****ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

**BÁO CÁO ĐỀ TÀI**

**continuous integration**

TRAVIS CI

Nhóm thực hiện: Họ tên MSSV

1. Nguyễn Thái Hòa 1642019
2. Hà Nguyễn Thái Học 1642021
3. Dương Tấn Huỳnh Phong 1642049
4. Nguyễn Xuân Phúc 1642051

Lớp: 16HCB

Môn học: Công cụ kiểm chứng phần mềm

Mục lục

[I. GIỚI THIỆU TRAVIS CI 3](#_Toc510240226)

[II. PHẠM VI HỔ TRỢ 3](#_Toc510240227)

[1. Hệ điều hành 3](#_Toc510240228)

[2. Ngôn ngữ lập trình 3](#_Toc510240229)

[III. CHỨC NĂNG CHÍNH 4](#_Toc510240230)

[IV. Demo Travis CI 5](#_Toc510240231)

[1. Integrate Travis CI cho Github 5](#_Toc510240232)

[2. Integrate Travis CI với Slack 15](#_Toc510240233)

[3. Automatically deploy cho heroku với Travis CI 18](#_Toc510240234)

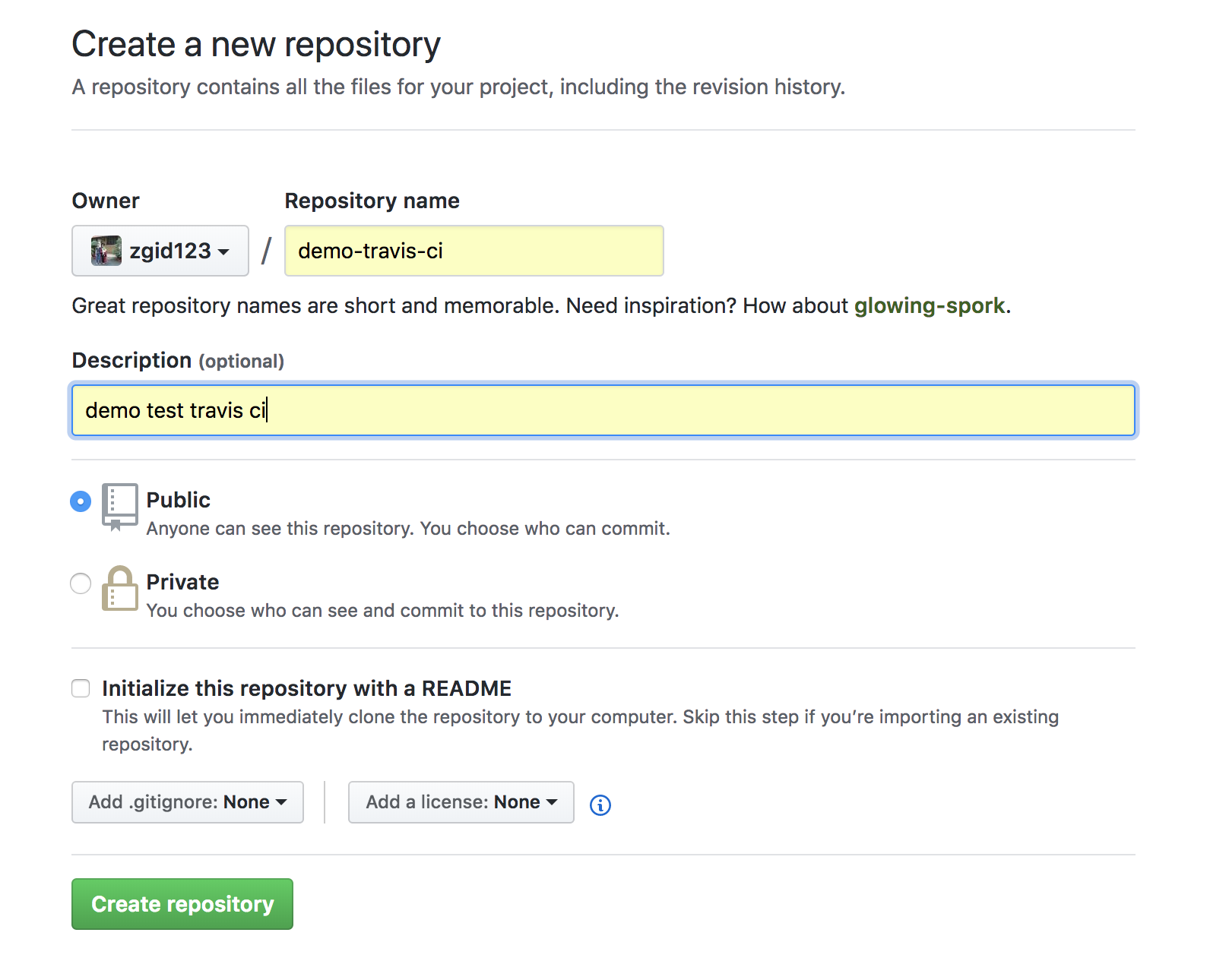
1. GIỚI THIỆU TRAVIS CI

Travis CI là một CI (continuous integration – tích hợp liên tục) giúp build, chạy unit test, deploy lên server một cách tự động mỗi khi push code mới hoặc tạo pull request.

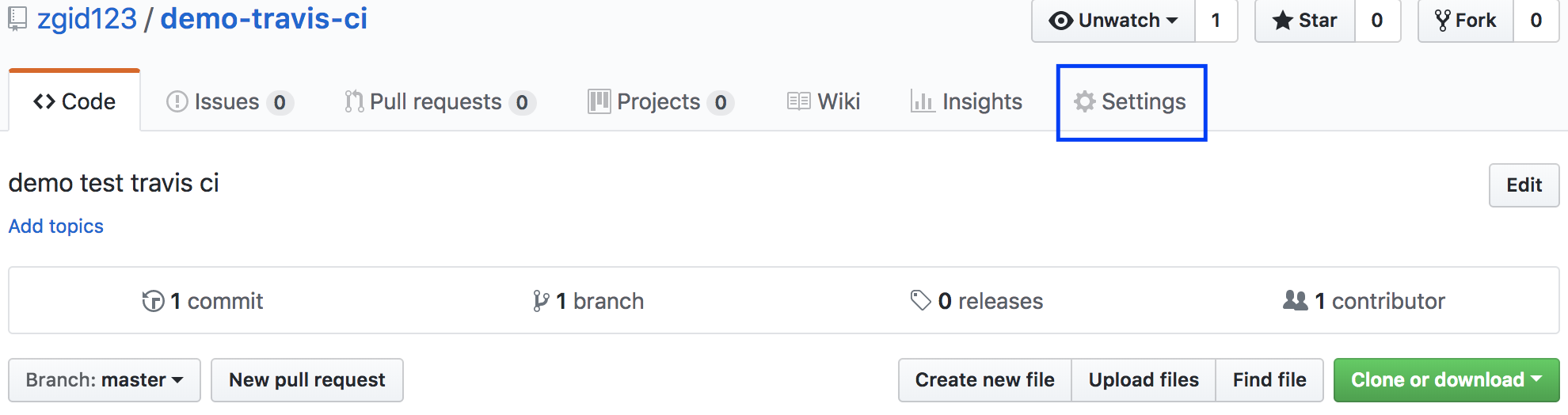
Travis CI thường được tích hợp chung với GitHub và Heroku để giúp trong việc build, chạy unit test và deploy sau khi build và chạy unit test thành công.

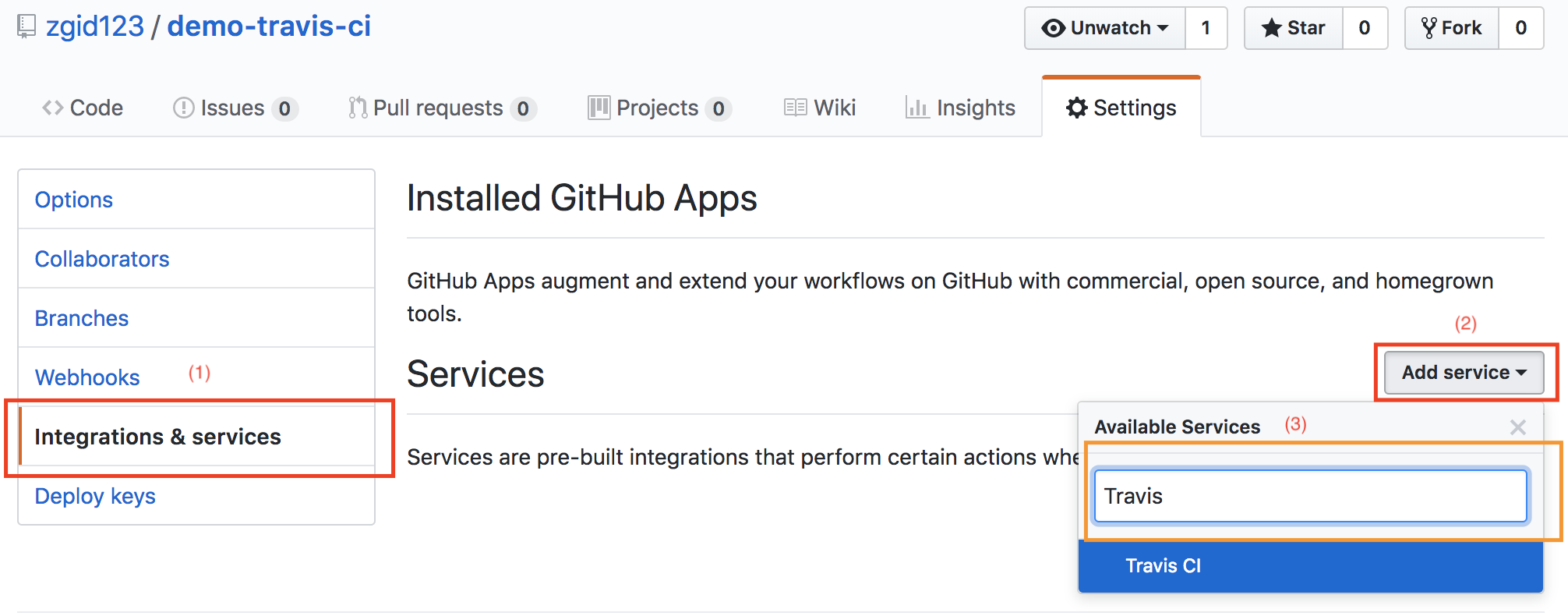
1. PHẠM VI HỔ TRỢ
   1. Hệ điều hành
      * + Ubuntu
        + MacOS
   2. Ngôn ngữ lập trình
      * + Android
        + C/C++
        + C#
        + Elixir
        + Go
        + Ruby
        + Java
        + JavaScript (Node.js only)
        + Objective-C
        + PHP
        + Python
        + Rust
        + Scala
        + Swift
        + Visual Basic
        + Perl
        + Perl6
        + F#
        + Clojure
        + Crystal
        + D
        + Dart
        + Erlang
        + Groovy
        + Haskell
        + Haxe
        + Julia
        + Nix
        + Smalltalk
2. CHỨC NĂNG CHÍNH
   * + - Build Project
         * Travis CI sẽ build project khi push code mới hoặc tạo pull request.
       - Run Unit Test
         * Travis CI sẽ run unit test khi push code mới hoặc tạo pull request.
         * Thông thường sẽ được cài đặt chạy sau khi build thành công.
       - Deploy Project
         * Travis CI sẽ deploy project lên host đã được cài đặt sẵn.
         * Thông thường sẽ được cài đặt chạy sau khi build và chạy unit test thành công.
       - Cron job
         * Travis CI cho phép tạo thời gian biểu (schedule) để chạy tự động một tiến trình nào đó trong một thời gian cụ thể.
       - Integration & Notification
         * Travis CI cho phép tích hợp vào bên thứ 3 như **Slack**, **GitHub**, **Heroku**, **…** và thông báo kết quả build, chạy unit test đến các bên thứ 3.
3. Demo Travis CI

