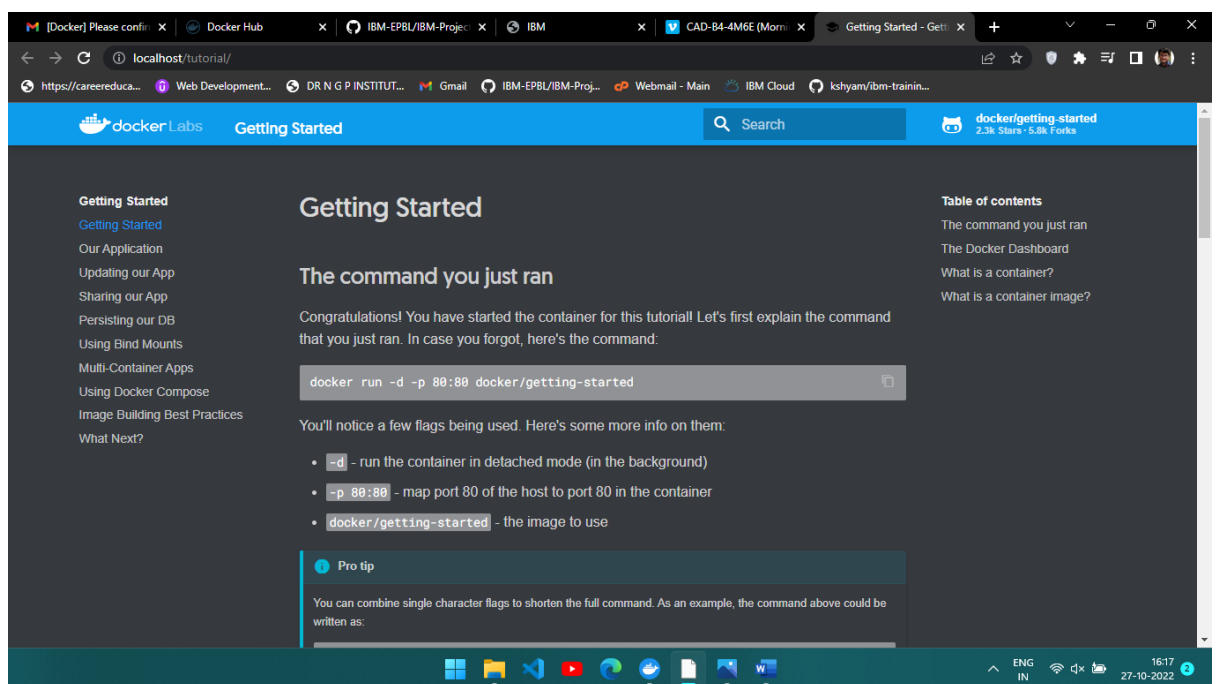
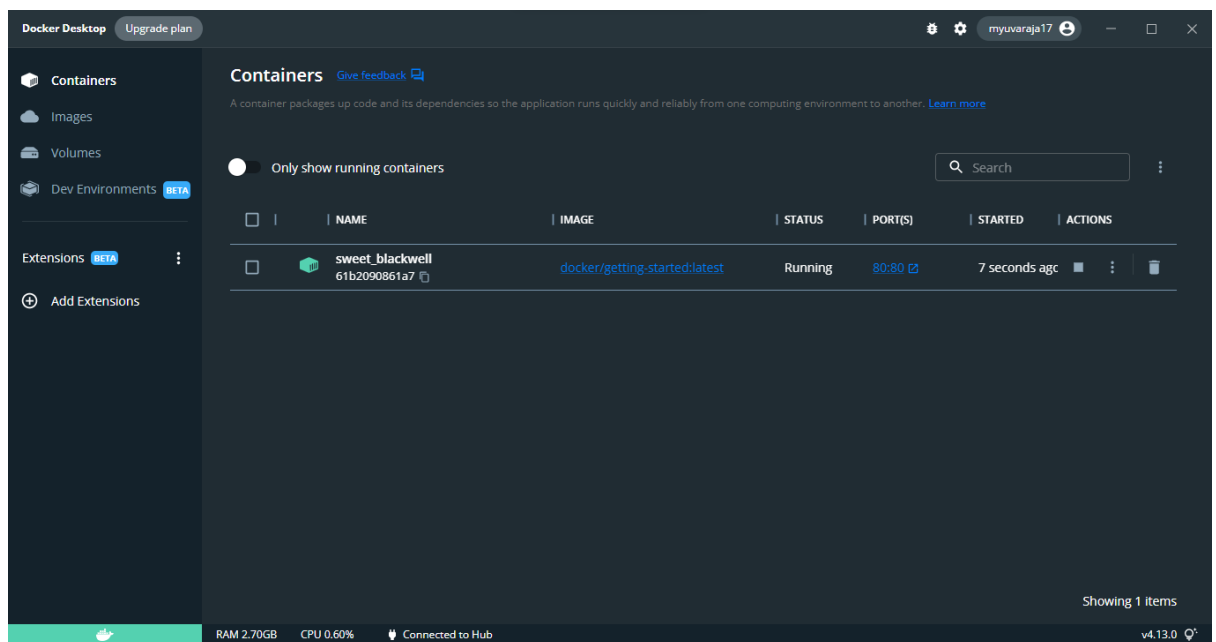


