

Data Glacier

Week 5: Cloud and API deployment

Name: Loh Wan Teng

Batch Code: LISUM11

Submission Date: 5 August 2022

Submitted to: Data Glacier

Table of Content

1. Model Deployment on Heroku
 - 1.1. Create new repo
 - 1.2. Clone repo to local
 - 1.3. Copy Week 4 files to local
 - 1.4. Generate **requirement.txt**
 - 1.5. Upload all files to GitHub
 - 1.6. Create new Heroku app
 - 1.7. Connect Heroku app with GitHub repo
 - 1.8. Deploy GitHub repo

1. Model Deployment on Heroku

1.1. Copy Week 4 files to local

A folder is created with the name **flask-depoly-heroku-demo**. The Week 4 files are copied to then copy into the folder.

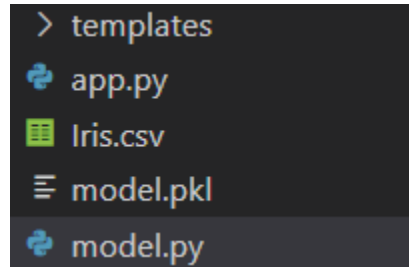


Figure 1: Copy Week 4 files to local

1.2. Create new repo

A new repo called **flask-deploy-heroku-demo** is initialised in local and is created on GitHub.

1.3. Generate requirement.txt

We first install **pipreqs** with `pip install pipreqs`. Then, we generate the **requirement.txt** using `pipreqs .`

```
PS C:\Users\WT\OneDrive\Desktop\GitHub\ml-model-flask-app-deployment-demo> pip install pipreqs
Collecting pipreqs
  Downloading pipreqs-0.4.11-py2.py3-none-any.whl (32 kB)
Collecting docopt
  Downloading docopt-0.6.2.tar.gz (25 kB)
  Preparing metadata (setup.py) ... done
Collecting yarg
  Downloading yarg-0.1.9-py2.py3-none-any.whl (19 kB)
Requirement already satisfied: requests in c:\users\wt\appdata\local\programs\python\python310\lib\site-packages (from yarg->pipreqs) (2.28.1)
create mode 100644 model.ipynb
create mode 100644 model.pkl
create mode 100644 requirements.txt
create mode 100644 templates/index.html
```

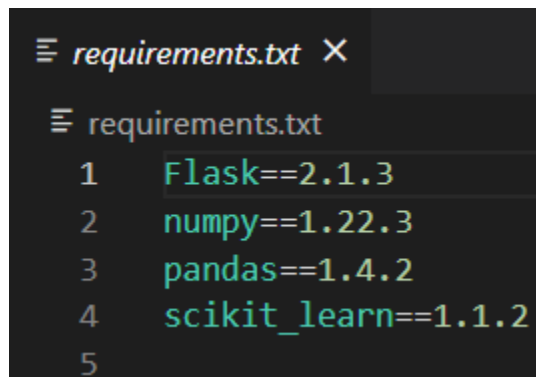


Figure 2: Generate requirement.txt

1.4. Push local to remote

Then, perform `git add`, `git commit` and `git push remote`.