Demo Travis CI bằng Ruby on Rails.

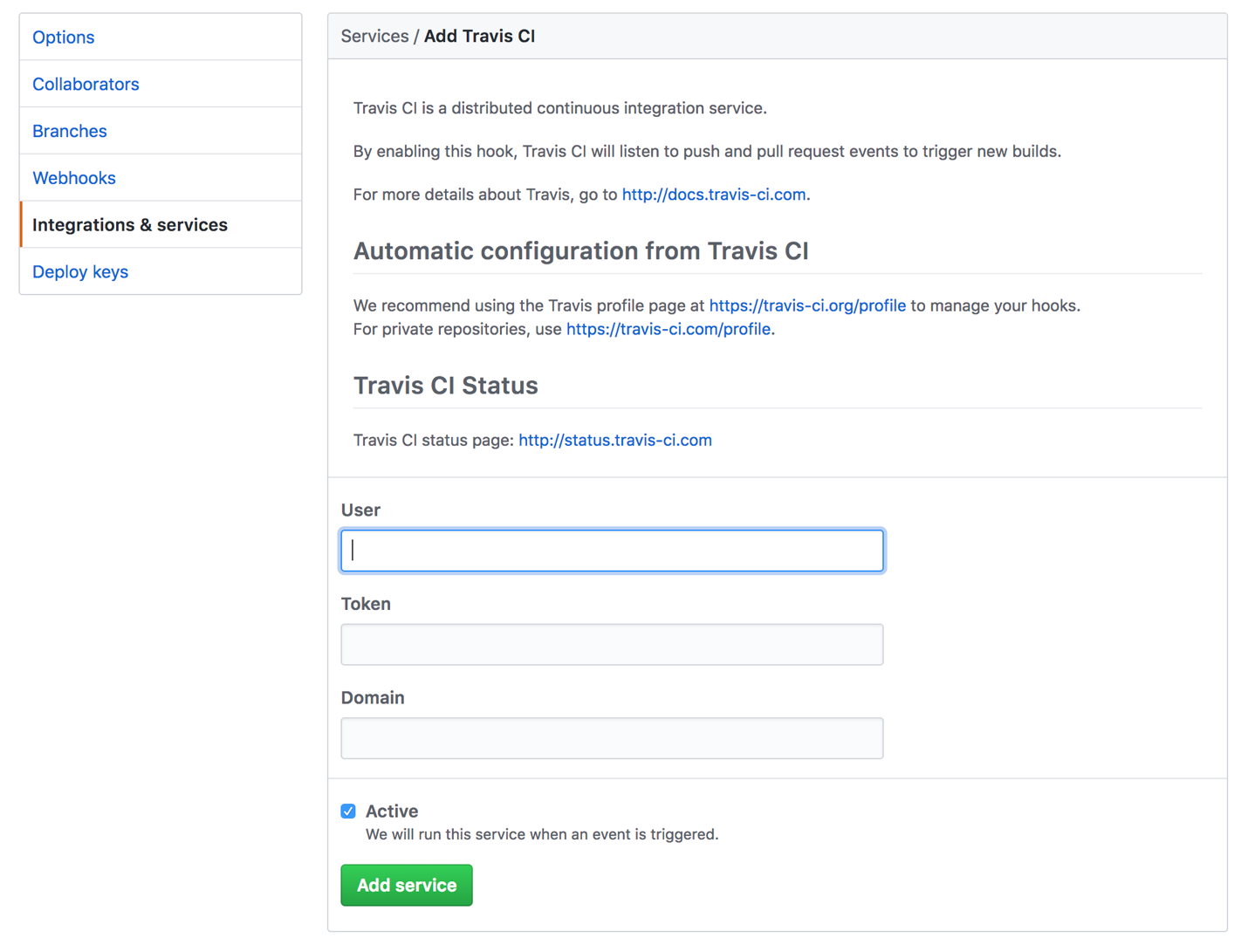
* 1. Integrate Travis CI cho Github
     + - Bước 1 – Tạo repository trên Github

