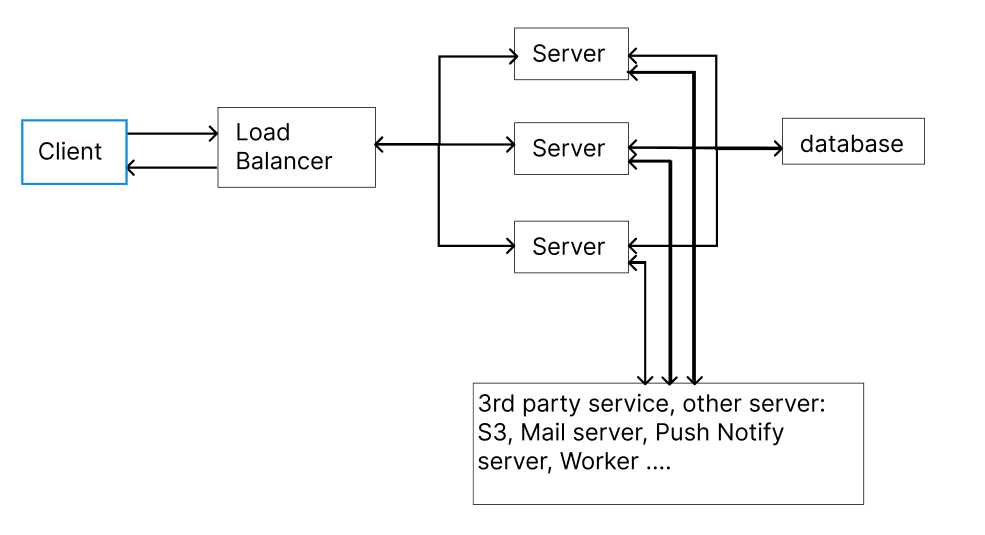
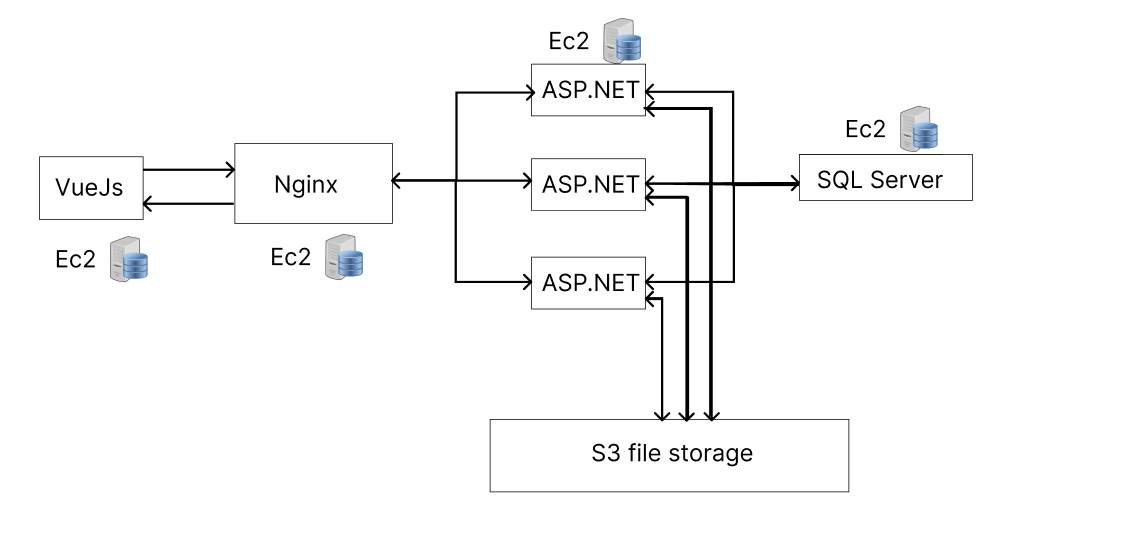
BÁO CÁO HANDON TRIỂN KHAI HỆ THỐNG + LAB CI/CD

1. Kiến trúc ứng dụng triển khai
2. Tổng quan kiến trúc



1. Công nghệ triển khai trên các thành phần tương ứng



- Trên thực tế sẽ triển khai 4 máy: EC2

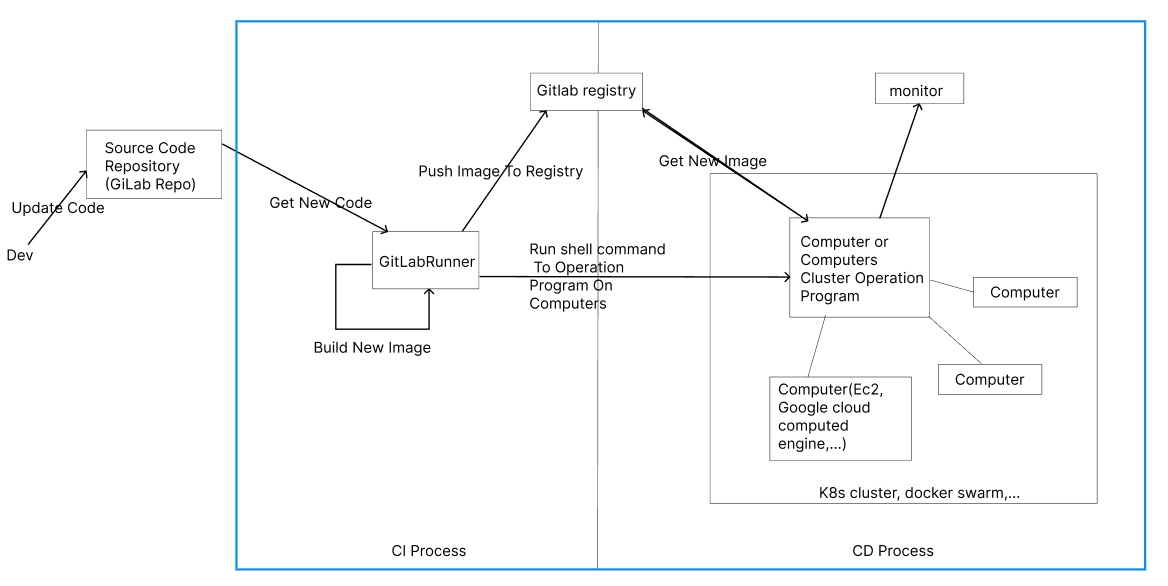
+ 1 máy chạy ứng dụng: VueJs

+ 1 máy chạy server: ASP.NET

+ 1 máy chạy Load balancer-nginx

+ 1 máy chạy Database: SQL Server

1. Mô hình, quy trình triển khai CI/CD



- Mô tả về quy trình triển khai, cùng các thành phần liên quan:

1. Sau khi dev phát triển xong các tính năng mới sẽ đẩy code lên gitlab repository.
2. Gitlab kích hoạt chạy các jobs được định nghĩa trong file .gitlab-ci.yml.
3. Các job được thực hiện tại thành phần gitlab runner. Gitlab runner là một máy tính có cài các phần mềm thực hiện(executor) như là shell, docker, virtual box….(ở đây chọn docker là executor và docker dind là image thực hiện các shell command trong các job) Một phiên thực hiện thường bao gồm 2 job: build, deploy.
4. Build job được thực hiện. Gitlab đã tự động lấy code mới nhất vào trong container thực hiện. Trong đó đã có mã nguồn chương trình và file Dockerfile - file chứa các lệnh tạo image tự động. Dựa vào source code và dockerfile, container thực hiện build ra image và đẩy lên gitlab registry.
5. Thực hiện tiếp Deploy job. Thực hiện ssh vào các máy triển khai chương trình, clone file docker-compose.yml và chạy file để tạo ra các container có image mới nhất vửa được đẩy lên registry ở bước 4.
6. Triển khai hệ thống

Thực hiện miêu tả các bước, các tài nguyên cần chuẩn bị để triển khai hệ thống, ci/cd.

1. Xác định những thành phần chính cần triển khai.

- 1 máy deploy frontend.

- 1 máy deploy backend

- 1 máy cài đặt nginx trỏ đến backend

- 1 máy cài đặt database

- 1 máy cài đặt gitlab-runner

1. Tài nguyên cần chuẩn bị:

- 5 máy Ec2 chạy ubuntu 20.04 trong đó có 1 máy EC2 ram min 2G để chạy database.

