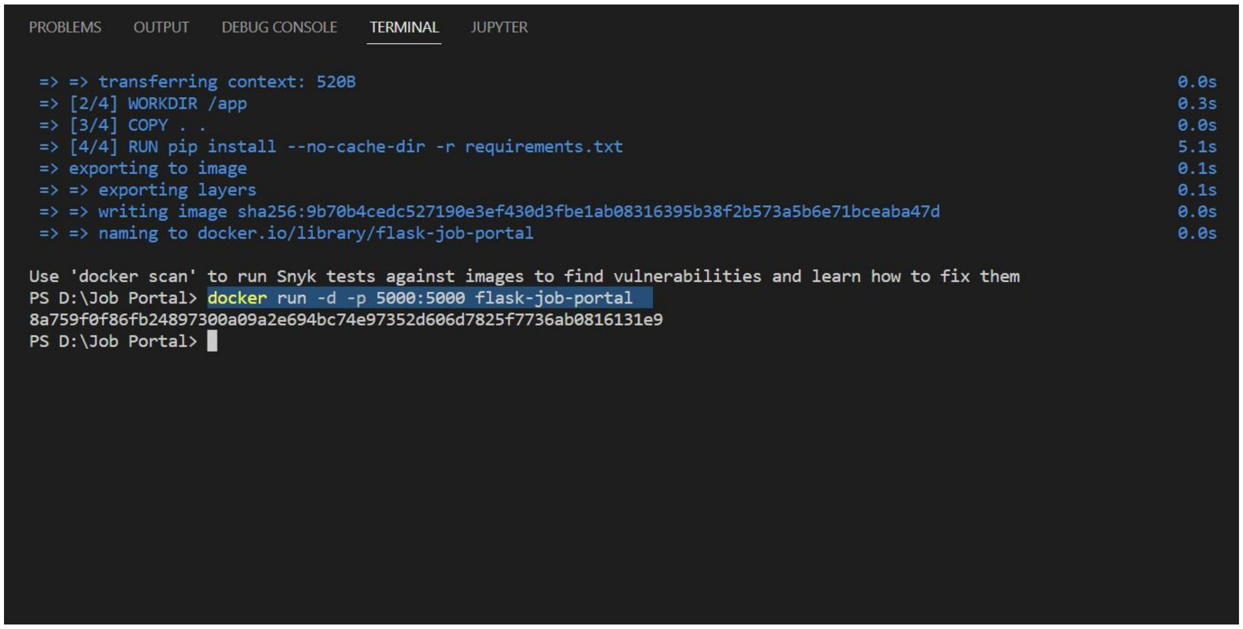
Build the Docker Image Using the Docker

docker build -t flask-job-portal .



