**TÀI LIỆU TRIỂN KHAI DỰ ÁN REACT LÊN VERCEL SỬ DỤNG GITHUB ACTIONS (CI/CD)**

1. **Giới thiệu**

Trong tài liệu này, tôi sẽ triển khai dự án **React** lên **Vercel** và tích hợp **CI/CD** thông qua **GitHub Actions**. Đây là công cụ tự động hóa của **GitHub**, giúp dự án được tự động build và deploy lên Vercel mỗi khi push code mới lên repository.

1. **Chuẩn bị và yêu cầu**

* Tài khoản GitHub.
* Đã có một repository trên GitHub chứa source code của dự án (có thể để public hoặc private)
* Tài khoản Vercel.
* Dự án đã chạy ổn định ở mô trường local.
* Các thông tin từ supabase để tạo Enviroment Variables trong Vercel
  + VITE\_SUPABASE\_KEY
  + VITE\_SUPABASE\_URL
* Các thông tin cần lấy từ Vercel để tạo secret trong GitHub:
  + VERCEL\_TOKEN
  + VERCEL\_ORG\_ID
  + VERCEL\_PROJECT\_ID

1. **Cấu hình trên Vercel**

Nếu chưa có tài khoản Vercel, hãy tạo một tài khoản Vercel theo các bước sau:

* Truy cập [**vercel.com**](https://www.vercel.com/) và nhấn Đăng ký.
* Có thể đăng ký bằng tài khoản GitHub để dễ dàng tích hợp.

A screenshot of a computer

AI-generated content may be incorrect.

Sau khi tạo tài khoản thành công hãy thực hiện theo những bước sau

**Bước 1: Kết nối repository trong Github chứa dự án cần triển khai với Vercel**

* Trong màn hình Dashboard của Vercel, nhấn vào **Add New** và chọn **Project**
* Từ danh sách repository, chọn **Import** dự án muốn triển khai.

A screenshot of a black screen

AI-generated content may be incorrect.

* Sau khi kết nối repository GitHub với Vercel, có thể chọn framework của dự án, thư mục gốc và cài đặt cấu hình của dự án đang triển khai.

A screenshot of a project

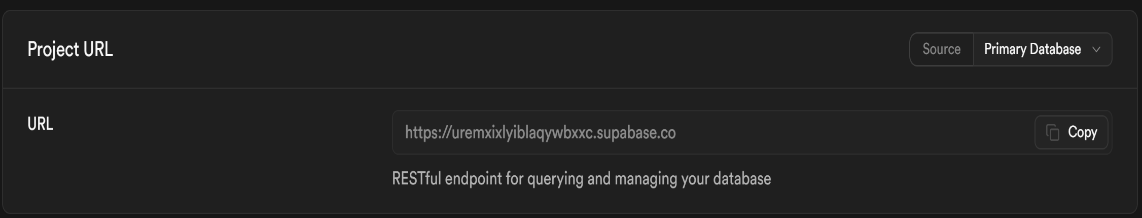
AI-generated content may be incorrect.

**Bước 2: Thiết lập biến môi trường (Enviroments Variable)**

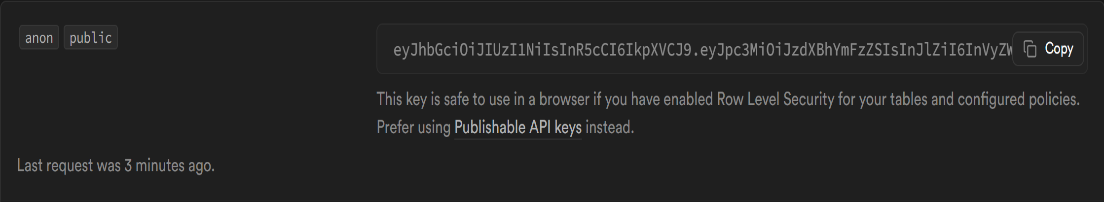
Trong phần cài đặt cấu hình của dự án, có thể thiết lập biến môi trường cho dự án cần triển khai. Ở dự án này dùng **Supabase** để làm Backend, vì vậy cần cấu hình Enviroment Variable để khai báo các thông tin kết nối.

Đầu tiên cần chuẩn bị thông tin từ Supabase để khai báo với Vercel là Project URL và Anon Public Key.

* Truy cập vào dự án ở tài khoản Supabase đang sử dụng cho dự án đang triển khai lên Vercel.
* Ở phần **Project Settings**, tìm đến mục **Data API** sẽ thấy thông tin của Project URL.



* Ngay bên dưới là mục **API Key** sẽ thấy thông tin của Anon Public Key.



Sau khi lấy thông tin từ Supabase, tiếp theo cấu hình biến môi trường cho dự án đang triển khai.

* Biến **VITE\_SUPABASE\_URL** có giá trị là **Project URL** từ Supabase.
* Biến **VITE\_SUPABASE\_KEY** có giá trị là **Anon Public Key** từ Supabase.

