

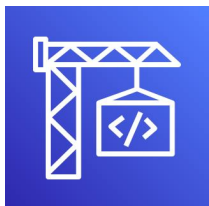


Devops on *AWS* for beginner

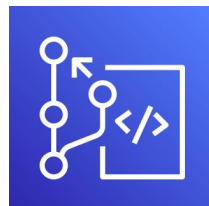
Instructor – Linh Nguyen

Engineering Consultant, AWS Cloud Solution Architect

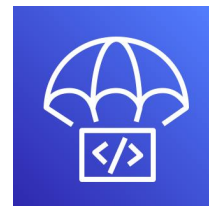
CodePipeline



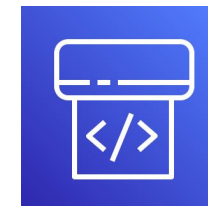
CodeBuild



CodeCommit



CodeDeploy



CodePipeline

“Không có việc gì khó, chỉ sợ không biết làm!” 😊

Target

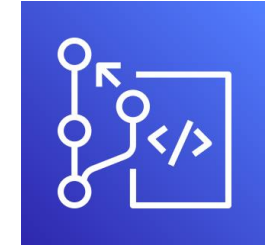
- Giới thiệu về CodeCommit , CodeBuild, CodeDeploy, CodePipeline.
- Sử dụng CodeCommit để lưu trữ và quản lý source code.
- Sử dụng CodeBuild để build một số project vd Java, Nodejs, Docker.
- Sử dụng CodeDeploy để deploy lên ElasticBeanstalk và ECS.
- Kết hợp tất cả trong CodePipeline để tạo thành flow CI/CD hoàn chỉnh.
- Tìm hiểu về chiến lược deployment Blue-Green và Canary sử dụng CodeDeploy.

Giới thiệu CodeCommit

CodeCommit là một dịch vụ được quản lý bởi AWS, giúp bạn lưu trữ và quản lý source code, cấu hình và các tài liệu liên quan trong một kho lưu trữ Git an toàn, riêng tư và có thể mở rộng.

Một số tính năng chính của AWS CodeCommit bao gồm:

- Quản lý mã nguồn: CodeCommit cung cấp một kho lưu trữ Git, cho phép bạn lưu trữ và quản lý mã nguồn, cấu hình và tài liệu.
- Bảo mật: CodeCommit mã hóa các tệp và metadata của bạn tại chỗ và trong quá trình transfer.
- Tích hợp với các dịch vụ AWS khác: CodeCommit có thể tích hợp với các dịch vụ AWS khác như **CodeBuild**, **CodeDeploy** và **CodePipeline** để tự động hóa quy trình phát triển phần mềm.
- Có thể mở rộng: CodeCommit có thể mở rộng để hỗ trợ dự án từ nhỏ đến lớn, từ một vài người dùng đến hàng trăm người dùng.
- Quản lý truy cập: Tích hợp trực tiếp với IAM để quản lý truy cập, setting các policy.



AWS CodeCommit

*Lưu ý: CodeCommit đã ngưng hỗ trợ khách hàng mới từ 25/7/2024. Các tài khoản AWS tạo trước đó vẫn có thể sử dụng tiếp tới khi ngưng hỗ trợ.

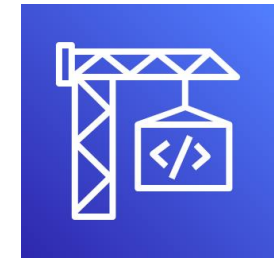
Giới thiệu CodeBuild

CodeBuild là một dịch vụ xây dựng và kiểm thử từ AWS giúp bạn biên dịch mã nguồn, chạy kiểm thử và tạo ra các **software package** mà có thể được triển khai.

Dịch vụ này hoạt động dựa trên cấu hình mà bạn cung cấp dưới dạng file '**buildspec**' hoặc thông qua một **Dockerfile** từ source code.