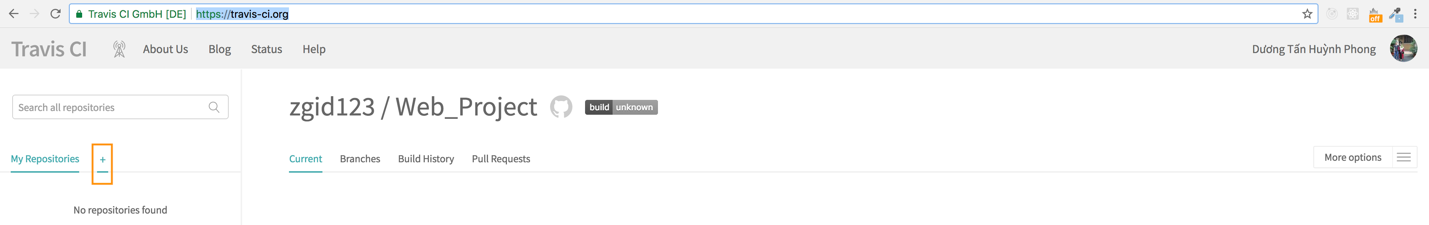
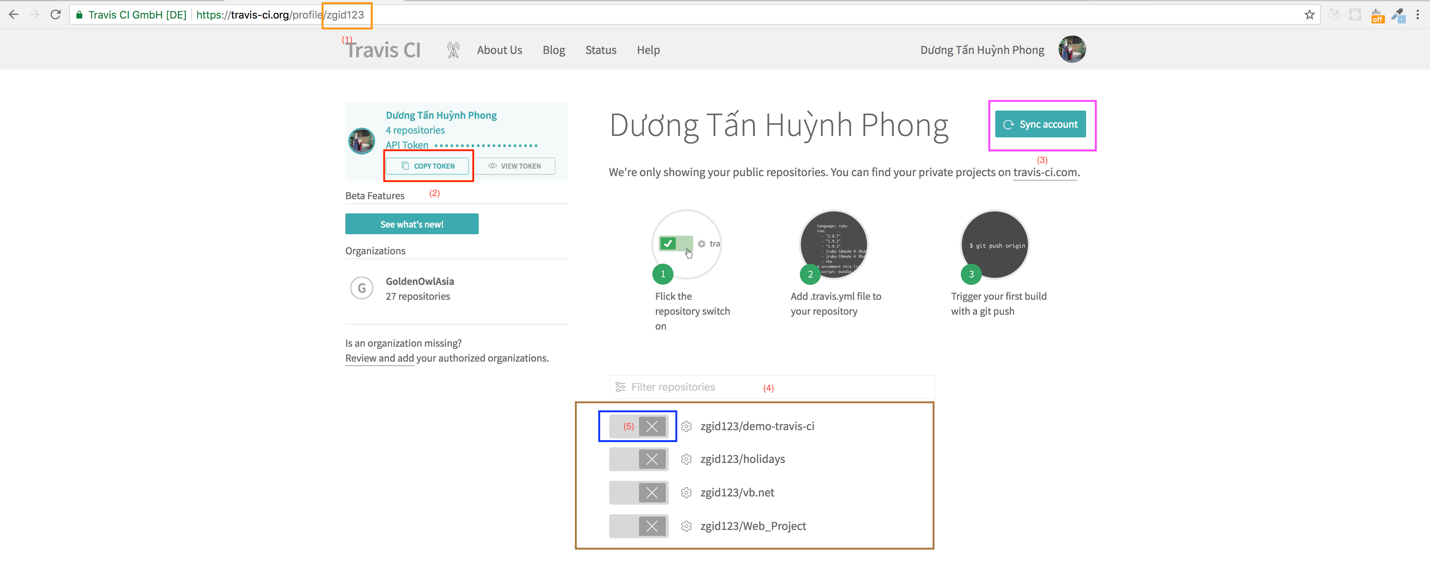
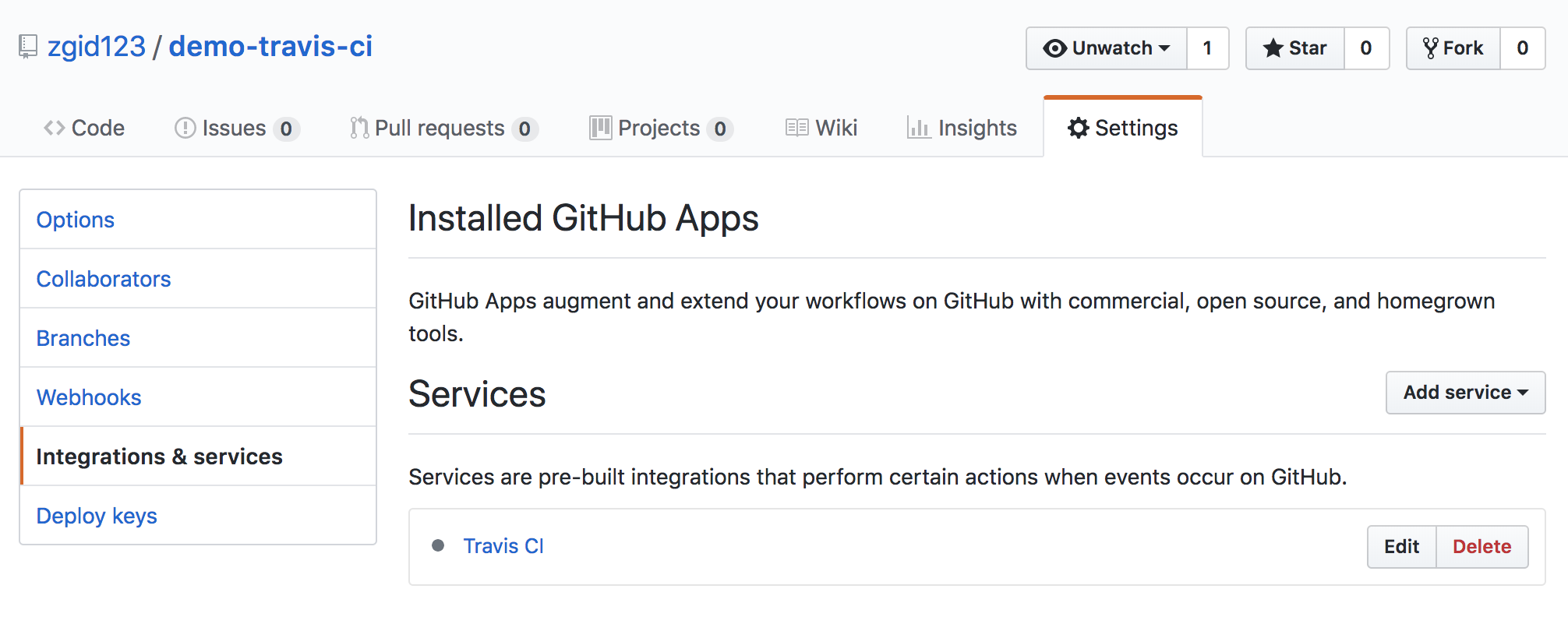
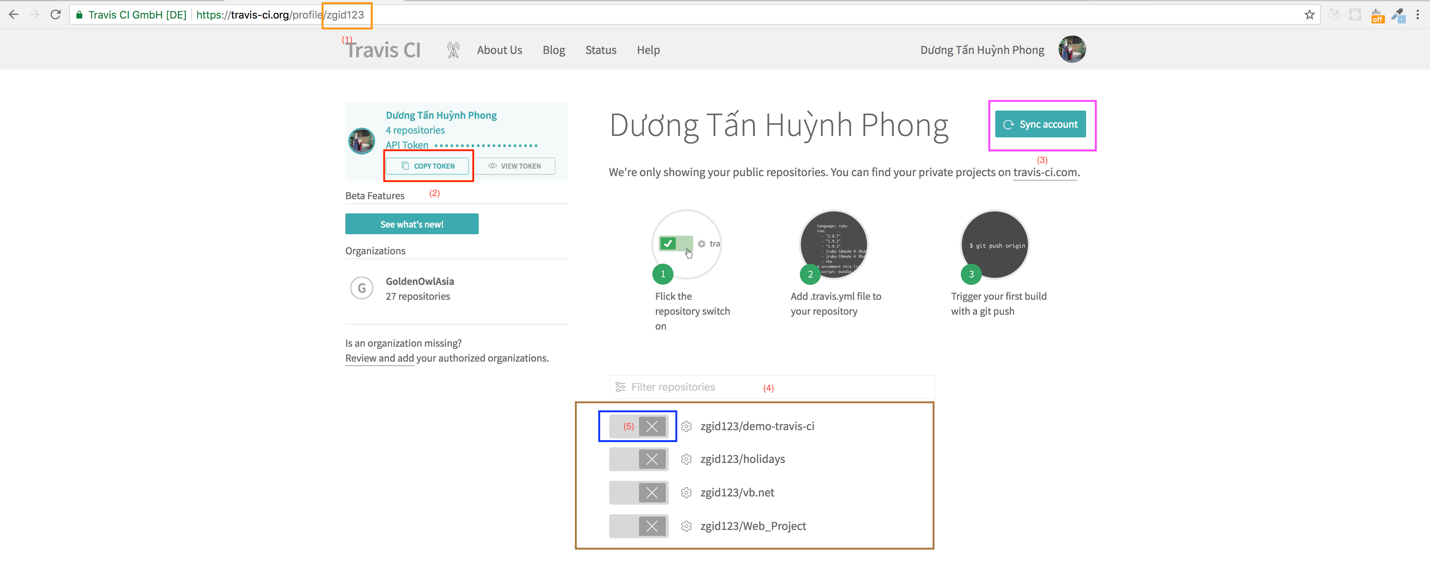
Sau đó push project lên Github.

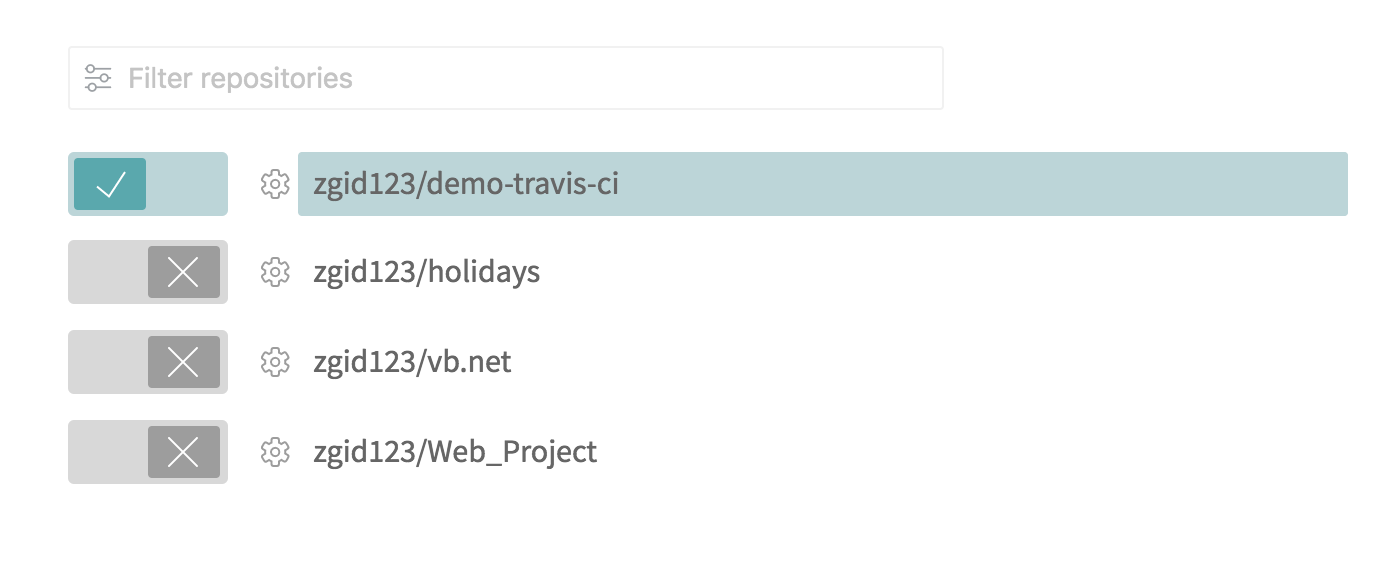
* + - * Bước 2 – Integrate Travis CI cho Github

Chọn mục Settings.

Chọn (1), sau đó chọn (2) và gõ **Travis** vào (3) và chọn kết quả search.

Để lấy **User** và **Token**, thực hiện các bước sau:

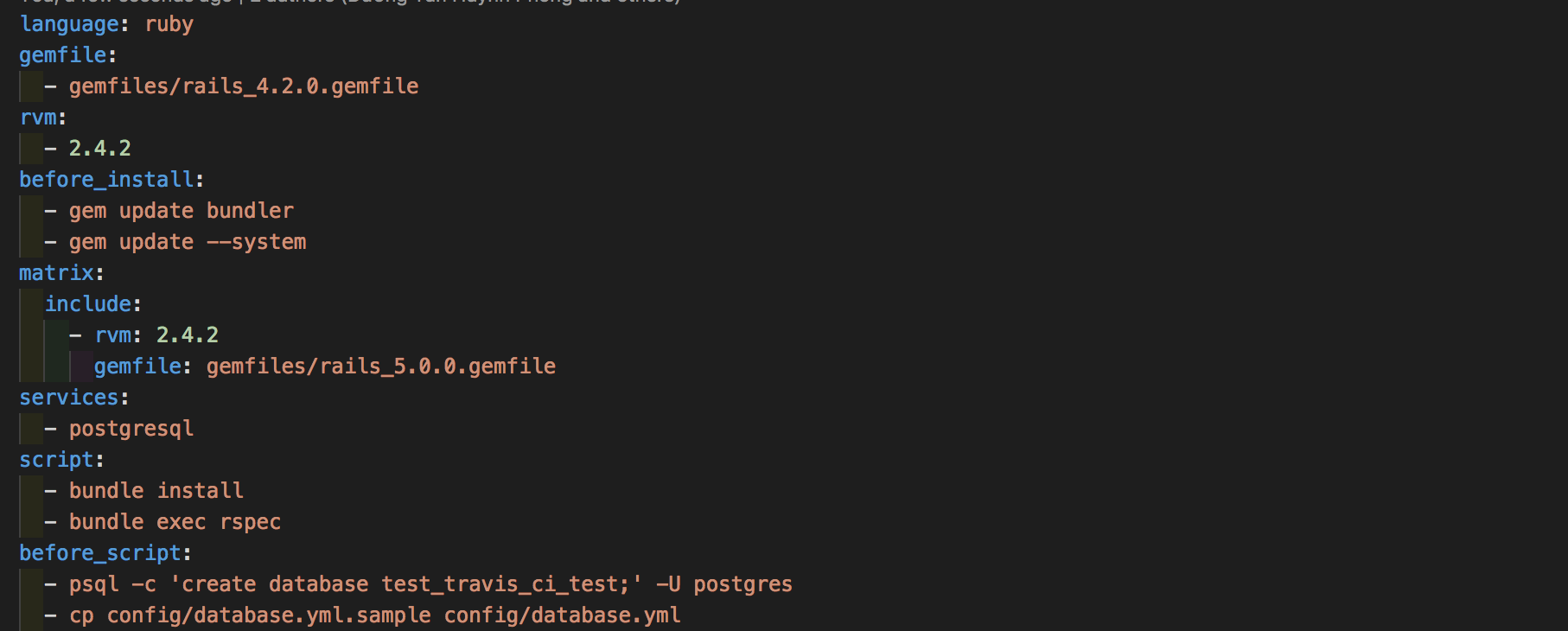
* + - * + Vào trang [**https://travis-ci.org/**](https://travis-ci.org/), đăng nhập tài khoản Travis.
        + Chọn **Add New Repository** như hình.
        + Giá trị (1) là **User**, nhấn vào (2) để copy **Token.**
        + Điền **User** và **Token** vào, sau đó nhấn nút **Add service**.
      * Bước 3 – Enable repository của Github trên Travis CI

(4) là danh sách các repositories của Github, chọn (5) ngay dòng của repository cần integrate với Github.

***Lưu ý****: Trường hợp nếu chưa thấy repository cần integrate, chọn (3) sau đó reload trang để refresh toàn bộ repositories, nếu vẫn chưa thấy thực hiện lại cho đến khi thấy được repository đó.*

* + - * Bước 4 – Config Travis CI cho project và cài đặt unit test và functional testing

Thực hiện các bước sau:

* + - * + Tạo file **.travis.yml** và thêm config sau:

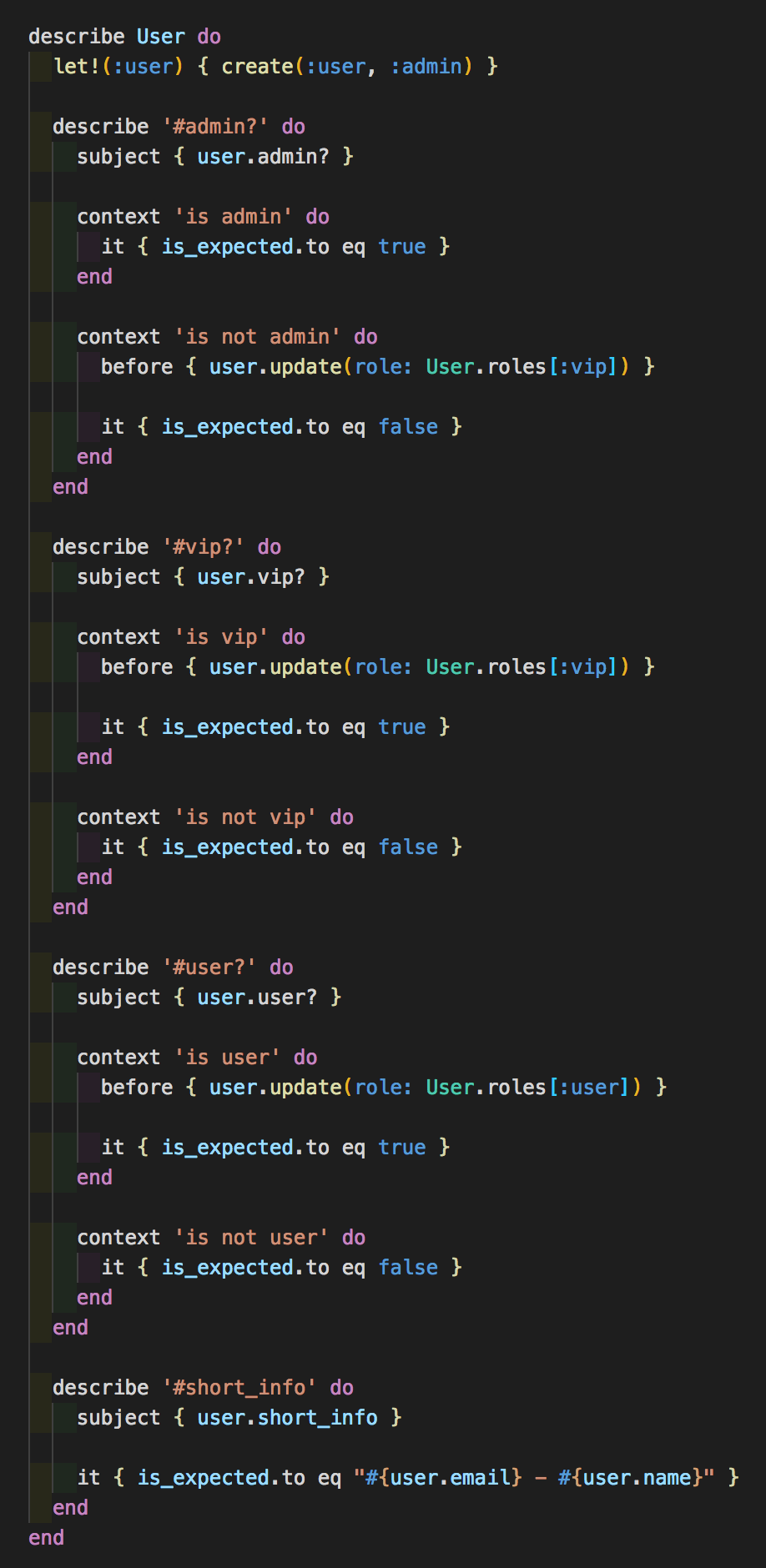
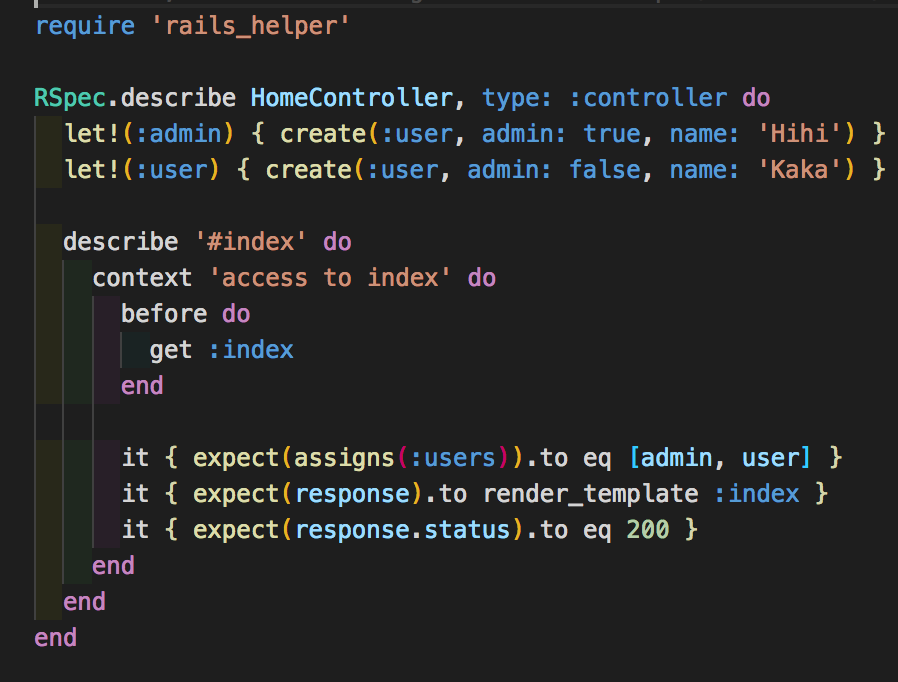
Language: là ngôn ngữ lập trình project đang sử dụng.

Before\_install: các lệnh chạy trước khi thực hiện Script.

