

## ## Docker for DevOps Engineers – Day 17

### ## Task:

- Create a Dockerfile for a simple web application (e.g. a Node.js or Python app)

➤ We have created dockerfile for an python project.

```
FROM python:2.7
```

```
# Creating Application Source Code Directory
```

```
RUN mkdir -p /usr/src/app
```

```
# Setting Home Directory for containers
```

```
WORKDIR /usr/src/app
```

```
# Installing python dependencies
```

```
COPY requirements.txt /usr/src/app/
```

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

```
# Copying src code to Container
```

```
COPY . /usr/src/app
```

```
# Application Environment variables
```

```
#ENV APP_ENV development
```

```
ENV PORT 8080
```

```
# Exposing Ports
```

```
EXPOSE $PORT
```

```
# Setting Persistent data
```

```
VOLUME ["/app-data"]
```

```
# Running Python Application
```

```
CMD gunicorn -b :$PORT -c gunicorn.conf.py main:app
```

- Build the image using the Dockerfile and run the container

➤ We have built image using below commad:

**`docker build -t python_using_flask .`**

➤ Now check if image is created or not using below command:

**`docker images`**

```
vagrant@vagrant:~/docker-with-python-app$ docker images
```

| REPOSITORY         | TAG    | IMAGE ID     | CREATED       | SIZE  |
|--------------------|--------|--------------|---------------|-------|
| python_using_flask | latest | 9214210267dc | 4 minutes ago | 906MB |
| <none>             | <none> | ede25867ee0f | 24 hours ago  | 704MB |

- Now we have to create container using created image:  
**docker run -it --name python\_app -e PYTHON\_APP\_ROOT\_PASSWORD=devops@123 -d python\_using\_flask**

```
vagrant@vagrant:~/docker-with-python-app$ docker run -it --name python_app -e PYTHON_APP_ROOT_PASSWORD=devops@123 -d python_using_flask
067fbf60e21acb4e4f27f93e75a00be7db332a7113756690654d99bab4cb414e
```

- Now check if container is created or not:  
**docker ps**

```
vagrant@vagrant:~/docker-with-python-app$ docker ps
```

| CONTAINER ID | IMAGE              | COMMAND                   | CREATED        | STATUS        | PORTS    | NAMES      |
|--------------|--------------------|---------------------------|----------------|---------------|----------|------------|
| 067fbf60e21a | python_using_flask | "/bin/sh -c 'gunicorn..." | 21 seconds ago | Up 17 seconds | 8080/tcp | python_app |

- Now we will execute our container:  
**docker exec -it 067fbf60e21a /bin/bash**

```
vagrant@vagrant:~$ docker exec -it 067fbf60e21a /bin/bash
root@067fbf60e21a:/usr/src/app# ls
Dockerfile README.md gunicorn.conf.py gunicorn_pid.txt main.py main.pyc requirements.txt
root@067fbf60e21a:/usr/src/app#
```

## - Verify that the application is working as expected by accessing it in a web browser

- Now go to the browser and give your IP address along with our port which is exposed in Dockerfile and if we are using ec2 instance then need to add inbound rule for our port and give access to ip.

## - Push the image to a public or private repository (e.g. Docker Hub )

- First check the differences if we are creating container using already exist container or pulled container using below command:

**docker diff**

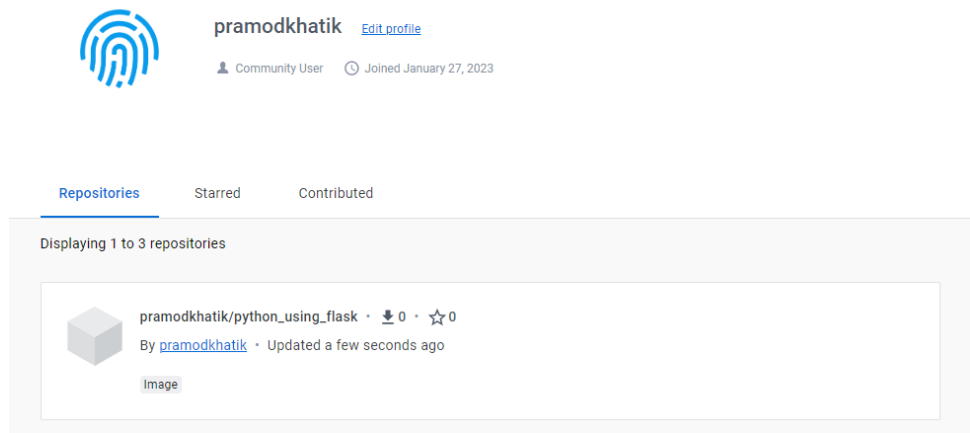
- Now create a tag for our image which we want to push:  
**docker tag python\_using\_flask pramodkhatik/python\_using\_flask:1.0.1**

```
vagrant@vagrant:~$ docker tag python_using_flask pramodkhatik/python_using_flask:1.0.1
```

- Now we can push the tag to github:  
**docker push pramodkhatik/python\_using\_flask:1.0.1**

```
vagrant@vagrant:~$ docker push pramodkhatik/python_using_flask:1.0.1
The push refers to repository [docker.io/pramodkhatik/python_using_flask]
13716f54adaf: Pushed
61216529d27f: Pushed
136ca20ff75d: Pushed
5f70bf18a086: Pushed
15de09171075: Pushed
e571d2d3c73c: Mounted from library/python
da7b0a80a4f2: Mounted from library/python
ceee8816bb96: Mounted from library/python
47458fb45d99: Mounted from library/python
46829331b1e4: Mounted from library/python
d35c5bda4793: Mounted from library/python
a3c1026c6bcc: Mounted from library/python
f1d420c2af1a: Mounted from library/python
461719022993: Mounted from library/python
1.0.1: digest: sha256:bdd6096e71613f19acfb17bd6340df0ff8c0dc04fa413a42fd1571e7b953cb8 size: 3261
```

- We can check from github profile if our image is pushed or not:



- We can pull this image later using below command:  
**docker pull pramodkhatik/python\_using\_flask**