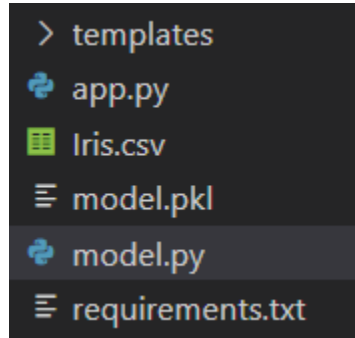


Figure 3: Push local to remote

1.5. Create new Heroku app

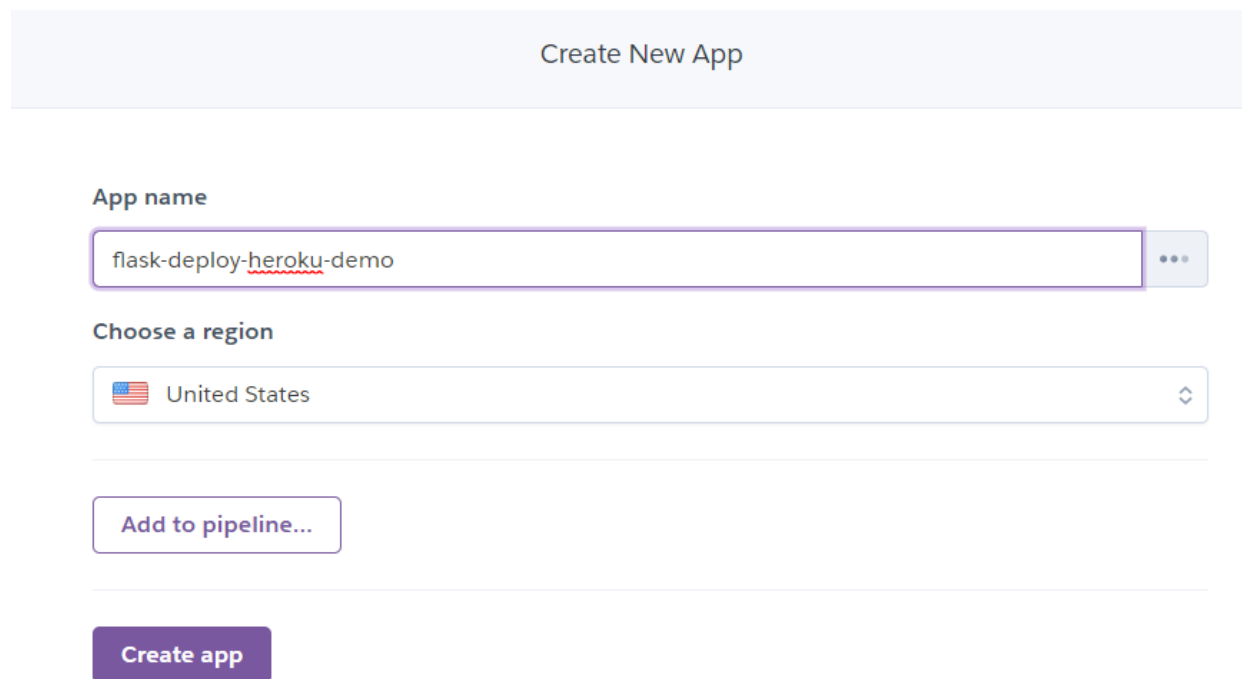
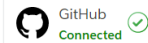
A screenshot of the Heroku 'Create New App' form. At the top is a light blue header with the text 'Create New App'. Below this is the 'App name' section, which has a text input field containing 'flask-deploy-heroku-demo' and a three-dot menu icon to its right. Underneath is the 'Choose a region' section, featuring a dropdown menu with a US flag icon and the text 'United States', and a small expand/collapse arrow on the right. Below the region selection is a button labeled 'Add to pipeline...'. At the bottom of the form is a large purple button labeled 'Create app'.

Figure 4: Create new Heroku app

1.6. Connect Heroku app with GitHub repo

Deployment method



App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to [Michelle-Lohwt/flask-deploy-heroku-demo](#) by [Michelle-Lohwt](#)

Disconnect...

Releases in the [activity feed](#) link to GitHub to view commit diffs

Figure 5: Connect Heroku app with GitHub repo

1.7. Deploy GitHub repo


Manual deploy

Deploy the current state of a branch to this app.

Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

 main

Deploy Branch

Receive code from GitHub



Build main 1f267046



Release phase



Deploy to Heroku



Your app was successfully deployed.

[View](#)

Figure 6: Deploy GitHub repo

The app is deployed on <https://flask-deploy-heroku-demo.herokuapp.com/>.