1. Pull a image from docker hub and run it in docker playground.



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

1. Application to be deployed in Docker.

The screenshot shows a web browser window with multiple tabs. The active tab is titled 'Add a New Message' and shows a URL of 'https://127.0.0.1:5000/create/'. The browser's address bar and tabs are visible at the top. Below the browser window, the FlaskApp interface is shown. It has a navigation bar with three links: 'FLASKAPP', 'CREATE', and 'ABOUT'. The main heading is 'Add a New Message'. Below the heading, there are two form fields: 'Title' with a placeholder 'Message title' and 'Message Content' with a placeholder 'Message content'. A 'Submit' button is located at the bottom left of the form.



II. Building the application : `docker build -t flask_app .`

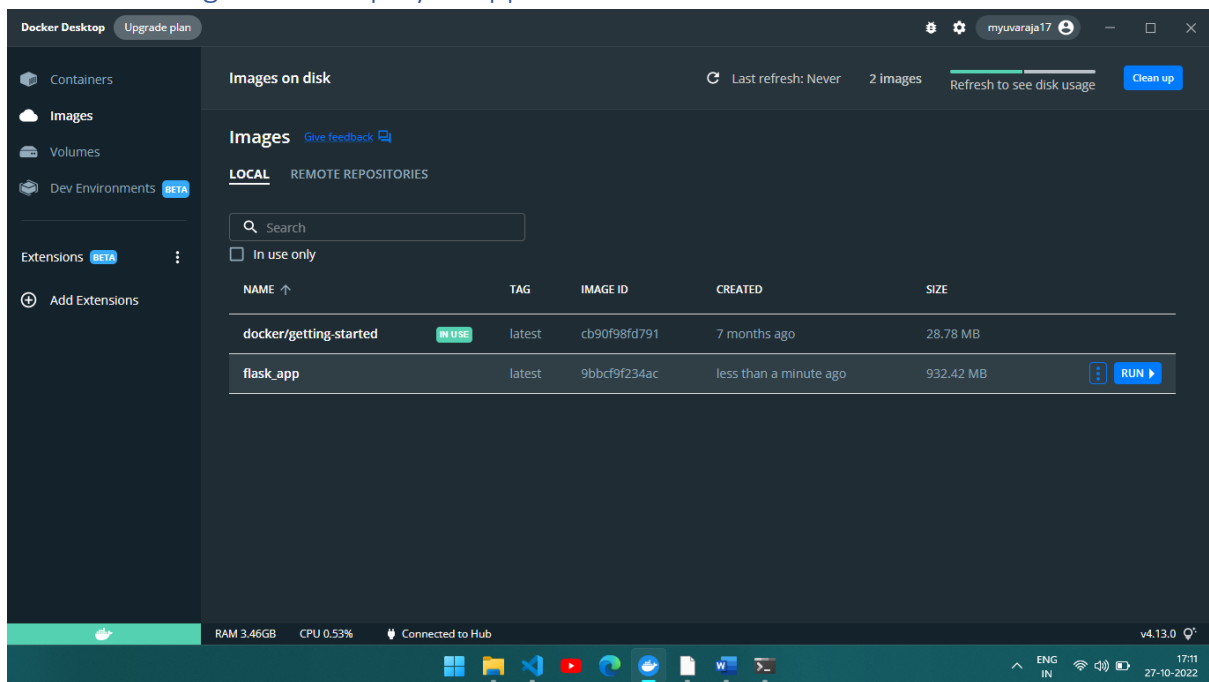
```
C:\Users\yuvaj\OneDrive\Documents\GitHub\IBM-Project-27219-166085866\Assignments\Team Member - Yuvaraja\Assignment 4\Question-2\flask_with_form_and_docker-main\docker build -t flask_app .
```

```
[+] Building 219.3s (11/11) FINISHED
=> [internal] Load build definition from Dockerfile
0.2s
=> transferring dockerfile: 198B
0.0s
=> [internal] Load dockerignore file
0.1s
=> transferring context: 28B
0.0s
=> [internal] Load metadata for docker.io/library/python:3.10.6
18.1s
=> [auth] library/python:pull token for registry-1.docker.io
0.4s
[1/5] FROM docker.io/library/python:3.10.6@sha256:7945ef0b7e4aac9a0472b08c62b0c35a693e8979a240d2677cb93d6aa91852
182.4s
=> resolve docker.io/library/python:3.10.6@sha256:7945ef0b7e4aac9a0472b08c62b0c35a693e8979a240d2677cb93d6aa91852
0.1s
=> sha256:3d9d1d150e74f177679740b6747180118704a0972086031f6e051009 / 5.10MB
9.0s
=> sha256:c556c638b010236818706667770ba3cd3012f206e7a11b059746dc0f418153d / 8.15MB
9.0s
=> sha256:fad7528c685216123e67b7f3627702e74a585a8554e41508336c5706 / 10.80MB
14.0s
=> sha256:7945ef0b7e4aac9a0472b08c62b0c35a693e8979a240d2677cb93d6aa91852 / 2.35MB
2 / 2.35MB
=> sha256:d1191c4e7432c3e545e7955235a6e0b70017049f4c5d0770c23fca / 2.22MB
2 / 2.22MB
=> sha256:1671965cc8d8c365c906a163f1c164736011f10430c6179568028f99a9da2e / 55.01MB / 55.01MB
64.9s
=> sha256:53ad077f9cd0f6c8e93b1825b28753a1cc6ebf0bbaef0bf1043c80b10b1a / 54.50MB / 54.50MB
71.6s
=> sha256:d08912732031261414f392da4f6d12c708b453aef2a1150e406 / 104.79MB / 104.79MB
109.2s
=> sha256:d8892d5d6d979747c38225eap4ccf497a8ef38aef06d28a18d0c7851c2448c / 6.20MB / 6.20MB
79.4s
=> extracting sha256:1671965cc8d8c365c906a163f1c164736011f10430c6179568028f99a9da2e
12.2s
=> sha256:c71efc37f059ac40c5f4c34850a20360b204007aa22c4ac8a1d5845 / 16.47MB / 16.47MB
91.9s
=> sha256:5d4130130c35c35fcd123aae1c1870b6c35f0473130a425513a0ab / 234B / 234B
91.9s
=> extracting sha256:433b047e293d19d0c1c1539089a8e8721601c74c581c16da6c0c40f8e41c / 3.00MB / 3.00MB
79.3s
=> extracting sha256:3e9d1d150e74f177679740b6747180118704a0972086031f6e051009 / 487084992089468d1f6e4
1.0s
=> extracting sha256:7945ef0b7e4aac9a0472b08c62b0c35a693e8979a240d2677cb93d6aa91852 / 8.15MB
0.0s
=> extracting sha256:53ad077f9cd0f6c8e93b1825b28753a1cc6ebf0bbaef0bf1043c80b10b1a / 54.50MB / 54.50MB
13.3s
=> extracting sha256:d08912732031261414f392da4f6d12c708b453aef2a1150e406 / 104.79MB / 104.79MB
24.7s
=> extracting sha256:d8892d5d6d979747c38225eap4ccf497a8ef38aef06d28a18d0c7851c2448c / 6.20MB / 6.20MB
1.0s
=> extracting sha256:c71efc37f059ac40c5f4c34850a20360b204007aa22c4ac8a1d5845 / 16.47MB / 16.47MB
3.2s
=> extracting sha256:504a10b3c70455248b5cfcd12fbaec107090f6365f07a15e8a2850413a40b
1.0s
=> extracting sha256:433b047e293d19d0c1c1539089a8e8721601c74c581c16da6c0c40f8e41c / 3.00MB
1.3s
=> [internal] Load build context
0.2s
=> transferring context: 11.56kB
0.0s
[2/5] WORKDIR /app
4.3s
[3/5] COPY requirements.txt /
4.3s
[4/5] RUN pip install -r requirements.txt
11.4s
[5/5] COPY .
0.1s
=> exporting to image
0.7s
=> exporting layers
0.5s
=> writing image sha256:9b0c9f9234ac339a2a9ac228a3ec238c49f9829a64e8089a3a31778cc4c
0.0s
=> naming docker.io/library/flask_app
0.0s
```

