CDEC B24

Name – Vaibhav Navneet Jorvekar

Hosting Python Flask App Using Docker

Step 1.

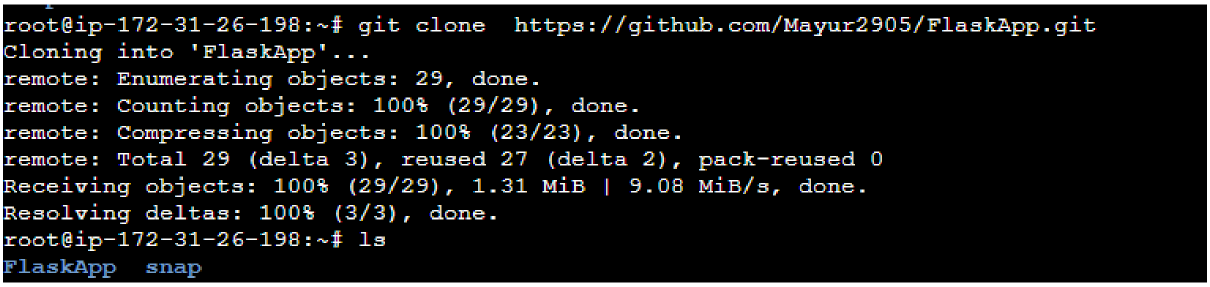
Login to AWS Console. Start your docker instance.

Connect the aws docker instance.

Step 2.

Clone the project from the git repo

-git clone<repo urel>

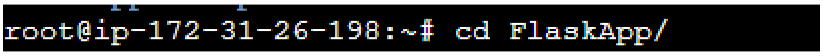


```
root@ip-172-31-26-198:~# git clone  https://github.com/Mayur2905/FlaskApp.git
Cloning into 'FlaskApp'...
remote: Enumerating objects: 29, done.
remote: Counting objects: 100% (29/29), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 29 (delta 3), reused 27 (delta 2), pack-reused 0
Receiving objects: 100% (29/29), 1.31 MiB | 9.08 MiB/s, done.
Resolving deltas: 100% (3/3), done.
root@ip-172-31-26-198:~# ls
FlaskApp  snap
```

Step 3.

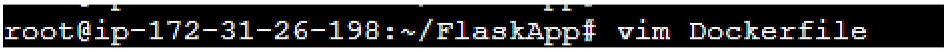
Create a Dockerfile in the project repository.

-cd FlaskApp



```
root@ip-172-31-26-198:~# cd FlaskApp/
```

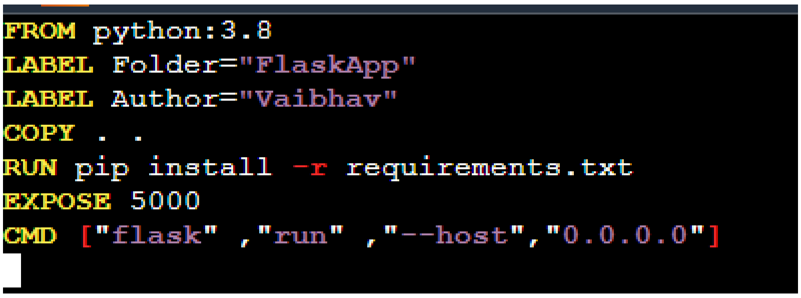
-vim Dockerfile



```
root@ip-172-31-26-198:~/FlaskApp# vim Dockerfile
```

Step 4.