Một số tính năng chính của AWS CodeBuild bao gồm:

- Tự động hóa
- Có thể mở rộng: CodeBuild có thể mở rộng để xử lý các công việc xây dựng lớn và đồng thời xử lý nhiều công việc xây dựng cùng một lúc.
- Tích hợp với các dịch vụ AWS khác như **CodeCommit**, **CodeDeploy** và **CodePipeline** để tự động hóa quy trình phát triển phần mềm.
- Bảo mật: CodeBuild chạy trong một VPC riêng, giúp bảo vệ môi trường build của bạn.
- Hiệu quả chi phí: Với CodeBuild, bạn chỉ phải trả cho thời gian build thực tế mà bạn sử dụng.



AWS CodeBuild

Giới thiệu CodeDeploy

CodeDeploy là một dịch vụ triển khai tự động từ AWS giúp bạn tự động hóa việc triển khai ứng dụng vào các môi trường dịch vụ như EC2, ECS, AWS Lambda, và cả máy chủ vật lý của bạn.

Một số tính năng chính của CodeDeploy bao gồm:

- Triển khai tự động: giúp giảm thời gian downtime và tăng tốc độ triển khai.
- Triển khai môi trường phức tạp: CodeDeploy hỗ trợ triển khai vào các môi trường dịch vụ phức tạp và cung cấp khả năng quản lý cấu hình, điều khiển phiên bản và quản lý trạng thái ứng dụng.
- Tích hợp với các dịch vụ AWS khác: như CodeCommit, CodeBuild và AWS CodePipeline để tự động hóa quy trình phát triển phần mềm.
- Quản lý chi phí: Với CodeDeploy, bạn chỉ phải trả cho những gì bạn sử dụng. Không cần trả trước và bạn có thể bắt đầu sử dụng CodeDeploy miễn phí.
- Cập nhật liên tục: CodeDeploy hỗ trợ cập nhật liên tục, giúp bạn triển khai các phiên bản mới của ứng dụng một cách nhanh chóng và dễ dàng.



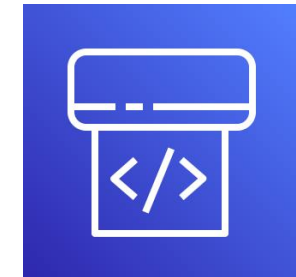
AWS CodeDeploy

Giới thiệu CodePipeline

CodePipeline là một dịch vụ triển khai tự động từ Amazon Web Services giúp bạn tự động hóa các bước trong quy trình phát triển phần mềm. Nó cho phép bạn xác định một chuỗi các bước, gọi là "pipeline", mà mỗi bước sẽ được thực hiện mỗi khi có thay đổi code. Các bước này có thể bao gồm việc build code, kiểm tra và triển khai ứng dụng.

Ví dụ, bạn có thể thiết lập một pipeline với các bước sau:

- Khi có thay đổi code trong kho lưu trữ Git, AWS CodePipeline sẽ tự động chạy.
- Code sẽ được build bằng cách sử dụng một công cụ build như AWS CodeBuild.
- Sau khi build thành công, các kiểm tra tự động sẽ được chạy để đảm bảo chất lượng code.
- Nếu tất cả các kiểm tra đều thành công, code sẽ được triển khai lên một môi trường như Elastic Beanstalk hoặc Fargate (ECS).
- Với CodePipeline, bạn có thể tự động hóa quy trình phát triển phần mềm, giúp giảm thời gian triển khai và tăng cường chất lượng sản phẩm.



AWS CodePipeline

Lab1 – Sử dụng CodeCommit làm repository quản lý source code

Yêu cầu:

- Tạo một repository trên CodeCommit
- Tạo một credential cho code commit (dạng username/password hoặc SSH).
**Xem lại bài lab CodeCommit trong chương thao tác cơ bản với Git.*
- Checkout source code, modify, push.