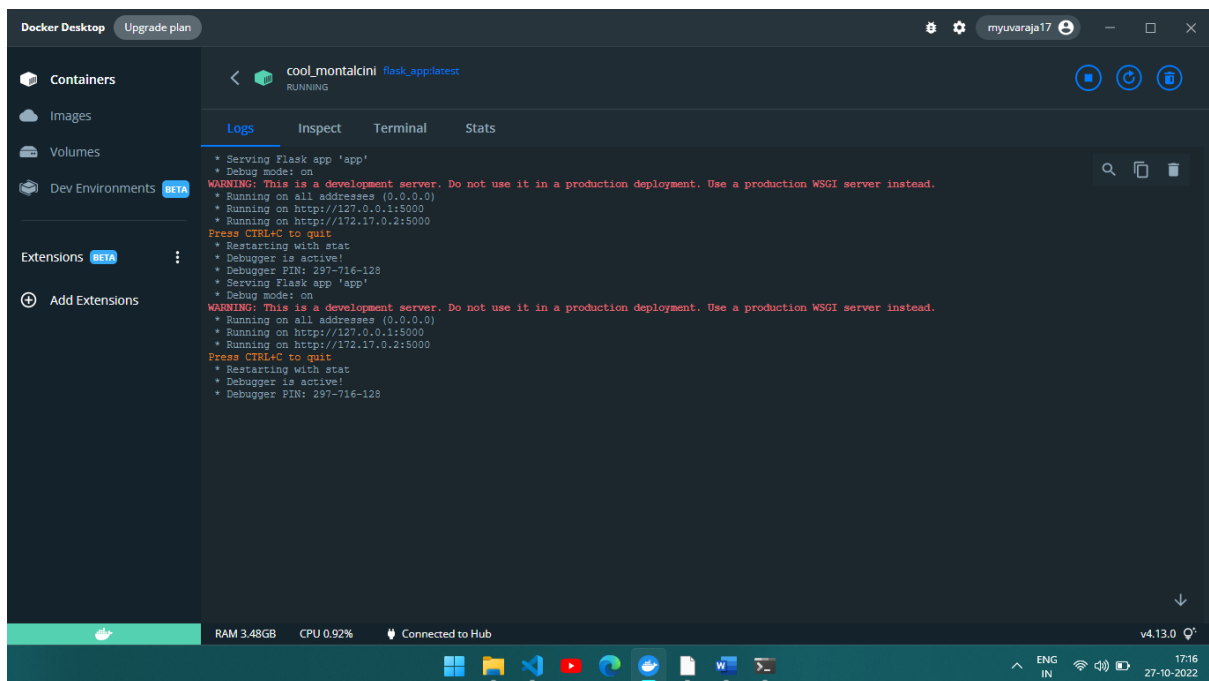
```
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