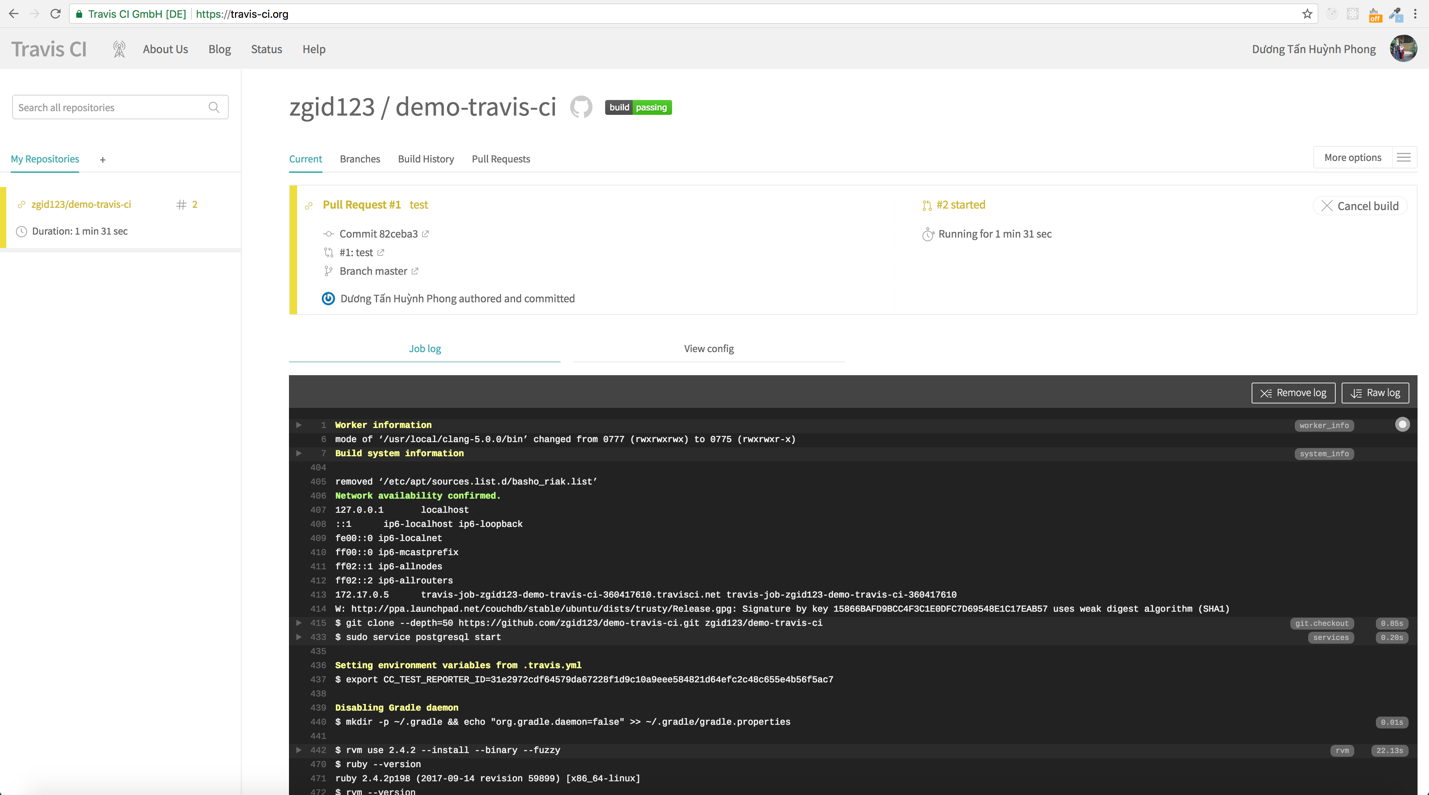
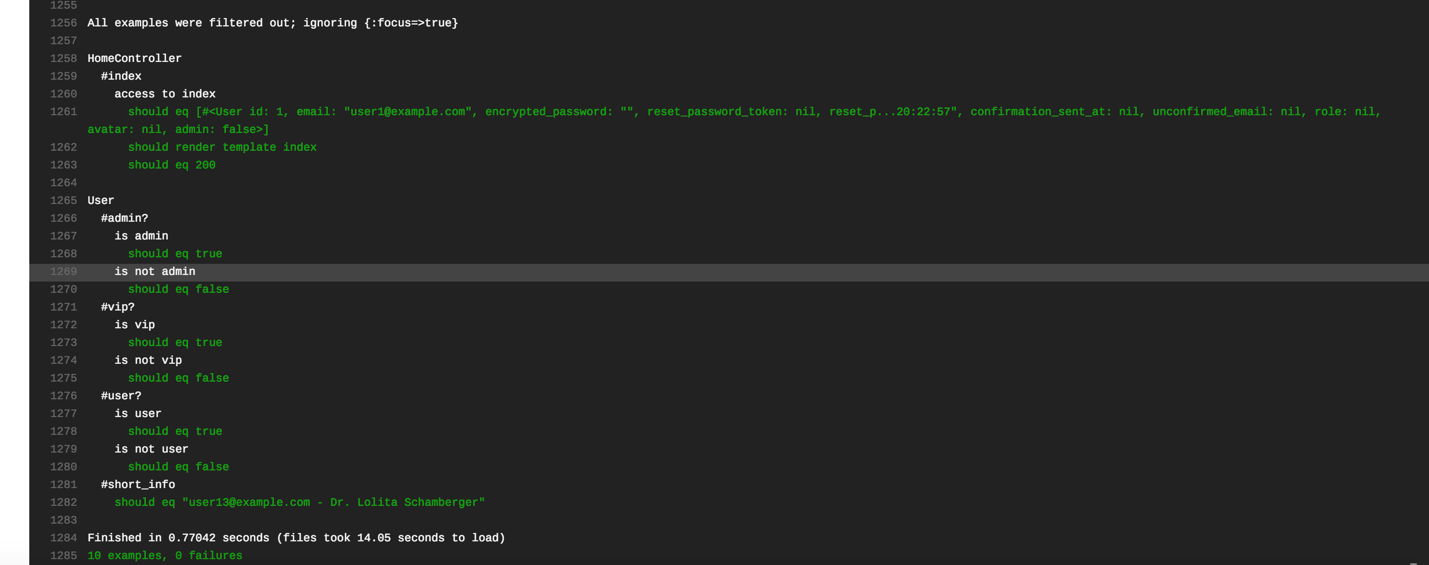
Services: là các dịch vụ chạy kèm, trong ví dụ là postgresql để chạy unit test và functional testing với data của postgresql.

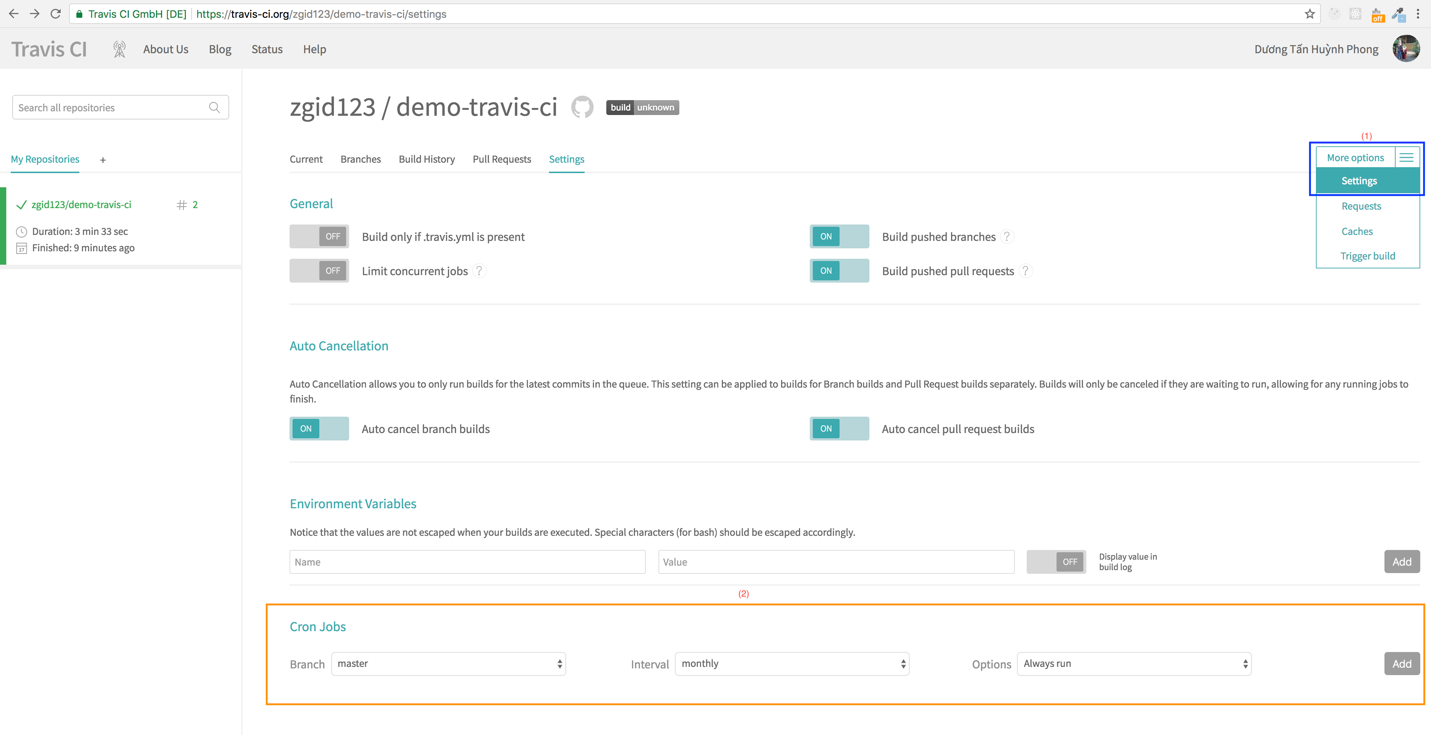
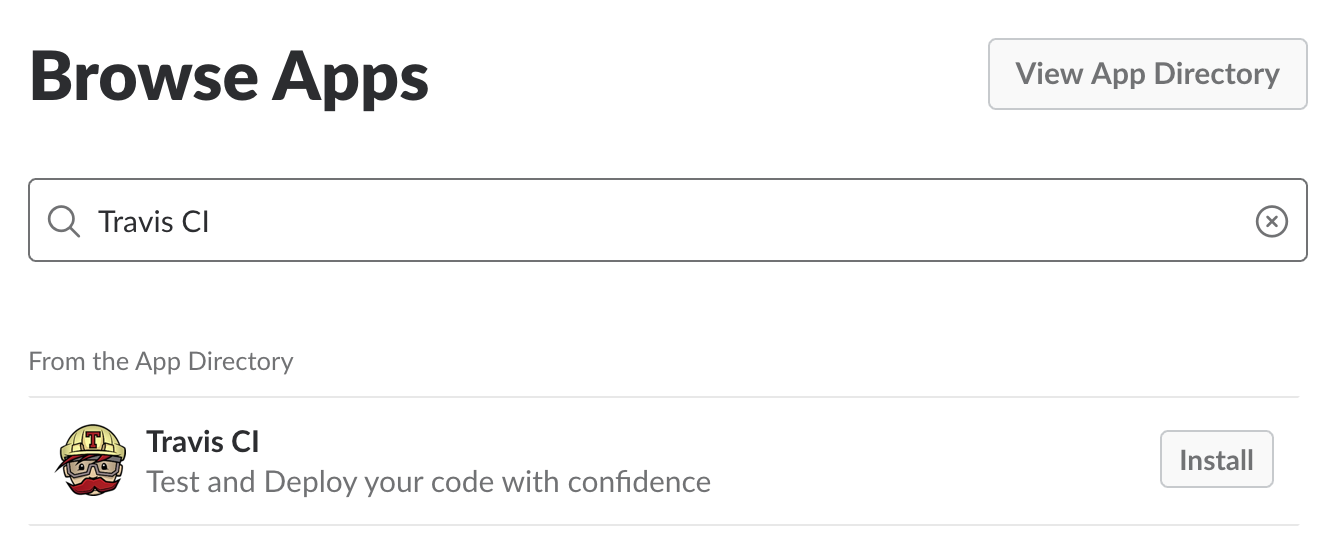
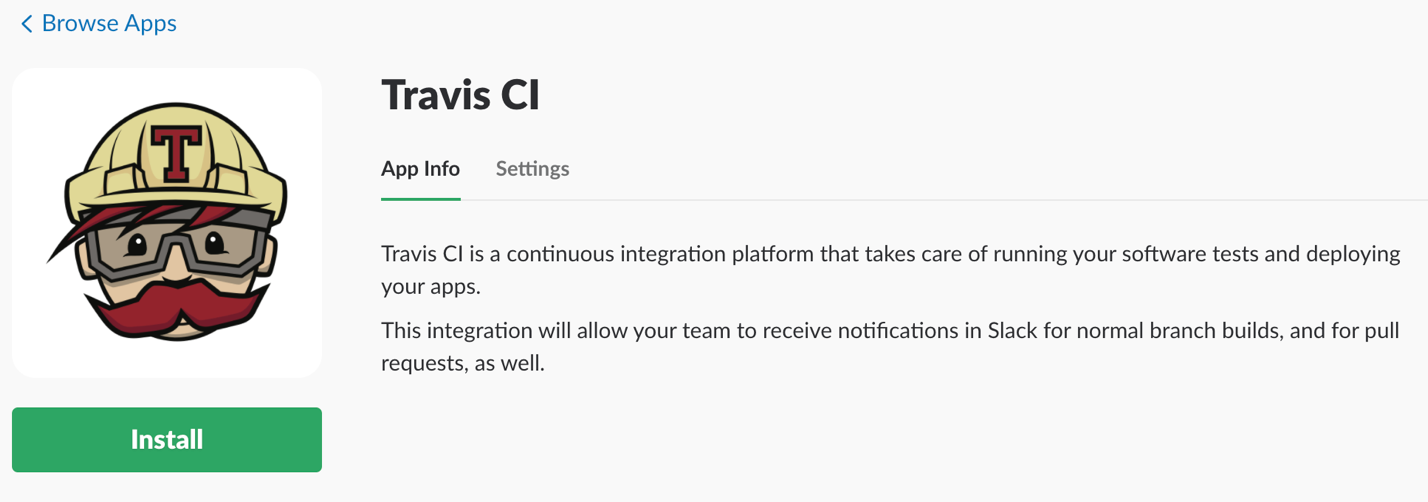
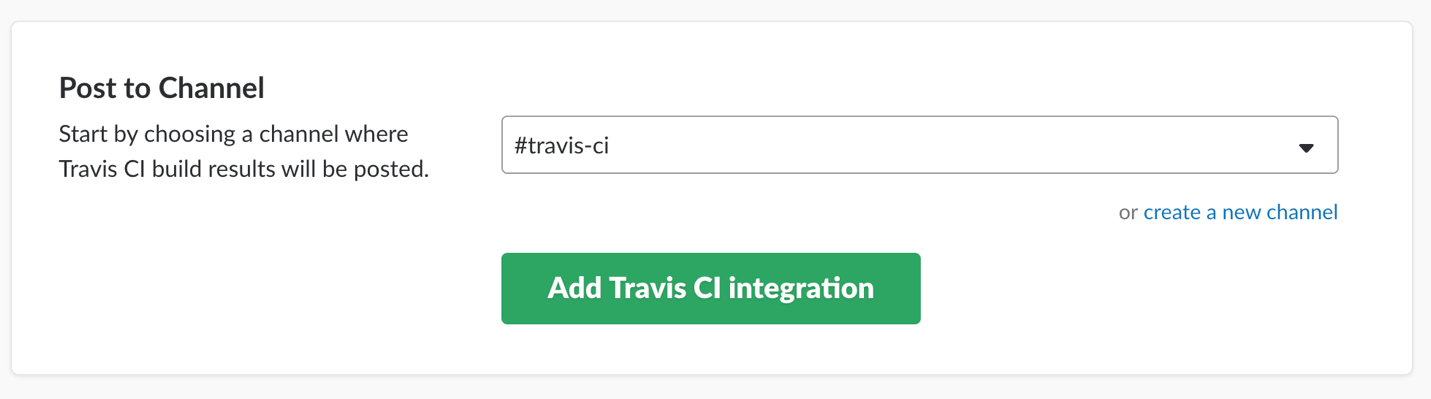
Script: là các lệnh chính cần chạy, trong ví dụ gồm ***bundle install*** – build project và ***bundle exec rspec*** – chạy unit test và functional testing.

Before\_script: là các lệnh chạy trước **script** và sau **before\_install**.

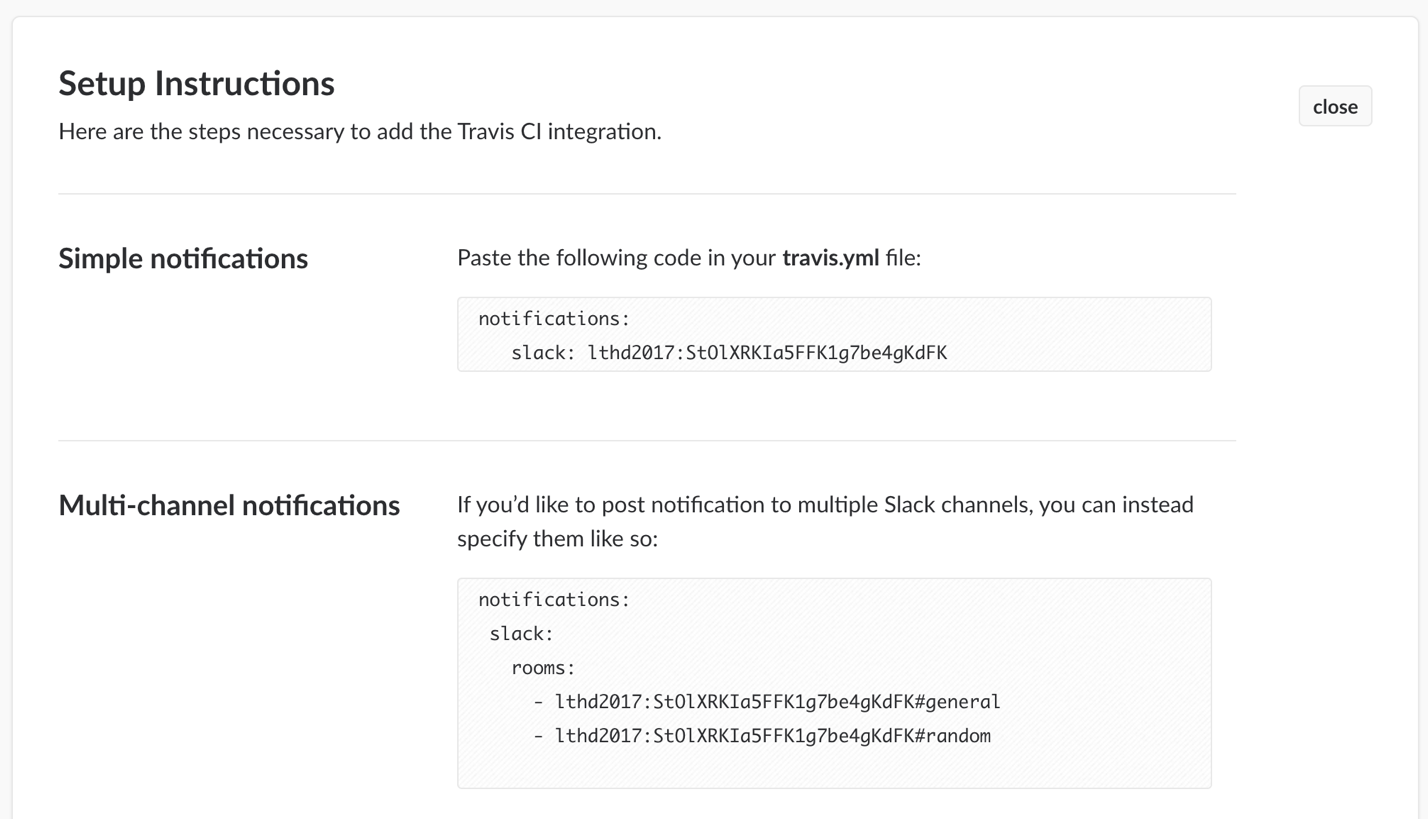
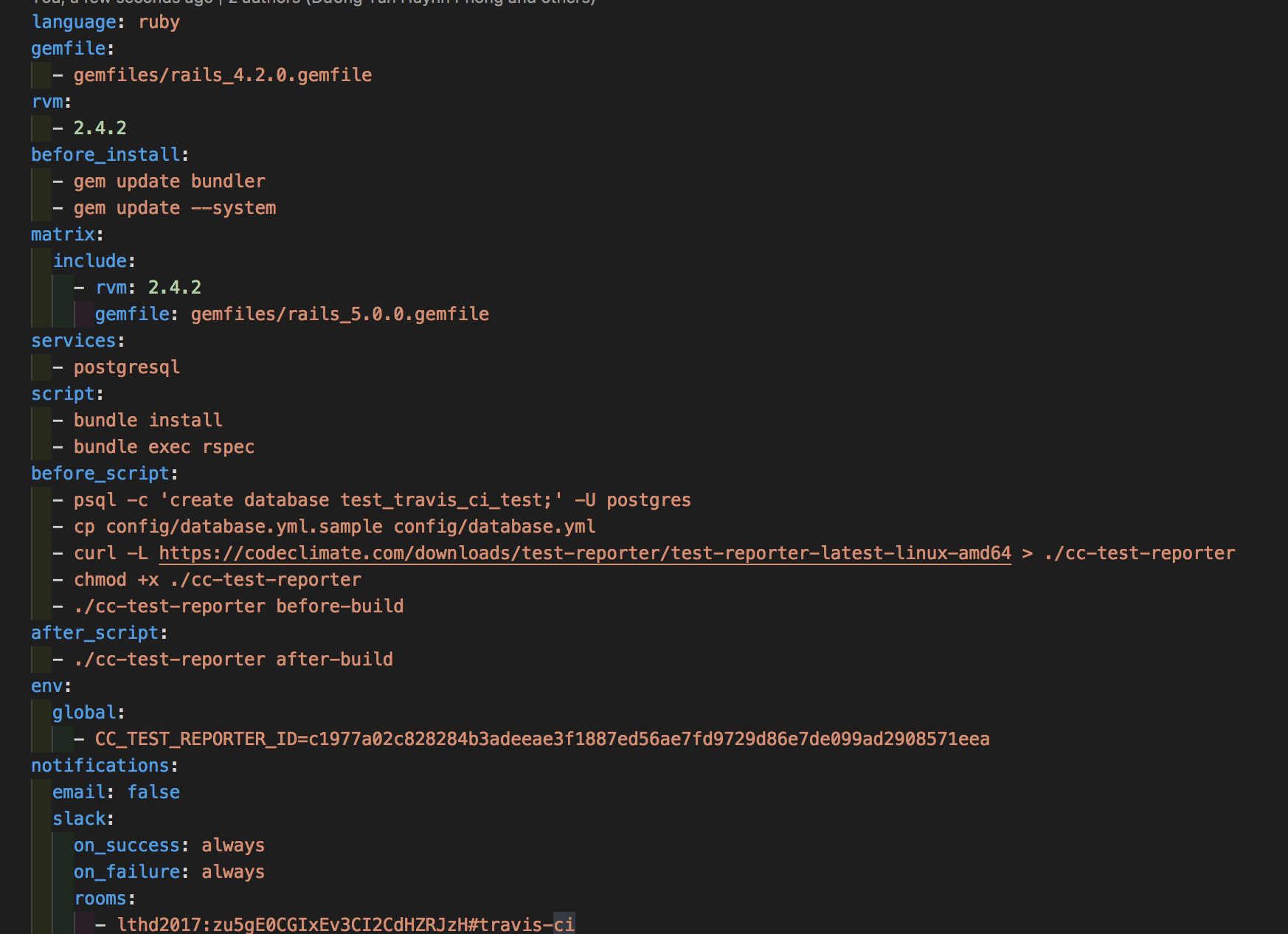
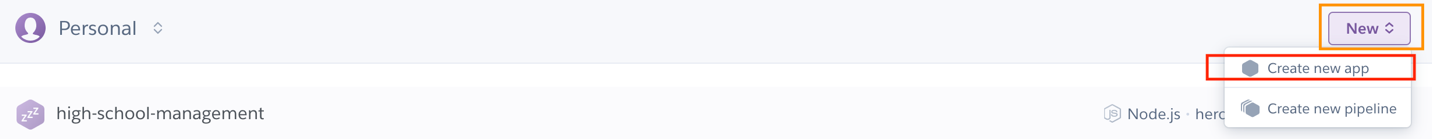
* + - * + Viết unit test
        + Viết functional testing
      * Bước 5 – Push code mới lên Github

