

CREATING DOCKER IMAGE FOR FLASK APP

Date	7 November 2022
Team ID	PNT2022TMID27090
Project Name	Project – Plasma Donor Application

```
Command Prompt - app.py
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ramya>cd C:\Users\Ramya\OneDrive\Desktop\python

C:\Users\Ramya\OneDrive\Desktop\python>python app.py
* 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://192.168.29.76:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 824-467-356
127.0.0.1 - - [02/Nov/2022 16:38:02] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 16:38:08] "GET /create/ HTTP/1.1" 200 -
127.0.0.1 - - [02/Nov/2022 16:38:12] "GET / HTTP/1.1" 200 -
192.168.29.76 - - [02/Nov/2022 16:38:28] "GET / HTTP/1.1" 200 -
192.168.29.76 - - [02/Nov/2022 16:38:29] "GET /favicon.ico HTTP/1.1" 404 -
192.168.29.76 - - [02/Nov/2022 16:38:36] "GET / HTTP/1.1" 200 -
192.168.29.76 - - [02/Nov/2022 16:38:37] "GET /create/ HTTP/1.1" 200 -
```

```
Command Prompt
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ramya>cd C:\Users\Ramya\OneDrive\Desktop\python

C:\Users\Ramya\OneDrive\Desktop\python>docker build -t app:latest .
[*] Building 29.9s (40/10) FINISHED
-> [internal] load build definition from Dockerfile
-> => transferring dockerfile: 32B
-> [internal] load .dockerignore
-> => transferring context: 2B
-> [internal] load metadata for docker.io/library/python:3.10-buster
-> [auth] library/python:pull token for registry-1.docker.io
-> [1/4] FROM docker.io/library/python:3.10-buster@sha256:adb1e09abb1ce3459dbd98efdc2f51eebb25f07c3c8535c3c973a1fce7cf93e
-> [internal] load build context
-> => transferring context: 1.86kB
-> CACHED [2/4] WORKDIR /app
-> [3/4] COPY .
-> [4/4] RUN pip install --no-cache-dir -r requirements.txt
-> exporting to image
-> => exporting layers
-> => writing image sha256:2b2bc22736655b51b69a142539c2949175c3672ad538661c82845ad885c5d295
-> => naming to docker.io/library/app:latest

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

C:\Users\Ramya\OneDrive\Desktop\python>
```