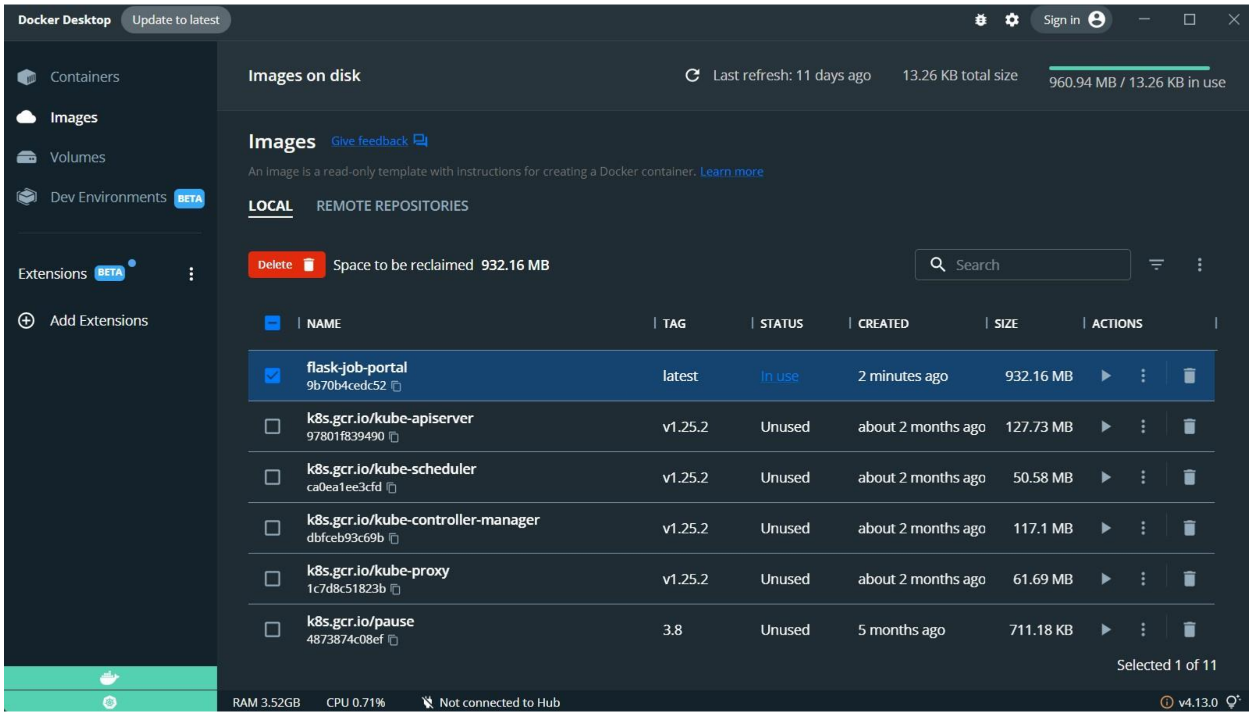
```
Build an image from a Dockerfile
PS D:\Job Portal> docker build -t flask-job-portal .
[+] Building 124.0s (9/9) FINISHED
 => [internal] load build definition from Dockerfile                                                     0.1s
 => => transferring dockerfile: 196B                                                                     0.0s
 => [internal] load .dockerignore                                                                        0.0s
 => => transferring context: 2B                                                                          0.0s
 => [internal] load metadata for docker.io/library/python:3.10.6                                         6.8s
 => [1/4] FROM docker.io/library/python:3.10.6@sha256:745efdfb7e4aac9a8422bd8c62d8bc35a693e8979a240d29677cb03e6a  111.6s
 => => resolve docker.io/library/python:3.10.6@sha256:745efdfb7e4aac9a8422bd8c62d8bc35a693e8979a240d29677cb03e6aa9  0.0s
 => => sha256:fa9c7528c685216129e8e67bf362a7702e7b1daa585ab85546a41508830657d6 10.88MB / 10.88MB          13.9s
 => => sha256:d25a66380b10283603ff696d7777bba5cdb1b9126fb0be7d118b9574946bcf84 8.53kB / 8.53kB             0.0s
 => => sha256:1671565cc8df8c365c9b661d3fbc164e73d01f1b0430c6179588428f99a9da2e 55.01MB / 55.01MB          53.6s
 => => sha256:3e94d13e55e7a4ef17ff21376f57fb95c7e1706931f8704aa99260968d81f6e4 5.16MB / 5.16MB             1.8s
 => => sha256:745efdfb7e4aac9a8422bd8c62d8bc35a693e8979a240d29677cb03e6aa91052 2.35kB / 2.35kB             0.0s
 => => sha256:8d1f943ceaaf3b3ce05df5c0926e7958836b048b700176bf9c56d8f37ac13fca 2.22kB / 2.22kB             0.0s
 => => sha256:53ad072f9cd16fc8eb93b182b20e758e11acc0ef60babef0bf1043c08de1901a 54.58MB / 54.58MB          22.2s
 => => sha256:d6b983117533b718374f1701ef593dd2afa6613c7908c6553be8e2a150e6448a 196.79MB / 196.79MB       102.6s
 => => sha256:d8092d56ded5476fe7c302256eb4dc6ff495ae8fb4dd28aa18dbcb7581e24a6c 6.29MB / 6.29MB            26.2s
 => => sha256:c71afc637d59adc44c5fd3c348504df82b35bbb204f0057ea22c6ac8a1d285a5 20.02MB / 20.02MB          34.0s
 => => sha256:864a10b3c704553e08cb5fcd12fbaee1c07048f6365f0fa35e84a285413da40b 234B / 234B                35.3s
 => => sha256:4334b2fe8293d19ddc1c3559093aae88f21601a7c85a31c6da6c0dc48fb6ed3c 3.04MB / 3.04MB            36.9s
 => => extracting sha256:1671565cc8df8c365c9b661d3fbc164e73d01f1b0430c6179588428f99a9da2e                 1.6s
 => => extracting sha256:3e94d13e55e7a4ef17ff21376f57fb95c7e1706931f8704aa99260968d81f6e4                 0.2s
 => => extracting sha256:fa9c7528c685216129e8e67bf362a7702e7b1daa585ab85546a41508830657d6                 0.2s
 => => transferring context: 520B                                                                        0.0s
```