**Lưu ý: CodeCommit đã ngưng hỗ trợ khách hàng mới từ 25/7/2024. Các tài khoản AWS tạo trước đó vẫn có thể sử dụng tiếp tới khi ngưng hỗ trợ.*

**Các bạn sở hữu account không thể tạo CodeCommit repo có thể skip qua bài lab này (hoặc chỉ xem để tham khảo). Mình có record lại các bài lab với prefix [NEW] cho các bài lab từ Lab 2 - Lab 8 sử dụng Github thay cho CodeCommit.*

Lab2 - Tạo một Job CodeBuild đơn giản (Java project)

Yêu cầu:

- Tạo một Job trong CodeBuild có nhiệm vụ checkout một Java project (từ Github/CodeCommit), build sau đó push lên S3.
- Chạy job build
- Kiểm tra artifact được push lên S3.

Lab3 - Tạo một Job CodeBuild đơn giản (Nodejs project)

Yêu cầu:

- Tạo một Job trong CodeBuild có nhiệm vụ checkout một Nodejs project (từ Github/CodeCommit).
- Install dependencies sử dụng lệnh: **npm install**
- Build static file sử dụng lệnh: **npm run build**
- Push artifact lên S3 (file tĩnh) sử dụng câu lệnh **aws s3 cp**

Lab4 - Tạo một Job CodeBuild đơn giản (Docker)

Yêu cầu:

- Tạo một ECR Repository
- Tạo một Job trong CodeBuild có nhiệm vụ checkout một Nodejs project (từ Github/CodeCommit).
- Build thành Docker image
- Push Docker image lên ECR Repository.

Lab5 - Tạo một CodePipeline đơn giản – Deploy lên Elastic Beanstalk

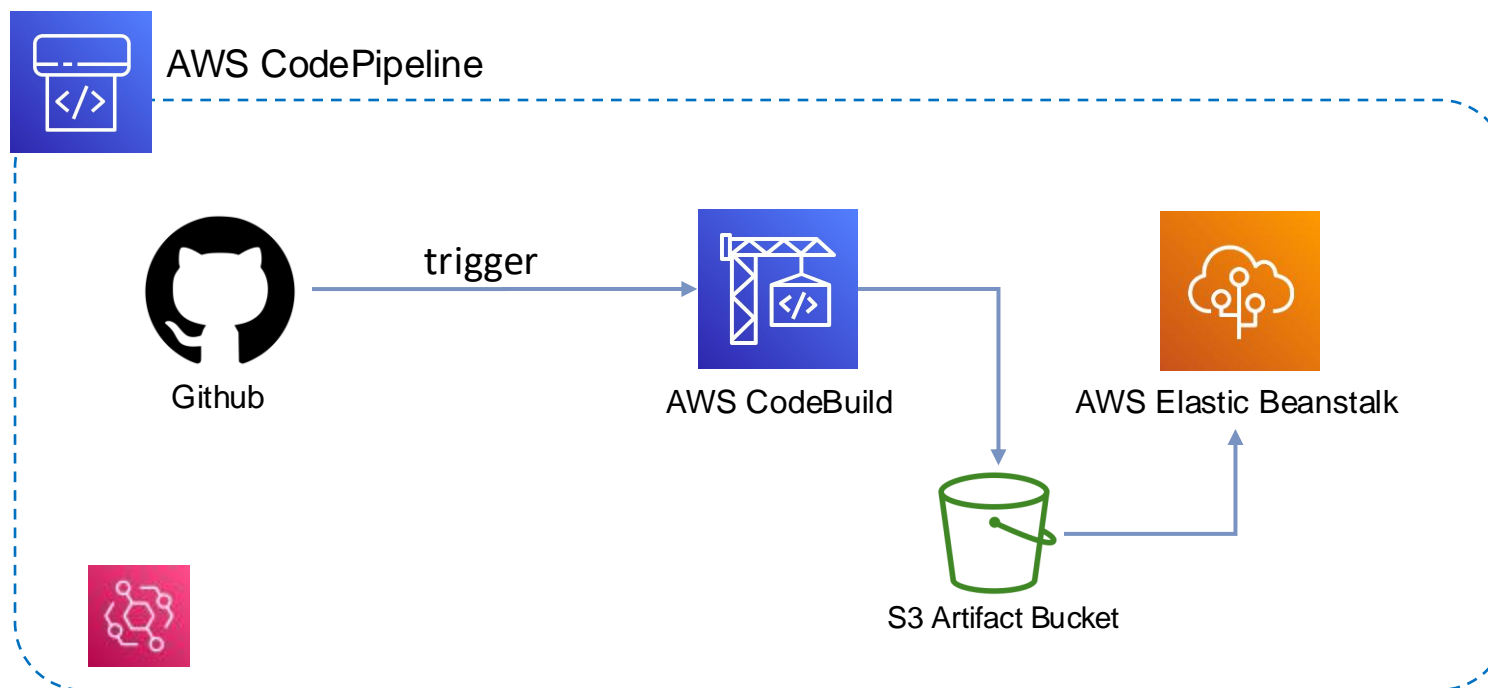
Yêu cầu:

**Sử dụng lại output của bài lab CodeBuild – Java project.*

- Tạo một Elastic Beanstalk Application+environment chạy Java, sử dụng file .jar của bài lab 2. Kiểm tra application chạy thành công.
- Tạo một pipeline trong CodePipeline có nhiệm vụ trigger job CodeBuild sau đó deploy artifact lên Elastic Beanstalk theo hình thức All-At-One.
- Modify code, push, kiểm tra pipeline được trigger tự động.
- Kiểm tra ElasticBeanstalk environment được triển khai version mới.
- Truy cập thử.

Lab5 - Tạo một CodePipeline đơn giản – Deploy lên Elastic Beanstalk

Sơ đồ hệ thống của bài lab



*Lưu ý: CodeCommit đã ngưng hỗ trợ khách hàng mới từ 25/7/2024. Các tài khoản AWS tạo trước đó vẫn có thể sử dụng tiếp tới khi ngưng hỗ trợ.

Lab5 - Tạo một CodePipeline đơn giản – Deploy lên Elastic Beanstalk

Clear resources:

- Xoá ElasticBeanstalk Application + Environment.
- Xoá bớt file lớn trên S3 bucket.
- CodeBuild, CodePipeline: Giữ lại để tham khảo & sử dụng cho các bài lab sau.

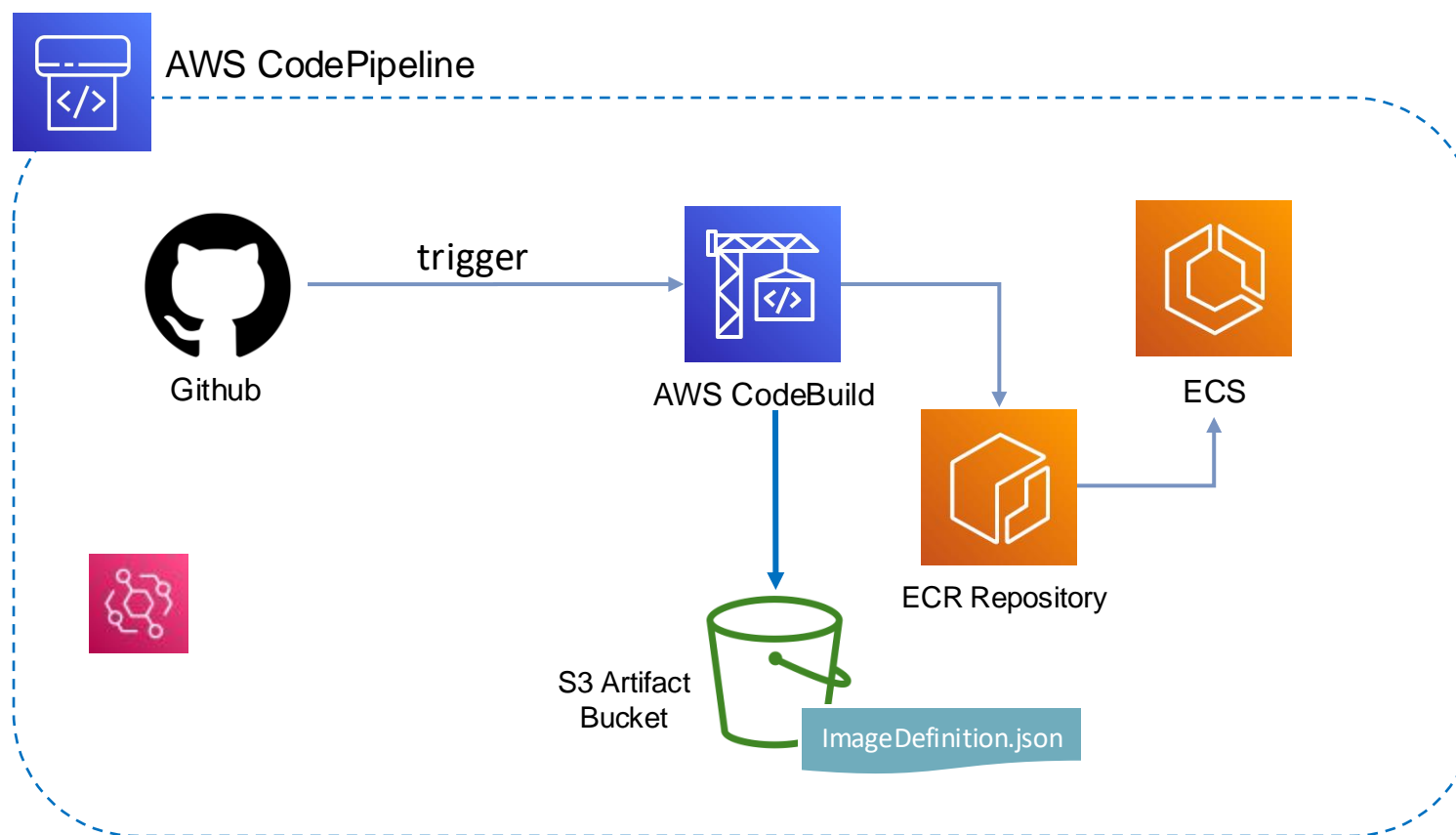
Lab6 - Tạo một CodePipeline– Deploy lên ECS sử dụng phương án Rollout

