Data Glacier

Data Science Intern at Data Glacier

Week 5: Deployment on Heroku

Name: Syed Sanaullah Shah

Batch Code: LISUM11: 30 June - 30 Sept 2022

Date: 30 Sept 2022

Submitted to: Data Glacier

Code Link:

Table of Contents:

Introduction

To deploy our webapp as an api and as a webapp, wee need to establish the environment. Following are the steps to deploy flask model on Heroku Server:

1. App preparation with API endpoints
2. Setup of Heroku account & CLI
3. App Deployment on Heroku
4. Postman ‘Get’ check up

1. API Endpoints Preparation

app.py | Procfile | requirements.txt

```
30
31
32  def clf_predict(img):
33
34      y_pred = model.predict(img)
35      y_pred = y_pred.argmax(axis=-1)
36      pred_class = CATEGORIES[y_pred[0]]
37
38      return pred_class
39
40
41  # Home page
42  @app.route('/', methods=['POST', 'GET'])
43  def index():
44      # Main page
45      return render_template('index.html')
46
47
48
49  # VGG home page
50  @app.route('/vgg/',methods = ['POST', 'GET'])
51  def vgg():
52
53
54      return render_template('vgg.html')
55
56
```

In the next few steps, we have install certain libraries and create a Procfile as follows in the terminal.

- pip install gunicorn
- Pip freeze > requirements.txt
- Procfile as follows:

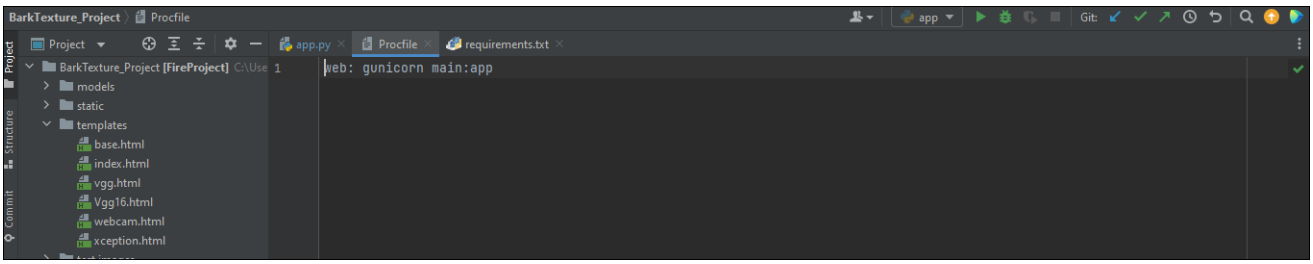

Figure: Procfile

2. Setup of Heroku account & CLI

We have to create an account on Heroku and install Heroku CLI for command line interface.

After installing the exe files, we are good to go after we see this in cmd.

```
C:\Users\User1>heroku
CLI to interact with Heroku

VERSION
  heroku/7.53.0 win32-x64 node-v12.21.0

USAGE
  $ heroku [COMMAND]

COMMANDS
  access          manage user access to apps
  addons          tools and services for developing, extending, and operating your app
  apps            manage apps on Heroku
  auth            check 2fa status
  authorizations  OAuth authorizations
  autocomplete    display autocomplete installation instructions
  buildpacks      scripts used to compile apps
  certs           a topic for the ssl plugin
  ci              run an application test suite on Heroku
  clients         OAuth clients on the platform
  config          environment variables of apps
  container       Use containers to build and deploy Heroku apps
  domains         custom domains for apps
  drains          forward logs to syslog or HTTPS
  features        add/remove app features
  git             manage local git repository for app
  help            display help for heroku
  keys            add/remove account ssh keys
  labs            add/remove experimental features
  local           run Heroku app locally
  logs            display recent log output
  maintenance     enable/disable access to app
  members         manage organization members
  notifications   display notifications
  orgs            manage organizations
  pg              manage postgresql databases
  pipelines       manage pipelines
  plugins         list installed plugins
  ps              Client tools for Heroku Exec
  psql            open a psql shell to the database
```

3. App Deployment on Heroku

Now that we are ready, we have to navigate to our project directory. We will create empty git repository meanwhile we will create an app on Heroku as follows:

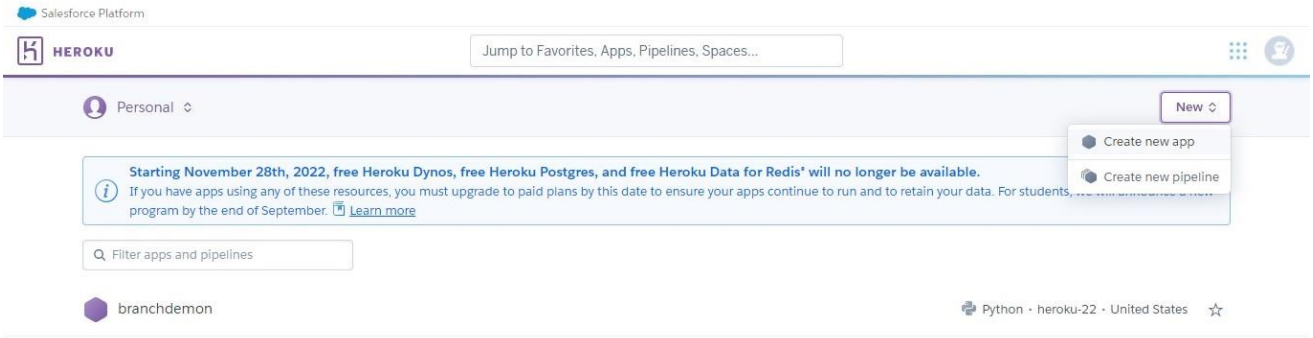

Figure: Creating an app

In cmd, we will write few git commands to add and push the file to deploy on Heroku as follows:

- heroku login
- cd to project dir
- git add –all
- git commit -m “comment”
- heroku git:remote -a “app name”
- git push heroku master

  App is deployed and you see the logs in case of errors.

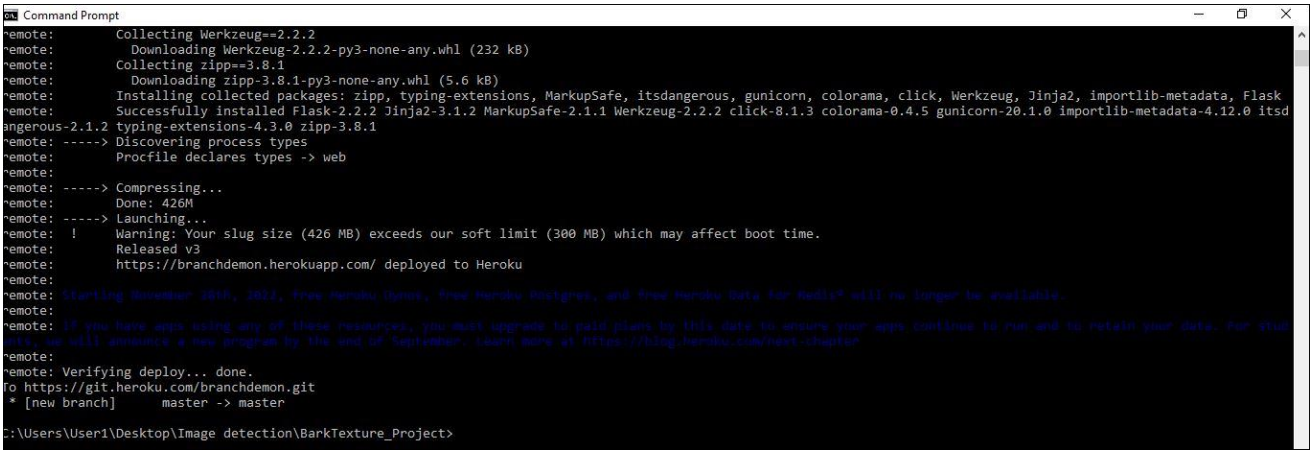

Figure: Flask app deployed via CMD

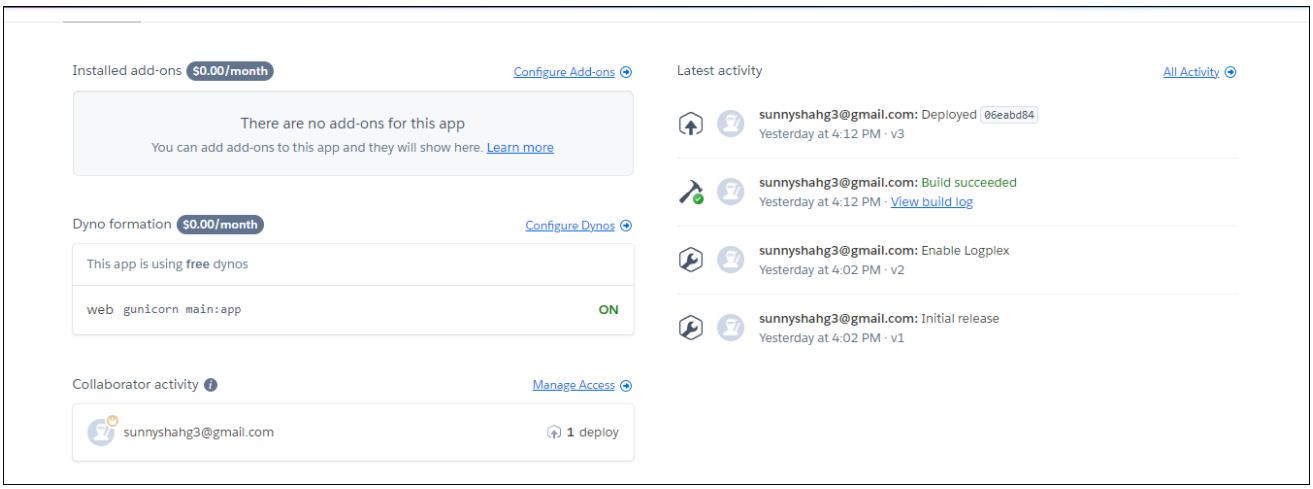

Figure: Heroku dashboard showing our app got deployed.

4. Postman API Check

We have to install Postmann as an app. Pass in the url in the panel to check the status as follows: