

TÀI LIỆU HƯỚNG DẪN TRIỂN KHAI SẢN PHẨM

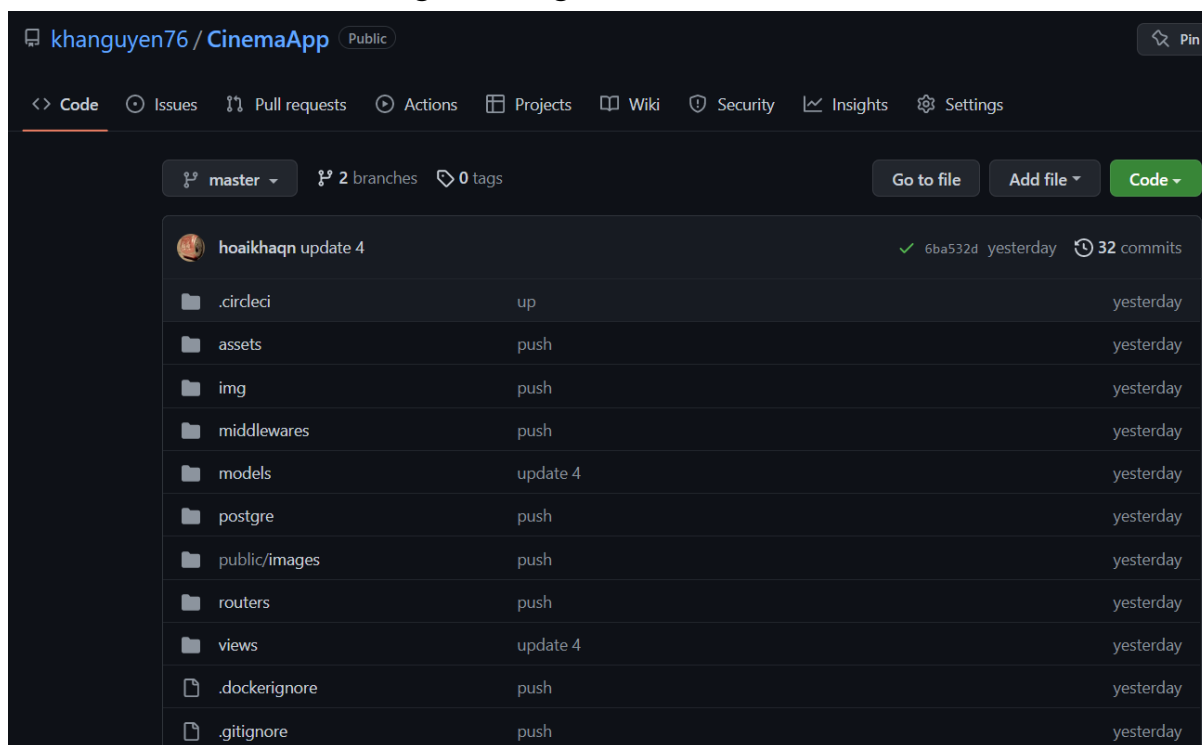
1. Các công cụ triển khai

Docker
Circle ci
Github
Heroku

2. Các bước cài đặt cấu hình

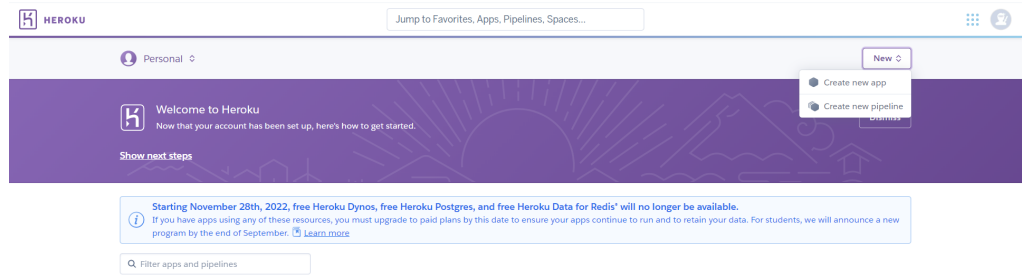
2.1. Quản lý mã nguồn ứng dụng với Github

- Đầu tiên, đăng ký tài khoản Github (nếu chưa có) và tạo một repo cho dự án, sau đó đưa mã nguồn lên github.