**Sử dụng lại output của bài lab 4 (docker image trên ECR repository)*

- Triển khai một cluster ECS sử dụng Terraform cho sẵn. Truy cập thử.
- Modify job CodeBuild của bài lab4, thêm bước tạo file **Json definition**.
- Tạo một pipeline trong CodePipeline có nhiệm vụ trigger job CodeBuild build ra ECR image sau đó deploy Docker lên ECS theo phương án rollout.
- Modify code, push, kiểm tra pipeline được trigger tự động.
- Kiểm tra ECS Service được triển khai version mới.
- Kiểm tra và truy cập thông qua ALB => new version.

Lab6 - Tạo một CodePipeline– Deploy lên ECS sử dụng Rollout

Sơ đồ hệ thống của bài lab



Lab6 - Tạo một CodePipeline– Deploy lên ECS sử dụng Rollout

Clear resource:

- Sử dụng Terraform destroy để xoá stack.
- Xoá bớt image trên ECR repository.
- CodeBuild, CodePipeline: Giữ lại để tham khảo & sử dụng cho bài lab sau.

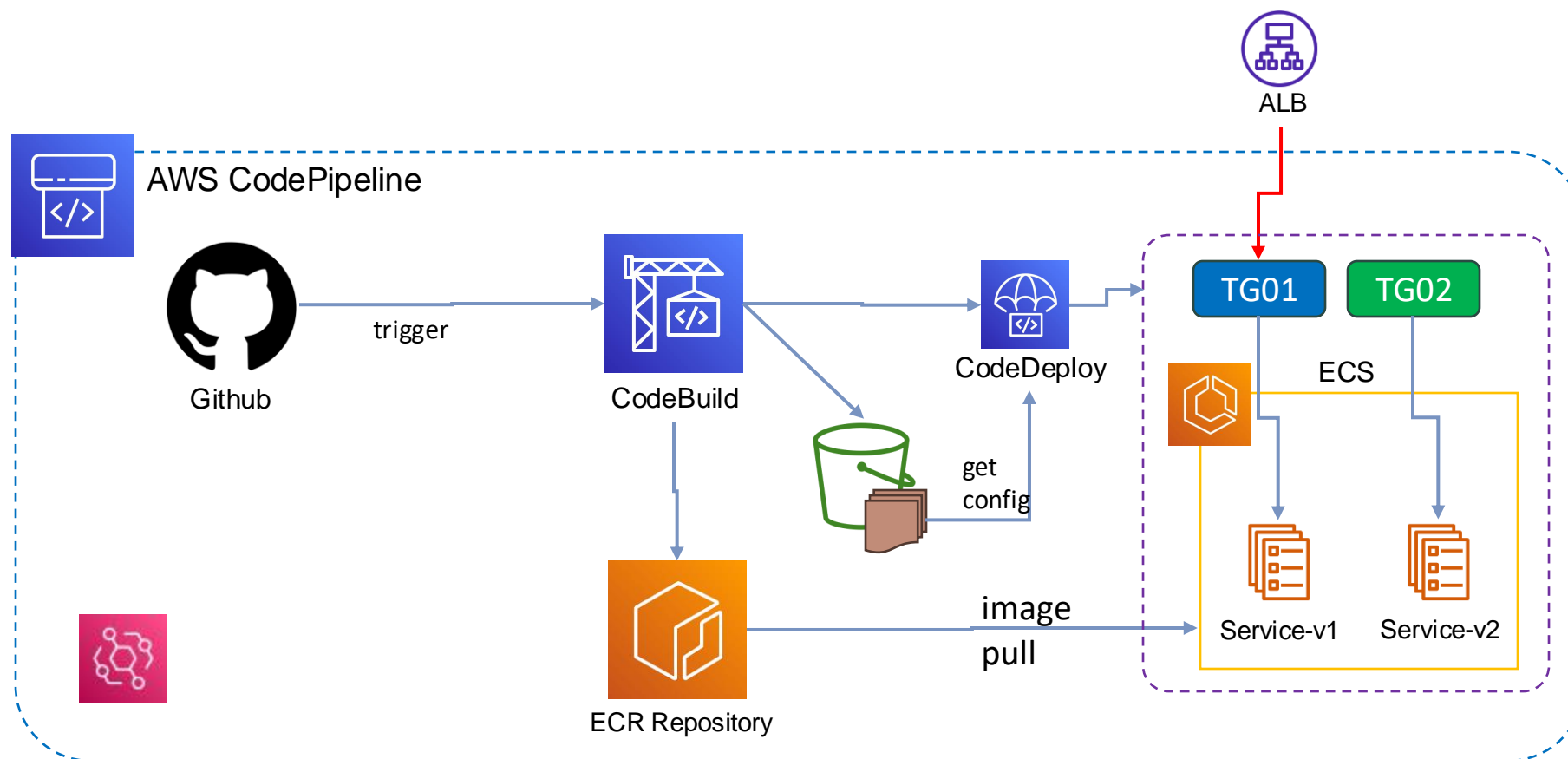
Lab7 - Tạo một CodePipeline – Deploy lên ECS sử dụng Blue-Green

Yêu cầu:

- Modify ECS Service (terraform) chuyển deployment controller thành **CODE_DEPLOY**.
- Thêm 1 target group và 1 listener trên ALB (port 81).
- Tạo một Pipeline trong đó step CodeDeploy sử dụng chiến lược Blue-Green
- Update code, push lên CodeCommit.
- Theo dõi pipeline
- Chờ đến thời gian traffic được switch sang version mới.
- Terminate version cũ.