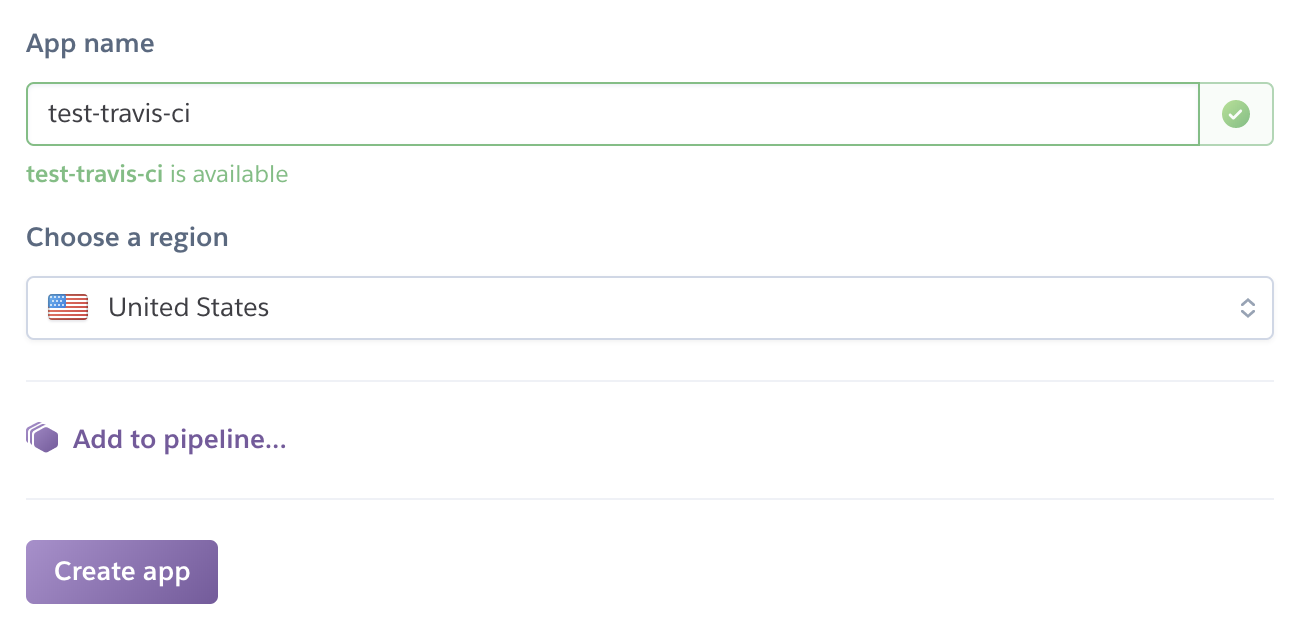
Sau khi thực hiện các bước trên, push code mới lên để Travis CI build project, chạy unit test và functional testing.

***Lưu ý****:**Travis CI sẽ chạy mỗi lần push code mới lên hoặc merge code vào branch master. Ngoài ra, Travis CI còn cho tạo cron job để build, chạy unit test và functional testing theo từng ngày, hoặc từng tháng hoặc mỗi tuần. Để sử dụng cron job, thực hiện như sau:*

* + - * + *Chọn (1) và chọn* ***Settings.***
        + *Chỉnh thông số ở (2) như: branch – tên branch muốn chạy cron job, interval – lịch biểu chạy cronjob, gồm 3 thông số* ***monthly*** *(chạy hàng tháng),* ***weekly*** *(chạy hàng tuần) và* ***daily*** *(chạy hàng ngày) và options gồm* ***Always run*** *– chỉ cần tới interval thì sẽ chạy build, unit test và functional testing, hoặc* ***Do not run if there has been a build in the last 24h*** *– nếu trong vòng 24 tiếng đồng hồ có một build nào đó, thì cron job sẽ không chạy khi đến interval.*
  1. Integrate Travis CI với Slack
     + - Bước 1 – Cài đặt **Travis CI** cho slack

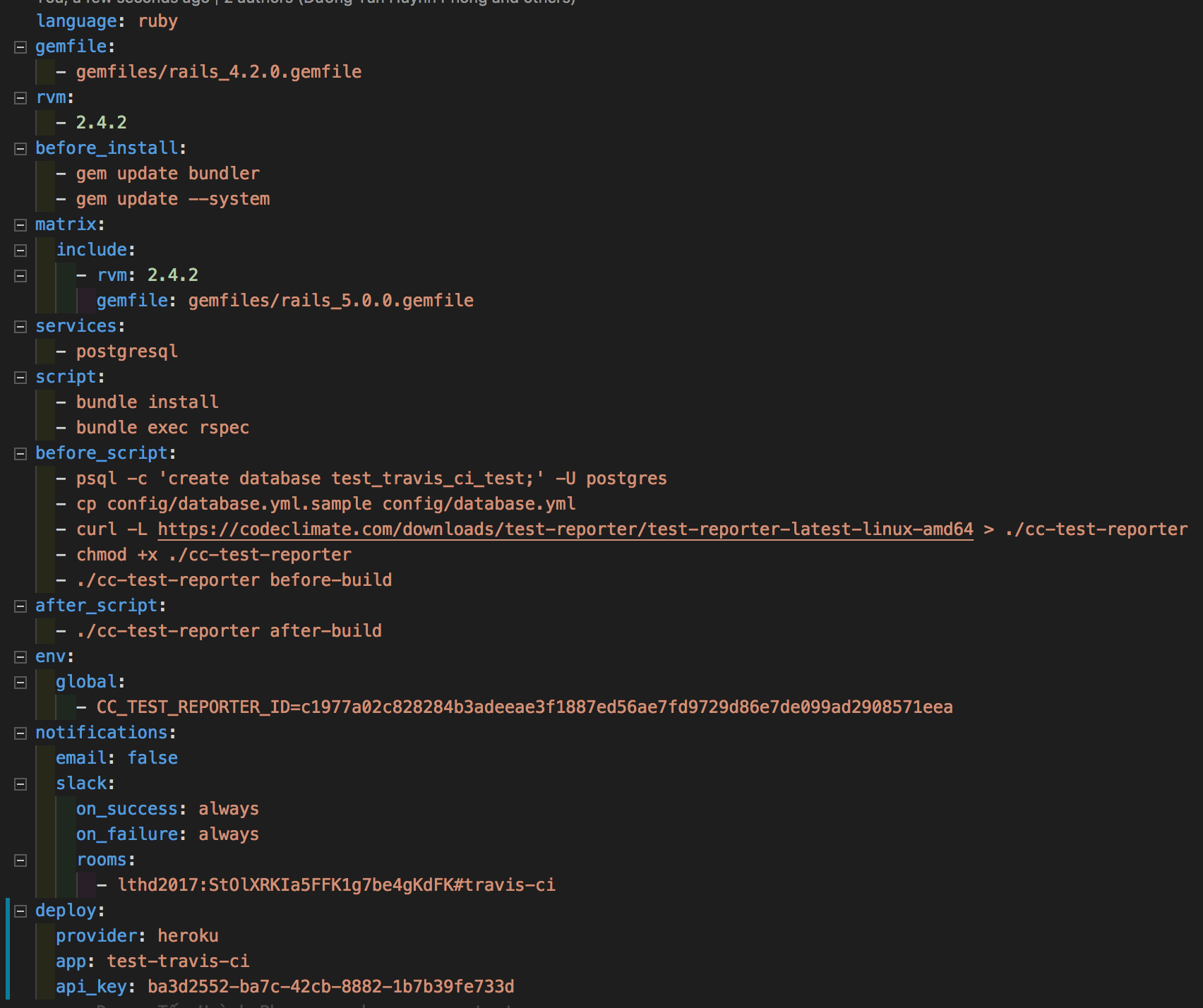
Chọn channel để integrate và nhấn nút **Add Travis CI integration**.