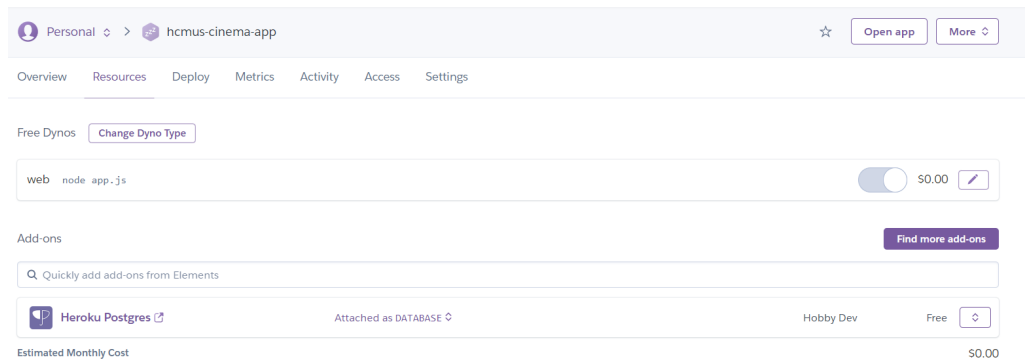


2.2. Tạo ứng dụng trên Heroku

- Đầu tiên, đăng ký tài khoản heroku (nếu chưa có) và tạo một app mới.

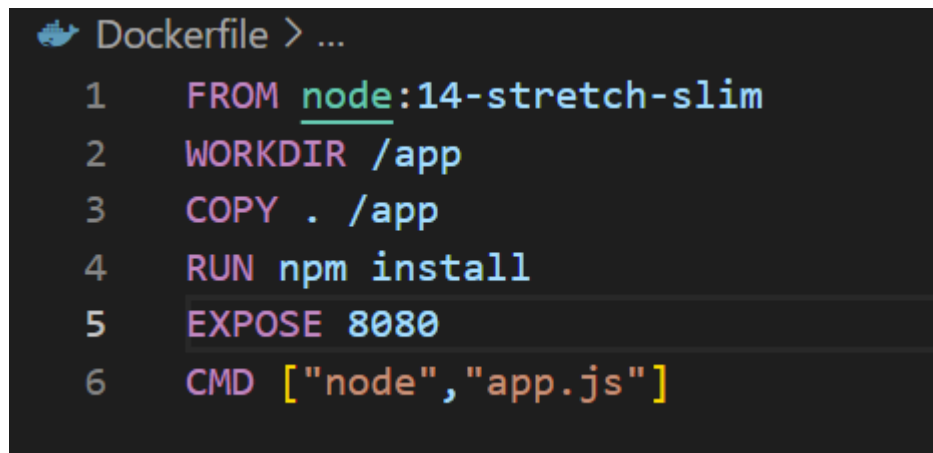


- **Tạo một database PostgreSQL**



2.3. **Viết unit test và tạo file Dockerfile cho mã nguồn**

- Viết Dockerfile và unit test. Sau đó push code lên github.



2.4. **Tạo project trên Circle CI**

- Đăng ký tài khoản Circle CI thông qua tài khoản Github có repo source của trang web, tìm repo và chọn “Set up project”.

Projects

Repo name

Follow All

...

Repo

CinemaApp

Unfollow Project

...

- Cấu hình file config.yml

```
.circleci > ! config.yml
1  version: 2.1
2  orbs:
3    node: circleci/node@4.7
4    heroku: circleci/heroku@2.0.0
5  jobs:
6    testing:
7      machine: true
8      steps:
9        - checkout
10       - run:
11           name: Testing
12           command: |
13             npm install
14             npm test
15    build-deploy:
16      machine: true
17      steps:
18        - checkout
19        - run:
20           name: Build and push Docker image to Heroku
21           command: |
22             set -x
23             HEROKU_API_KEY=${HEROKU_PASSWORD} heroku container:login
24             HEROKU_API_KEY=${HEROKU_PASSWORD} heroku container:push -a hcmus-cinema-app web
25             HEROKU_API_KEY=${HEROKU_PASSWORD} heroku container:release -a hcmus-cinema-app web
26  workflows:
27    pipeline:
28      jobs:
29        - testing
30        - build-deploy:
31            requires:
32              - testing
33            filters:
34              branches:
35                only:
36                  - master
37
```

Kịch bản triển khai tự động trên Circle CI như sau:

- Code được push lên Github
- Circle CI tự động chạy các lệnh được cấu hình ở file config.
 - Test: Chạy lệnh test các unit test đã được viết trong mã nguồn. Nếu test thành công và không có lỗi thì sẽ hiện passed và đến bước tiếp theo. Ngược lại sẽ báo failed và gửi email lỗi đến các thành viên và dừng triển khai.
 - Build-Deploy: Bắt buộc phải passed bước test và code được push ở nhánh master. Nếu thỏa chạy các lệnh

build image Docker từ Dockerfile và push image lên registry của Heroku và run container.

- Heroku tự động cập nhật lại source mới.