Add the following code in the Dockerfile

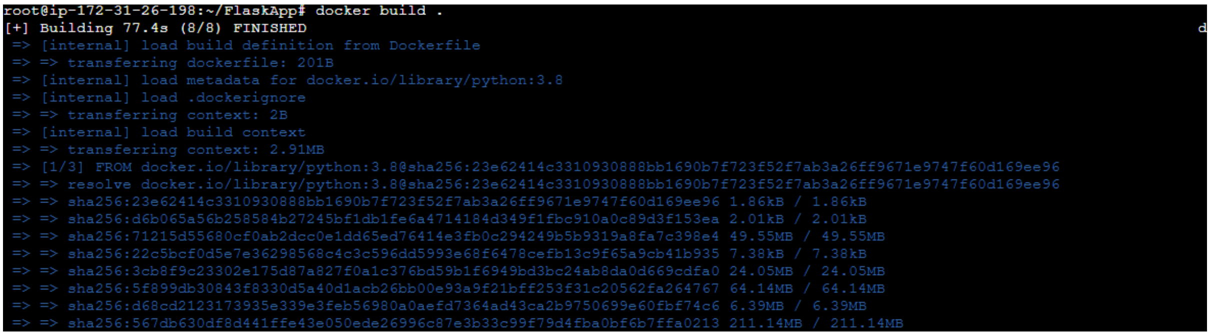


```
FROM python:3.8
LABEL Folder="FlaskApp"
LABEL Author="Vaibhav"
COPY . .
RUN pip install -r requirements.txt
EXPOSE 5000
CMD ["flask" ,"run" ,"--host","0.0.0.0"]
```

Step 5.

Create a docker image using docker build.

-docker build .

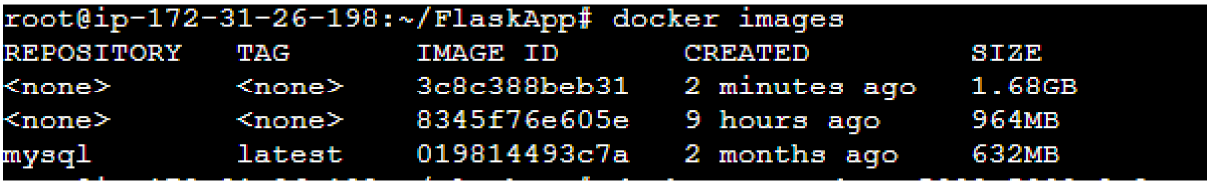


```
root@ip-172-31-26-198:~/FlaskApp# docker build .
[+] Building 77.4s (8/8) FINISHED                                                                                                        d
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 201B
 => [internal] load metadata for docker.io/library/python:3.8
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build context
 => => transferring context: 2.91MB
 => [1/3] FROM docker.io/library/python:3.8@sha256:23e62414c3310930888bb1690b7f723f52f7ab3a26ff9671e9747f60d169ee96
 => => resolve docker.io/library/python:3.8@sha256:23e62414c3310930888bb1690b7f723f52f7ab3a26ff9671e9747f60d169ee96
 => => sha256:23e62414c3310930888bb1690b7f723f52f7ab3a26ff9671e9747f60d169ee96 1.86kB / 1.86kB
 => => sha256:d6b065a56b258584b27245bf1db1fe6a4714184d349f1fbc910a0c89d3f153ea 2.01kB / 2.01kB
 => => sha256:71215d55680cf0ab2dcc0e1dd65ed76414e3fb0c294249b5b9319a8fa7c398e4 49.55MB / 49.55MB
 => => sha256:22c5bcf0d5e7e36298568c4c3c596dd5993e68f6478cefb13c9f65a9cb41b935 7.38kB / 7.38kB
 => => sha256:3cb8f9c23302e175d87a827f0a1c376bd59b1f6949bd3bc24ab8da0d669cdfa0 24.05MB / 24.05MB
 => => sha256:5f899db30843f8330d5a40d1acb26bb00e93a9f21bff253f31c20562fa264767 64.14MB / 64.14MB
 => => sha256:d68cd2123173935e339e3feb56980a0aefd7364ad43ca2b9750699e60fbf74c6 6.39MB / 6.39MB
 => => sha256:567db630df8d441ffe43e050ede26996c87e3b33c99f79d4fba0bf6b7ffa0213 211.14MB / 211.14MB
```

Step 6.

