

Week 5: Cloud Deployment on Heroku

Name: Han-Fu Lin

Batch Code: LISUM14

Submission Date: 05 Nov 2022

Submitted to: Data Glacier

Step 1: Test the Flask API

```
1 import numpy as np
2 import pickle
3 from flask import Flask, request, render_template
4
5 app = Flask(__name__)
6 model = pickle.load(open('/Users/han-fulin/VC/Week 4/Game.sav', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14     flag = False
15     if request.method == "POST":
16         bedroom = int(request.form.get('bedroom'))
17         bathroom = int(request.form.get('bathroom'))
18         surface = float(request.form.get('surface'))
19         longitude = float(request.form.get('longitude'))
20         latitude = float(request.form.get('latitude'))
21         ptype = int(request.form.get('ptype'))
22
23         raw_features = [bathroom, bedroom, surface, longitude, latitude, ptype]
24         features = [np.array(raw_features)]
25
26         prediction = model.predict(features)
27         output = round(prediction[0], 2)
28         return render_template('index.html', flag=True, prediction_text=f'House price should be €{output}.')
29
30
31
32
```

Step 2: Save the model

main

VC / Week 4 /

Go to file

Add file

...

This branch is 19 commits ahead of DataGlacier:main.

Contribute

Sync fork

HankData

Add files via upload

541c715 10 days ago

History

..

Game.sav

Add files via upload

10 days ago

Gaming pdf

Create Gaming pdf

10 days ago

Presentation of Data.key

Add files via upload

10 days ago

Presentation of Data.pdf

Add files via upload

10 days ago

Step 3: Create a Branch for heroku deploy and create profile for it

Last login: Mon Nov 7 15:48:31 on ttys000

The default interactive shell is now zsh.

To update your account to use zsh, please run `chsh -s /bin/zsh`.

For more details, please visit <https://support.apple.com/kb/HT208050>.

Han-Fude-MacBook-Pro:~ han-fulin\$ git model checkout - model deploy

The screenshot shows the CircleCI web interface for a project named 'HankData'. The left sidebar contains navigation links: Dashboard, Projects, Insights, Organization Settings, Plan, Notifications, Status (OPERATIONAL), Docs, Orbs, and Support. A notification banner at the top of the sidebar reads: 'CI behind your firewall just got easier. Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less.' The main content area displays the configuration page for 'HankData/VC'. A message at the top states: 'We'll commit the config below to HankData/VC on a new branch called circleci-project-setup. If you prefer, you can use an existing config.' Below this, there are buttons for 'Change: Python', 'Use Existing Config', and 'Commit and Run'. The configuration file content is shown in a dark-themed editor with line numbers 1 through 22. The configuration includes comments about the CircleCI pipeline engine, version 2.1, and the use of the 'python' orb. It also defines a job named 'build-and-test' with a Docker executor. The configuration ends with a comment: '# Change the version below to your required version of python'. At the bottom, a green status bar indicates 'config.yml is valid and ready to commit'.

```
1 # Use the latest 2.1 version of CircleCI pipeline process engine.
2 # See: https://circleci.com/docs/2.0/configuration-reference
3 version: 2.1
4
5 # Orbs are reusable packages of CircleCI configuration that you may share across projects, enabling you to create encaps
6 # See: https://circleci.com/docs/2.0/orb-intro/
7 orbs:
8   # The python orb contains a set of prepackaged CircleCI configuration you can use repeatedly in your configuration fil
9   # Orb commands and jobs help you with common scripting around a language/tool
10  # so you dont have to copy and paste it everywhere.
11  # See the orb documentation here: https://circleci.com/developer/orbs/orb/circleci/python
12  python: circleci/python@1.5.0
13
14 # Define a job to be invoked later in a workflow.
15 # See: https://circleci.com/docs/2.0/configuration-reference/#jobs
16 jobs:
17   build-and-test: # This is the name of the job, feel free to change it to better match what you're trying to do!
18     # These next lines defines a Docker executors: https://circleci.com/docs/2.0/executor-types/
19     # You can specify an image from Dockerhub or use one of the convenience images from CircleCI's Developer Hub
20     # A list of available CircleCI Docker convenience images are available here: https://circleci.com/developer/images/1
21     # The executor is the environment in which the steps below will be executed - below will use a python 3.10.2 contain
22     # Change the version below to your required version of python
```

HankData
HankData

Dashboard

Projects

Insights

Organization Settings

Plan

CI behind your firewall

just got easier

Install a more scalable, Kubernetes-friendly self-hosted runner in 5 minutes or less.

Notifications

Status OPERATIONAL

Docs

Orbs

Dashboard

Project

All Pipelines > VC

VC Add team members

Edit Config

Trigger Pipeline

Project Settings

Filters

Everyone's Pipelines

VC

All Branches

All days

Auto-expand

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
VC 2	Running	sample	circleci-project-setup 5f24142 Add_circleci/config.yml	13s ago	15s	Refresh Cancel Close More
VC 1	Running	sample	circleci-project-setup 5f24142	8s ago	15s	Refresh Cancel Close More

Step 4: Select the branch that application is written and deploy. Let heroku run install and deploy

<https://dublin-house-predict.herokuapp.com/>