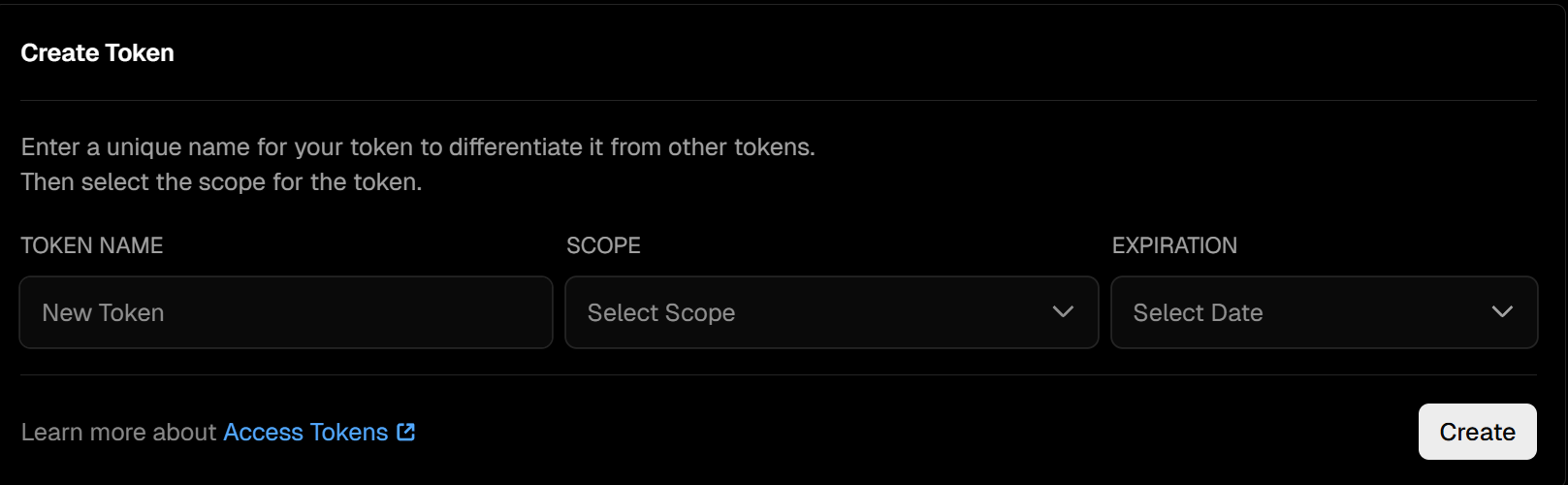
A screenshot of a computer

AI-generated content may be incorrect.

**Bước 3: Tạo Access Token**

Để GitHub Actions có thể triển khai dự án lên Vercel, cần có Access Token từ Vercel

* Ở màn hình Dashboard của Vercel, chọn **Account Setting**.
* Ở phần **Tokens**, có thể thiết lập Token Name, Scope và Expiration.



* Sau khi thiết lập các thông tin trên, nhấn **Create**. Vercel sẽ trả lại giá trị của Access Token.

A screenshot of a black screen

AI-generated content may be incorrect.

**Bước 4: Lấy VERCEL\_ORG\_ID và VERCEL\_PROJECT\_ID**

Ngoài Access Token, cần lấy **VERCEL\_ORG\_ID** và **VERCEL\_PROJECT\_ID** dùng để khai báo dự án cần triển khai với GitHub Action.

* Ở màn hình **Dashboard** của Vercel, chọn **Setting**.
* Ở phần General, tìm đến mục **Project ID** – đây là biến **VERCEL\_PROJECT\_*ID.***



* Ngay bên dưới cũng có mục **Team/Org ID** – đây là biến **VERCEL\_ORG\_ID**

A black screen with white text

AI-generated content may be incorrect.

1. **Cấu hình secret trong Github**

Sau khi đã có đủ Access Token, ORG\_ID và PROJECT\_ID, bước tiếp theo là thêm chúng vào GitHub Secrets. Đây là nơi lưu trữ an toàn các thông tin nhạy cảm, giúp GitHub Actions có thể truy cập và sử dụng trong quá trình build và triển khai mà không cần phải lộ trực tiếp trong mã nguồn.

Bước 1: Truy cập vào repository trong GitHub và nhấn vào **Setting**.

Bước 2: Tìm đến mục **Secrets and variable** ở phần sidebar và chọn **Action**.

Bước 3: Chọn **New repository secret** và tạo lần lượt 3 secret tương ứng với 3 giá trị lấy từ Vercel.

* Secret có tên **VERCEL\_TOKEN** có giá trị là Access Token.
* Secret có tên **VERCEL\_ORG\_ID** có giá trị là ORG\_ID.
* Secret có tên **VERCEL\_PROJECT\_ID** có giá trị là PROJECT\_ID.