List the images using

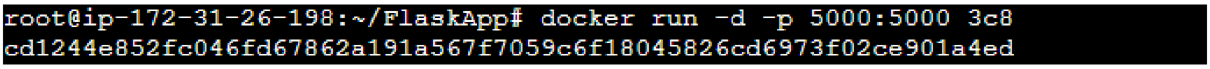
-docker images



```
root@ip-172-31-26-198:~/FlaskApp# docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
<none>        <none>    3c8c388beb31   2 minutes ago    1.68GB
<none>        <none>    8345f76e605e   9 hours ago      964MB
mysql         latest    019814493c7a   2 months ago     632MB
```

Step 7 : Create a container using the created image.

–docker run -d -p 5000:5000 <image id>

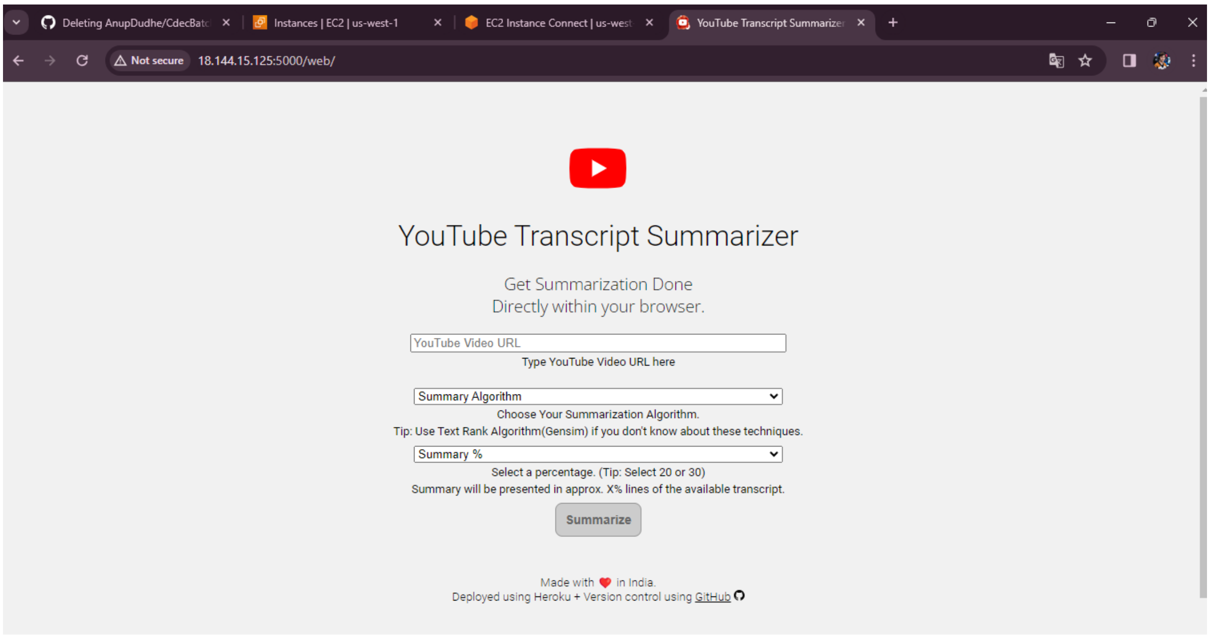


```
root@ip-172-31-26-198:~/FlaskApp# docker run -d -p 5000:5000 3c8
cd1244e852fc046fd67862a191a567f7059c6f18045826cd6973f02ce901a4ed
```

Step 8: Check the application

http://<ip>:5000

In my case it is http://18.144.15.125:5000/web/



Deleting AnupDudhe/CdecBatc
Instances | EC2 | us-west-1
EC2 Instance Connect | us-west
YouTube Transcript Summarizer
Not secure
18.144.15.125:5000/web/

YouTube Transcript Summarizer

Get Summarization Done
Directly within your browser.

YouTube Video URL

Type YouTube Video URL here

Summary Algorithm

Choose Your Summarization Algorithm.

Tip: Use Text Rank Algorithm(Gensim) if you don't know about these techniques.

Summary %

Select a percentage. (Tip: Select 20 or 30)

Summary will be presented in approx. X% lines of the available transcript.

Summarize

Made with ❤ in India.

Deployed using Heroku + Version control using GitHub