Lab7 - Tạo một CodePipeline – Deploy lên ECS sử dụng Blue-Green

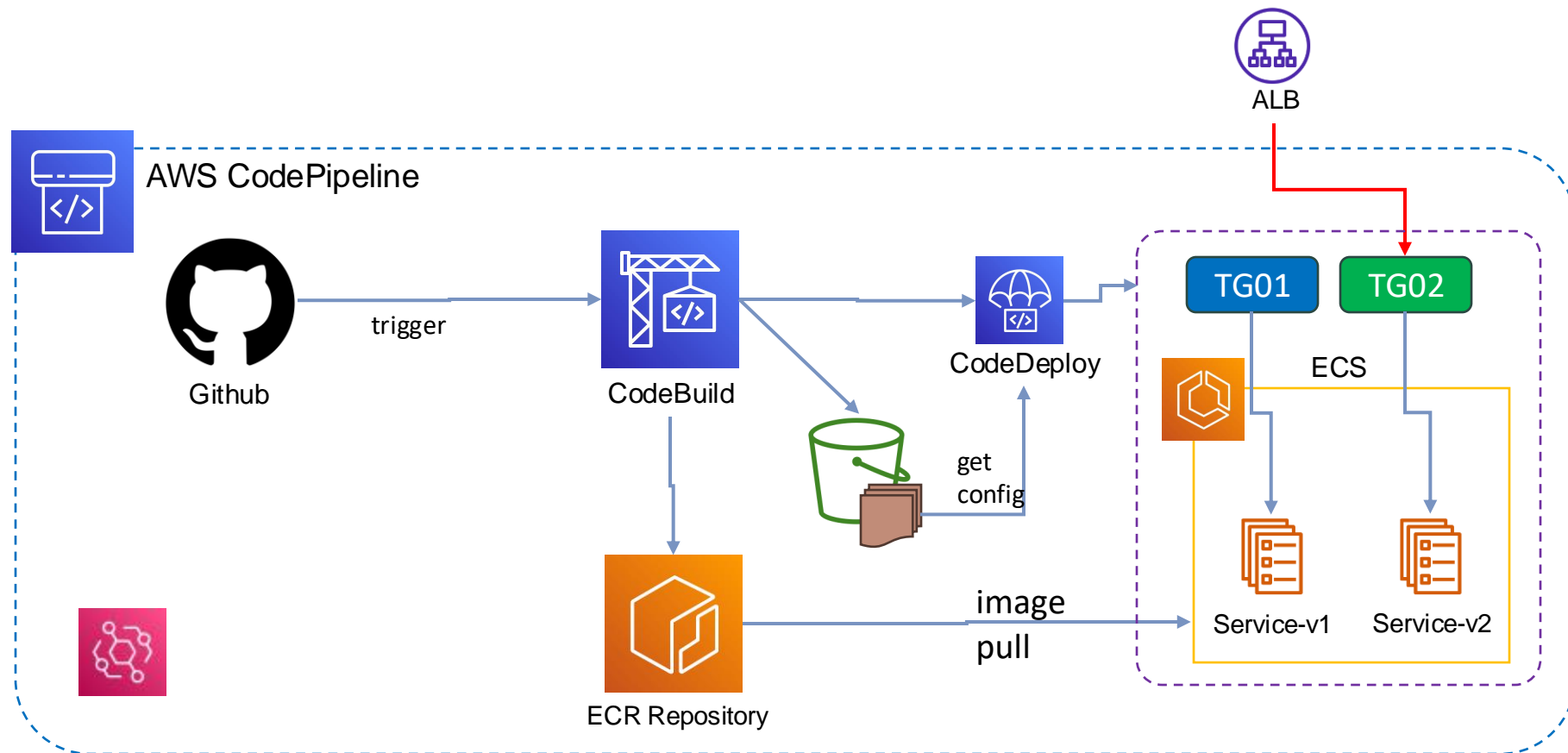
Sơ đồ hệ thống của bài lab



*Lưu ý: Blue-Green trong trường hợp này là service level (không phải cluster level)

Lab7 - Tạo một CodePipeline – Deploy lên ECS sử dụng Blue-Green

Sơ đồ hệ thống của bài lab



*Lưu ý: Blue-Green trong trường hợp này là service level (không phải cluster level)