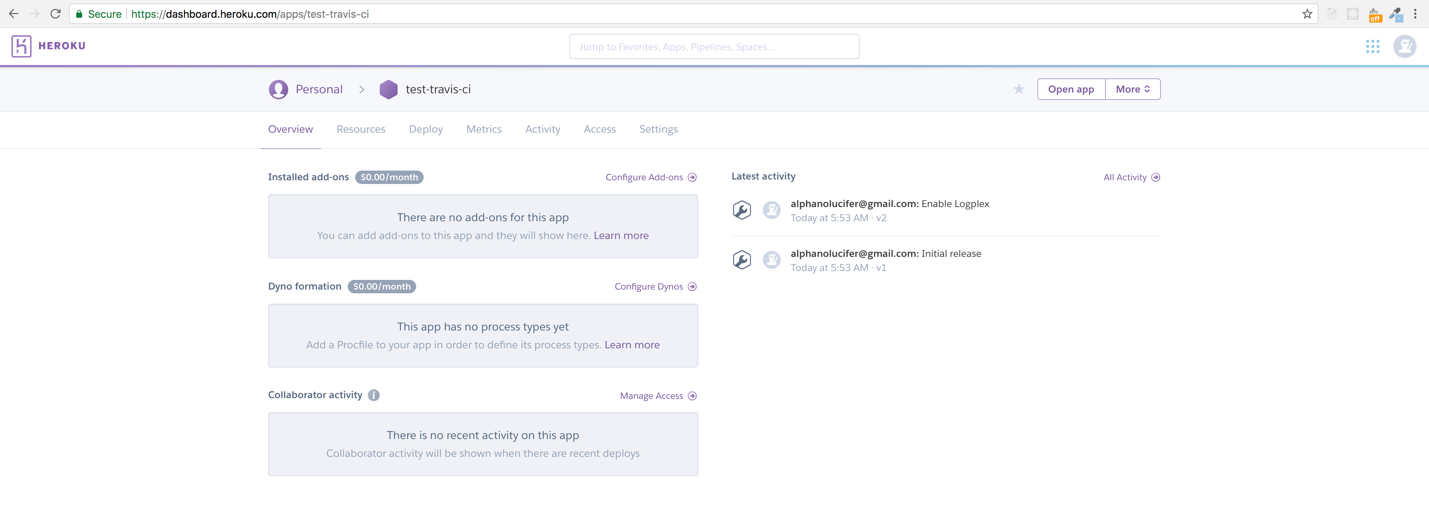
* + - * Bước 2 – Copy config vào file **.travis.yml**.
        + Email: Nếu không muốn **Travis CI** gửi mail sau khi chạy build, unit test và functional testing xong, thì set là **false**.
        + Rooms: là token của config phía trên.
        + On\_success: set **always** để khi mỗi lần success, sẽ push thông báo lên slack, hoặc **never** để không gửi thông báo, hoặc **change** để thông báo mỗi lần status thay đổi.
        + On\_failure: tương tự như **on\_success** và áp dụng khi fail.
      * Bước 4 – Push code mới lên để kiểm tra kết quả.
  1. Automatically deploy cho heroku với Travis CI­
     + - Bước 1 – Tạo Heroku app

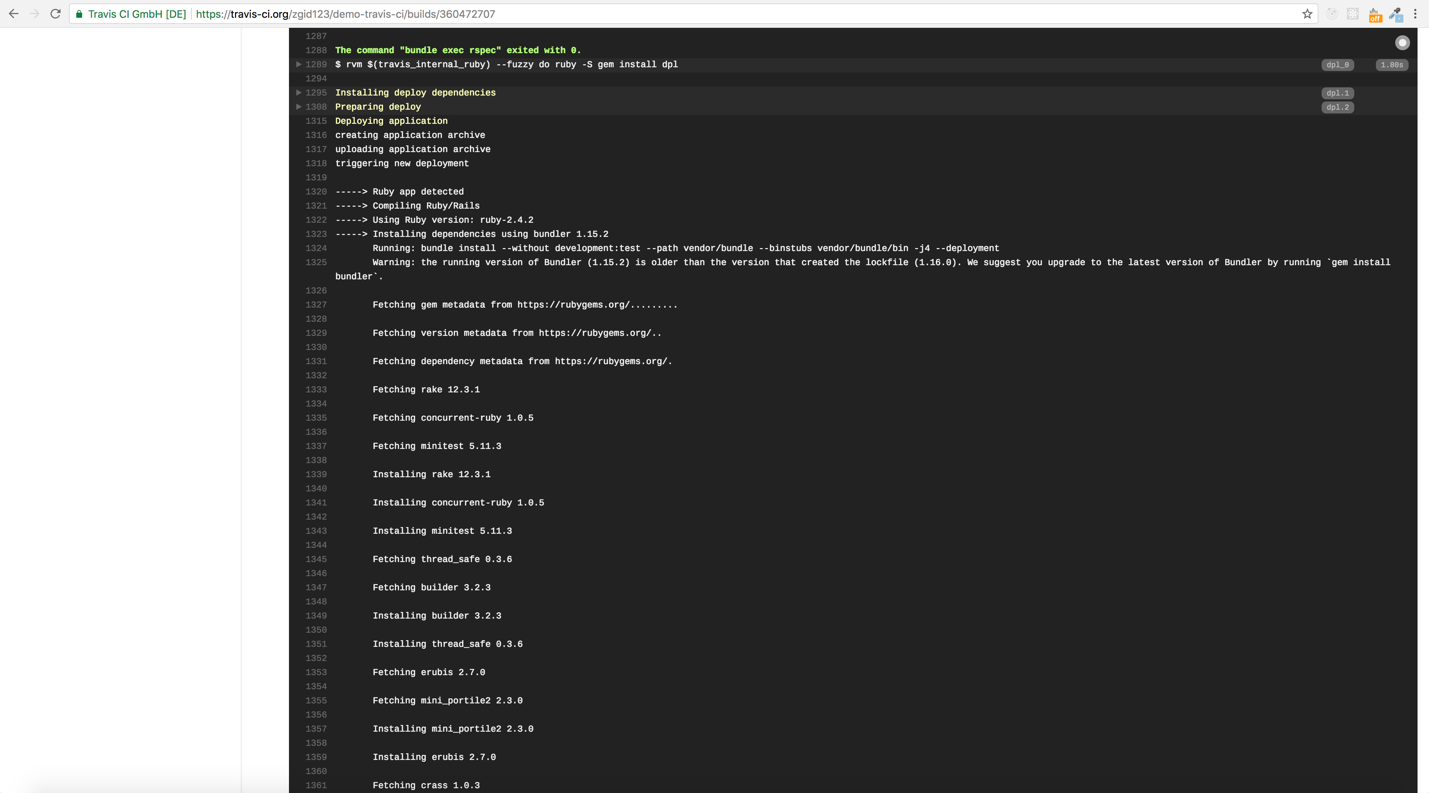
Chọn **New** rồi chọn **Create new app**.

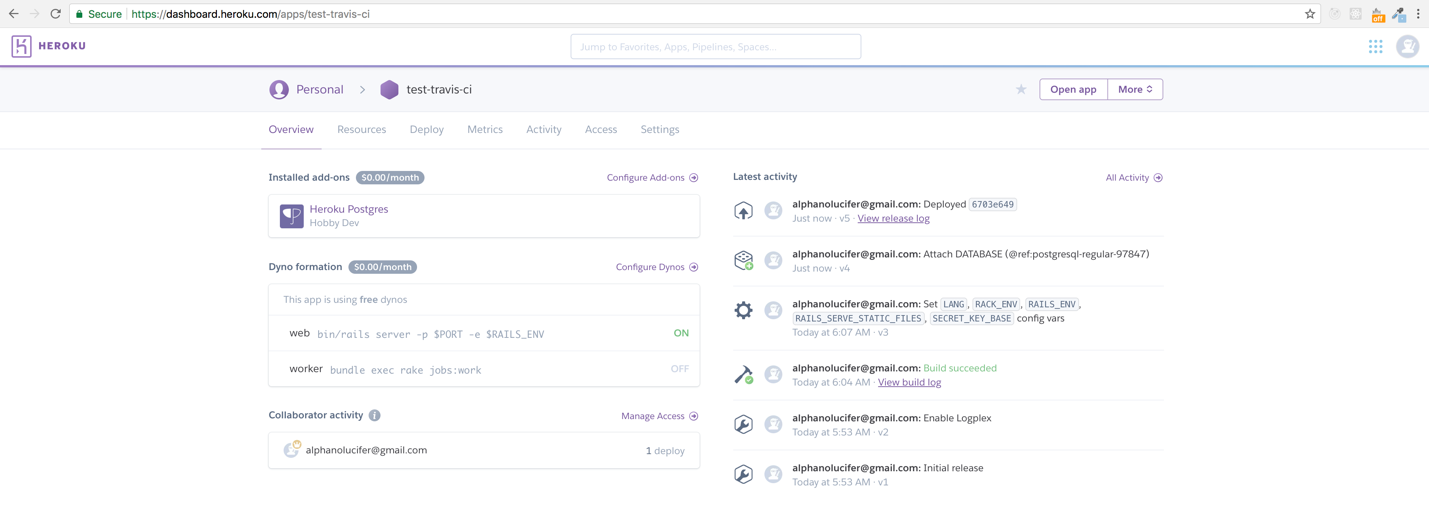
Nhập tên app rồi nhấn nút **Create app**.

Chạy lệnh **heroku git:remote -a <tên app>** trong terminal để git remote đến app vừa tạo.

* + - * Bước 2 – Thêm config vào file **.travis.yml** như sau:
        + Provider: Tên host sẽ deploy lên.
        + App: Tên app heroku.
        + Api\_key: lấy từ câu lệnh **heroku auth:token** từ terminal.
      * Bước 3 – Chạy lệnh **heroku labs:enable runtime-dyno-metadata -a <tên app>** để mở dynos meta data của heroku lên (nếu không khi build có thể bị lỗi).
      * Bước 4 – Push code mới để xem kết quả.

Trước khi deploy tự động:

**Travis CI** deploy tự động:

Sau khi deploy tự động:

Nguồn tham khảo:

https://docs.travis-ci.com/user/getting-started/