Run the Docker Image using Docker Command docker run -d -p 5000:5000 flask-job-portal



```
 => => transferring context: 520B                                                                           0.0s
 => [2/4] WORKDIR /app                                                                                      0.3s
 => [3/4] COPY . .                                                                                          0.0s
 => [4/4] RUN pip install --no-cache-dir -r requirements.txt                                                5.1s
 => exporting to image                                                                                      0.1s
 => => exporting layers                                                                                     0.1s
 => => writing image sha256:9b70b4cedc527190e3ef430d3fbe1ab08316395b38f2b573a5b6e71bceaba47d                0.0s
 => => naming to docker.io/library/flask-job-portal                                                         0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
PS D:\Job Portal> docker run -d -p 5000:5000 flask-job-portal
8a759f0f86fb24897300a09a2e694bc74e97352d606d7825f7736ab0816131e9
PS D:\Job Portal> 
```

An image is Created in the Docker desktop



Images on disk   Last refresh: 11 days ago   13.26 KB total size   960.94 MB / 13.26 KB in use

Images

An image is a read-only template with instructions for creating a Docker container. Learn more

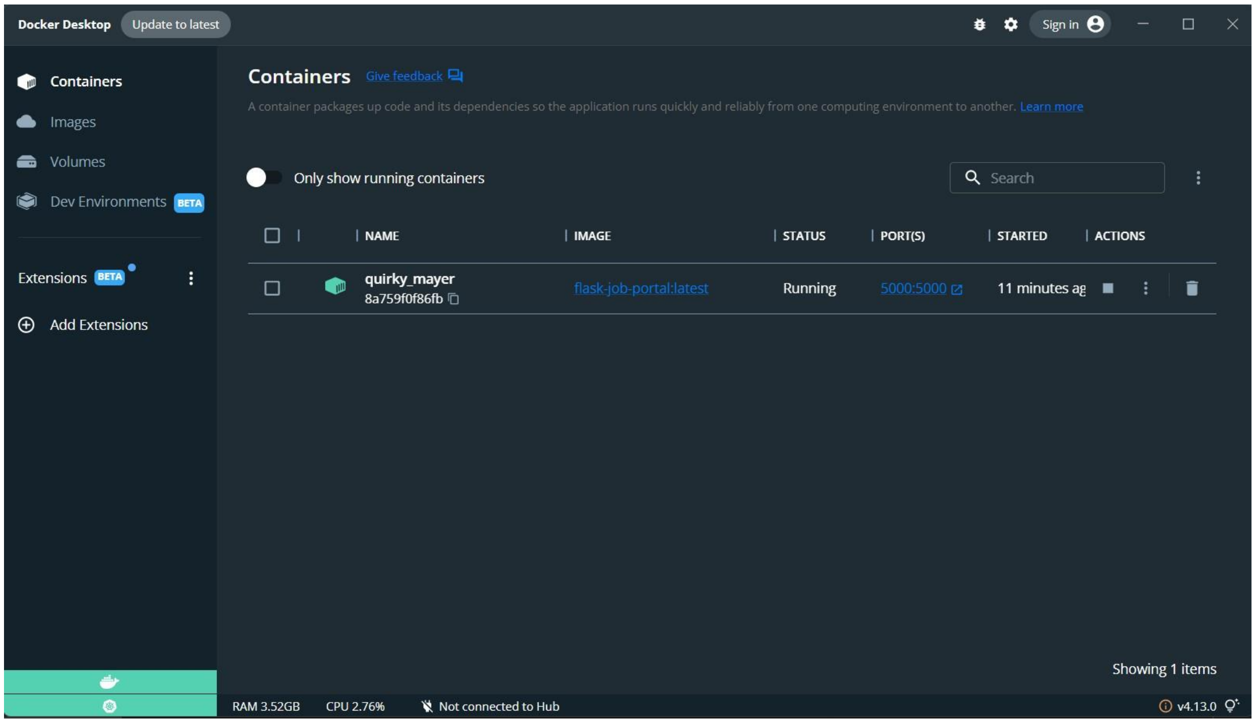
Space to be reclaimed 932.16 MB

NAME	TAG	STATUS	CREATED	SIZE	ACTIONS
flask-job-portal 9b70b4cedc52	latest	In use	2 minutes ago	932.16 MB	
k8s.gcr.io/kube-apiserver 97801f839490	v1.25.2	Unused	about 2 months ago	127.73 MB	
k8s.gcr.io/kube-scheduler ca0ea1ee3cfd	v1.25.2	Unused	about 2 months ago	50.58 MB	
k8s.gcr.io/kube-controller-manager dbfceb93c69b	v1.25.2	Unused	about 2 months ago	117.1 MB	
k8s.gcr.io/kube-proxy 1c7d8c51823b	v1.25.2	Unused	about 2 months ago	61.69 MB	
k8s.gcr.io/pause 4873874c08ef	3.8	Unused	5 months ago	711.18 KB	

Selected 1 of 11

RAM 3.52GB   CPU 0.71%   Not connected to Hub   v4.13.0

A Container is created on the port 5000



Containers

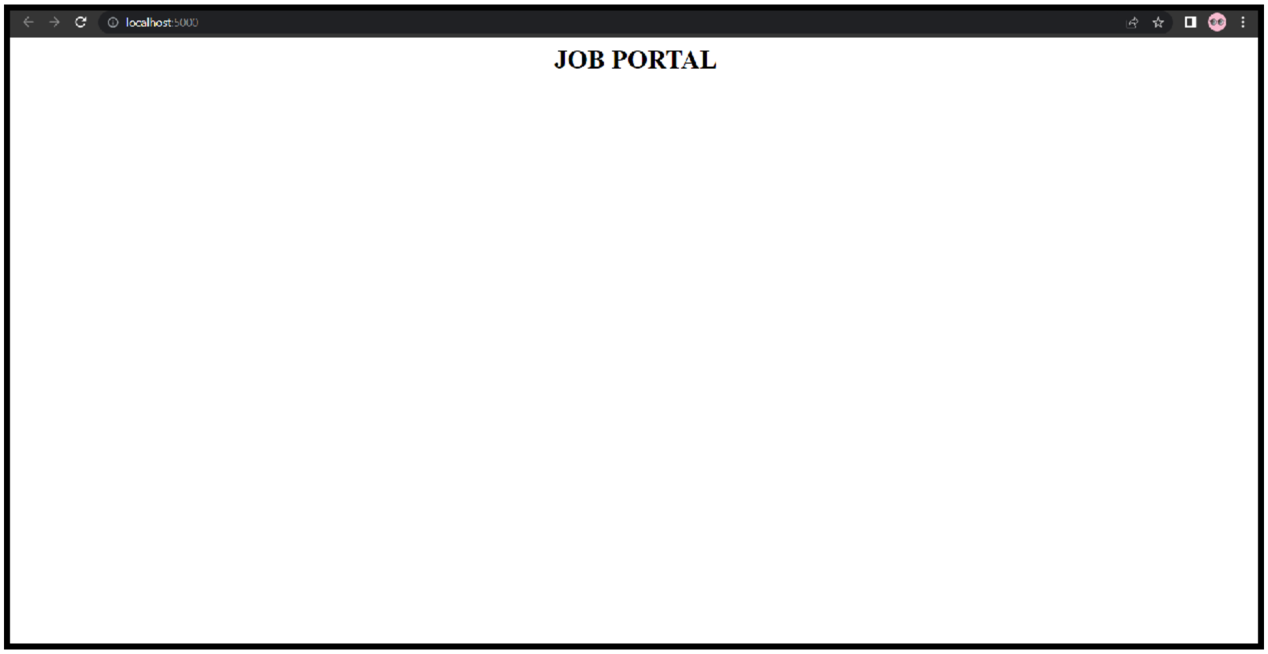
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
quirky_mayer 8a759f0f86fb	flask-job-portal:latest	Running	5000:5000	11 minutes ag	

Showing 1 items

RAM 3.52GB CPU 2.76% Not connected to Hub v4.13.0

Our app is running in the browser on the localhost
http://127.0.0.1:5000/



JOB PORTAL