```
C:\Users\yuvaj\OneDrive\Documents\GitHub\IBM-Project-27219-166085866\Assignments\Team Member - Yuvaraja\Assignment 4\Question-2\flask_with_form_and_docker-main>
```

- Image of the deployed application.



The Pushed application is running in docker



Run using cmd

```
Command Prompt - docker r x + v
=> sha256:d8892d56ded5476fe7c302256eb4dc6ff495ae8fb4dd28aa18dbcb7581e24a6c 6.29MB / 6.29MB 74.0s
=> extracting sha256:1671565cc8df8c365c9b661d3fbc164e73d01f1b0430c6179588428f99a9da2e 12.2s
=> sha256:c71afc637d59adc44c5fd3c348504df82b35bbb204f0057ea22c6ac8a1d285a5 20.02MB / 20.02MB 91.9s
=> sha256:864a10b3c704553e08cb5fcd12fbaeelc07048f6365f0fa35e84a285413da40b 234B / 234B 74.5s
=> sha256:4334b2fe8293d19ddc1c3559093aae88f21601a7c85a31c6da6c0dc48fb6ed3c 3.04MB / 3.04MB 79.3s
=> extracting sha256:3e94d13e55e7a4ef17ff21376f57fb95c7e1706931f8704aa99260968d81f6e4 1.4s
=> extracting sha256:fa9c7528c685216129e8e67bf362a7702e7b1daa585ab85546a41508830657d6 0.8s
=> extracting sha256:53ad072f9cd16fc8eb93b182620e758e1acc6ef00babe0bf1043c08de1901a 13.3s
=> extracting sha256:d6b993117533b7183704f1701ef593d42afa6613c7908c6553be8a2a150e6448a 24.7s
=> extracting sha256:d8092d56ded5476fe7c302256eb4dc6ff495ae8fb4dd28aa18dbcb7581e24a6c 1.5s
=> extracting sha256:c71afc637d59adc44c5fd3c348504df82b35bbb204f0057ea22c6ac8a1d285a5 3.2s
=> extracting sha256:864a10b3c704553e08cb5fcd12fbaeelc07048f6365f0fa35e84a285413da40b 0.8s
=> extracting sha256:4334b2fe8293d19ddc1c3559093aae88f21601a7c85a31c6da6c0dc48fb6ed3c 1.3s
[internal] load build context
=> transferring context: 11.56kB 0.2s
[2/5] WORKDIR /app 0.1s
[3/5] COPY requirements.txt ./ 4.3s
[4/5] RUN pip install -r requirements.txt 0.1s
[5/5] COPY . . 11.0s
=> exporting to image 0.7s
=> exporting layers 0.6s
=> writing image sha256:9bbcf9f234ac339a2a9e4c228a3ec230c49f9829a64ee0d098a3ea31778cce4c 0.0s
=> naming to docker.io/library/flask_app 0.0s

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

C:\Users\yuval\OneDrive\Documents\GitHub\IBM-Project-27219-1660050866\Assignments\Team Member - Yuvaraja\Assignment 4\Question-2\flask_with_form_and_docker-main>docke
r run -p 5000:5000 flask_app
* 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://172.17.0.3:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 309-754-929
```

3.