- Cài đặt các phần mềm cần thiết vào các máy:

+ Cài đặt database - 1 máy: mô tả trong file cùng thư mục

+ Cài đặt gitlab runner - 1 máy: mô tả trong file cùng thư mục

+ Cài đặt nginx - 1 máy: mô tả trong file cùng thư mục

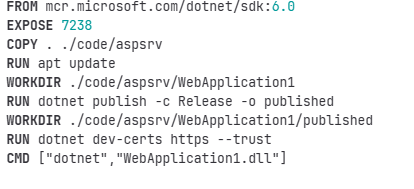
+ Cài đặt docker, docker-compose: 2 máy: Mô tả cùng thư mục.

1. Triển khai ci/cd

Miêu tả triển khai ci/cd phần backend. Frontend triển khai tương tự chỉ khác về các command build image.

- Chuẩn bị source code kèm các file để push lên repository gitlab:

+ Dockerfile - file tạo image backend



+ docker-compose.file - file khởi chạy các container có image được build từ Dockerfile



+ deploy.sh - file command shell điều khiển các máy ec2 chạy ứng dụng.



- Chuẩn bị các biến được sử dụng khi thực hiện các job:

+ Tại web repository vào tab Setting -> CICD -> tab Variable -> add Variable -> cấu hình các biến. Các biến gồm tên biến và giá trị.

+ Các biến cần thêm: SSH\_USER - user trên máy ec2 chạy chương trình(ubuntu), SSH\_PRIVATE\_KEY - private key để ssh, SSH\_IP\_SERVER\_TEST- ip máy ec2 chạy chương trình

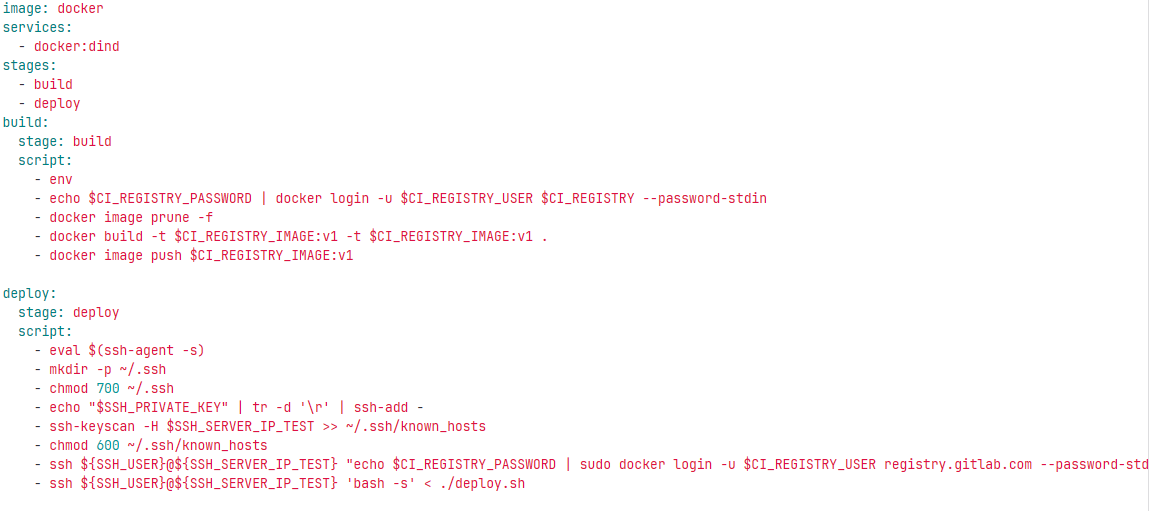
+ Lấy giá trị key biến SSH\_PRIVATE\_KEY để ssh từ container docker dind vào các máy chạy chương trình:

* Copy file key.pem của bạn lên máy chạy chương trình:
  + scp -i key.pem key.pem [ubuntu@ip-ec2:/home/ubuntu/.ssh](mailto:ubuntu@ip-ec2:/home/ubuntu/.ssh)
* Ssh lên máy chạy chương trình, truy cập vào folder chứa file key.pem mới copy, xuất ra private key.
  + ssh command ..
  + cd $home/ubuntu/.ssh
  + open ssl rsa -in key.pem -out id\_rsa
  + cat id\_rsa

- Đẩy code lên repository.

- Cấu hình pipeline - tức thêm file .gitlab-ci.yml:

+ Tại web gitlab repository, vào tab CI/CD -> chọn pipeline -> chọn một template mẫu bất kỳ -> sửa nội dung edit như ảnh bên dưới -> change commit changes



- Xem kết quả thực hiện pipeline tại tab cicd -> job và thực hiện chỉnh sửa dựa vào các logs lỗi.