Bước 4: Lưu lại 3 secret đã tạo



1. **Cấu hình Github Actions**

Sau khi đã cài đặt các Secrets, bước tiếp theo là cấu hình GitHub Actions để tự động build và triển khai dự án lên Vercel. Trong bước này, chúng ta sẽ tạo một workflow để mô tả các công việc cần chạy.

**Bước 1: Tạo file workflow**

Ở repository dự án đang triển khai, chọn phần **Actions**. Tại đây nhấn **New workflow**. Sau đó sẽ xuất hiện màn hình chọn workflow, nhấn **set up a workflow yourself**.

Tại đây, có thể đặt tên và tạo nội dung cho file workflow.

**Bước 2: Nội dung của file**

A screen shot of a computer

AI-generated content may be incorrect.

Giải thích về workflow:

* **name: Deploy to Vercel**

Đặt tên workflow là **CI/CD with Docker + Vercel**. Tên này sẽ hiển thị trong tab **Actions** của GitHub.

* **on: push: branches: main**

+ Định nghĩa cho sự kiện kích hoạt flow.

+ Workflow sẽ kích hoạt khi có push lên nhánh main.

* **jobs: deploy**

Tạo một job có đặt tên là deploy.

* **runs-on: ubuntu-latest**

Job sẽ chạy trên máy ảo Ubuntu mới nhất do GitHub cung cấp.

* **steps:** các bước thực thi

**- name: Checkout repository**

**uses: actions/checkout@v3**

* + - Lấy toàn bộ mã nguồn từ repository về máy ảo.

**- name: Setup Node.js**

**uses: actions/setup-node@v3**

**with:**

**node-version: 18**

* + - Cài đặt môi trường Node.js phiên bản **18.**

**- name: Install dependencies**

**run: npm install**

* + - Chạy ***npm install*** để tải các gói phụ thuộc cần thiết cho project.

**- name: Build project**

**run: npm run build**

* + - Chạy lệnh build (theo script trong package.json) → tạo ra bản build production của dự án.

**- name: Deploy to Vercel**

**uses: amondnet/vercel-action@v25**

**with:**

**vercel-token: ${{ secrets.VERCEL\_TOKEN }}**

**vercel-org-id: ${{ secrets.VERCEL\_ORG\_ID }}**

**vercel-project-id: ${{ secrets.VERCEL\_PROJECT\_ID }}**

**working-directory: ./**

**scope: ${{ secrets.VERCEL\_ORG\_ID }}**

* + - Dùng ***action amondnet/vercel-action*** để deploy lên Vercel.
    - Các thông tin vercel-token, vercel-org-id, vercel-project-id được lấy từ **GitHub Secrets** đã cấu hình trước đó.
    - ***working-directory: ./*** chỉ định thư mục gốc của project.
    - ***scope*** giúp xác định phạm vi (tổ chức / team) khi deploy.

**Bước 3: Push file workflow lên GitHub repository**

Sau khi tạo nội dung cho file workflow, nhấn **Commit changes** để push file workflow lên GitHub repository

A screenshot of a computer screen

AI-generated content may be incorrect.

Sau khi push file workflow lên repository, GitHub Actions sẽ tự động kích hoạt triển khai dựa trên quy trình đã thiết lập.

1. **Kiểm tra kết quả**
2. **Deployment**

Sau khi thực hiện tất cả các bước trên, truy cập vào dự án đang triển khai trên Vercel. Nhấn vào phần **Deployments**, Vercel sẽ tự động triển khai dự án.

1. **CI/CD**

Để đảm bảo quá trình CI/CD hoạt động đúng như mong đợi, cần tiến hành kiểm tra lại toàn bộ pipeline giúp xác nhận mỗi khi có thay đổi được đẩy lên GitHub, workflow sẽ tự động build và triển khai dự án lên Vercel.

**Bước 1: Push code lên GitHub**

Thực hiện commit và push một thay đổi bất kỳ để kích hoạt workflow.

**Bước 2: Theo dõi trạng thái workflow**

Truy cập vào tab **Actions** trong repository để kiểm tra pipeline có được chạy thành công hay không.

**Bước 3: Kiểm tra log build**

Mở chi tiết workflow để xem log build nhằm đảm bảo không có lỗi xảy ra trong quá trình cài đặt, build và deploy.

**Bước 4: Xác minh trên Vercel**

Truy cập trang quản lý dự án trong Vercel để đảm bảo deployment mới đã được cập nhật.

**Bước 5: Truy cập ứng dụng thực tế**

Mở URL của ứng dụng để kiểm tra giao diện và chức năng có hoạt động đúng như những thay đổi mới nhất.

1. **Kết luận**

Qua toàn bộ quá trình trên, dự án đã được thiết lập CI/CD thành công với GitHub Actions và triển khai tự động lên Vercel. Việc cấu hình này giúp rút ngắn thời gian triển khai và đảm bảo thay đổi trong mã nguồn đều được kiểm tra.