Lab8 - Tạo một CodePipeline – Deploy lên ECS sử dụng Canary/Linear

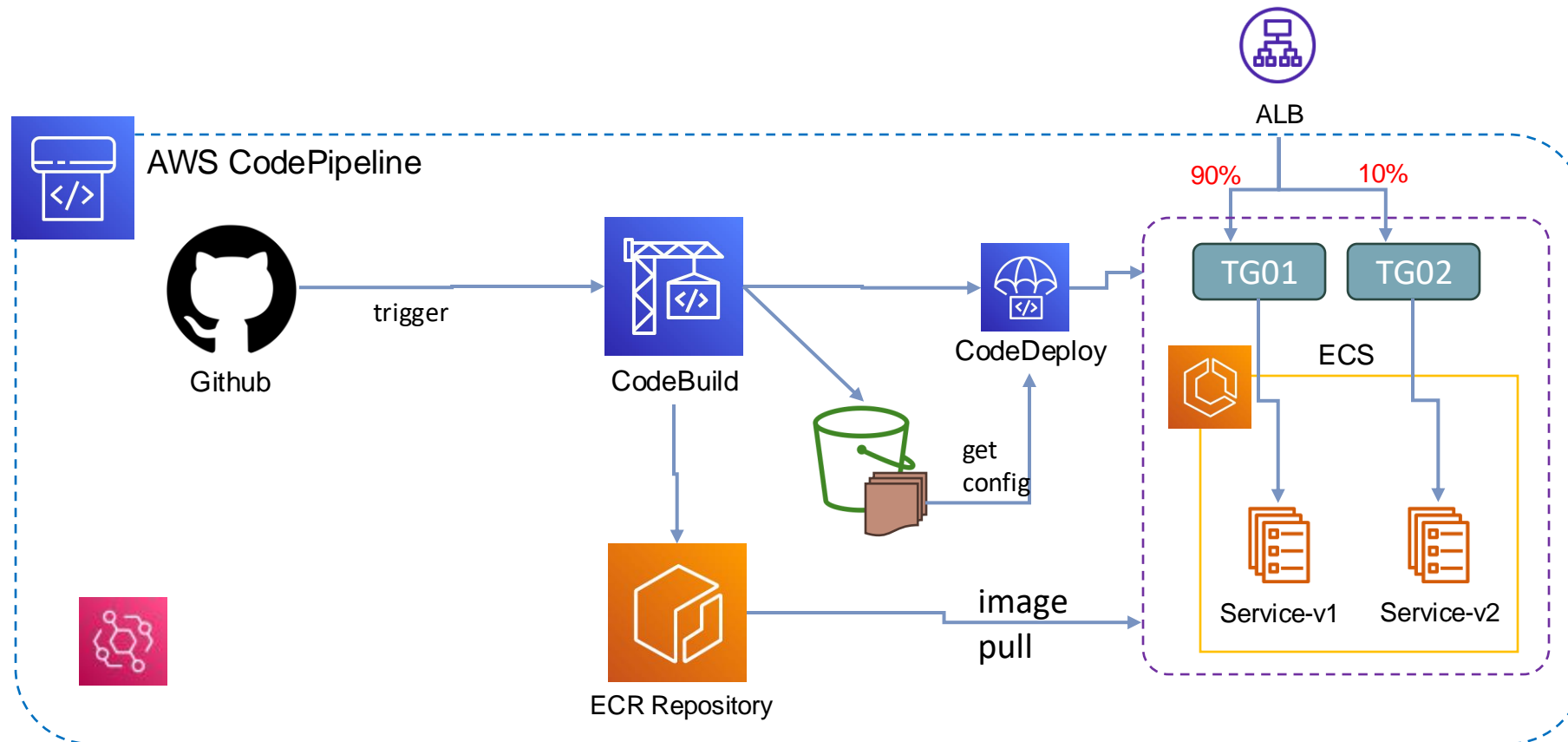
Yêu cầu:

Sử dụng lại resource của bài lab 6

- Modify job deploy, chuyển thành chiến lược deploy theo kiểu canary.
- Cấu hình chiến lược deploy theo kiểu canary
 - Start: 25% nodes (lưu ý chỉnh số tasks của service lên 4).
 - Waiting time: 5 minutes.
- Kiểm tra và truy cập thông qua ALB -> 25% request nhận version mới.
- Đợi hết thời gian waiting -> truy cập ALB -> 100% traffic chuyển sang new version.

Lab8 - Tạo một CodePipeline – Deploy lên ECS sử dụng Canary

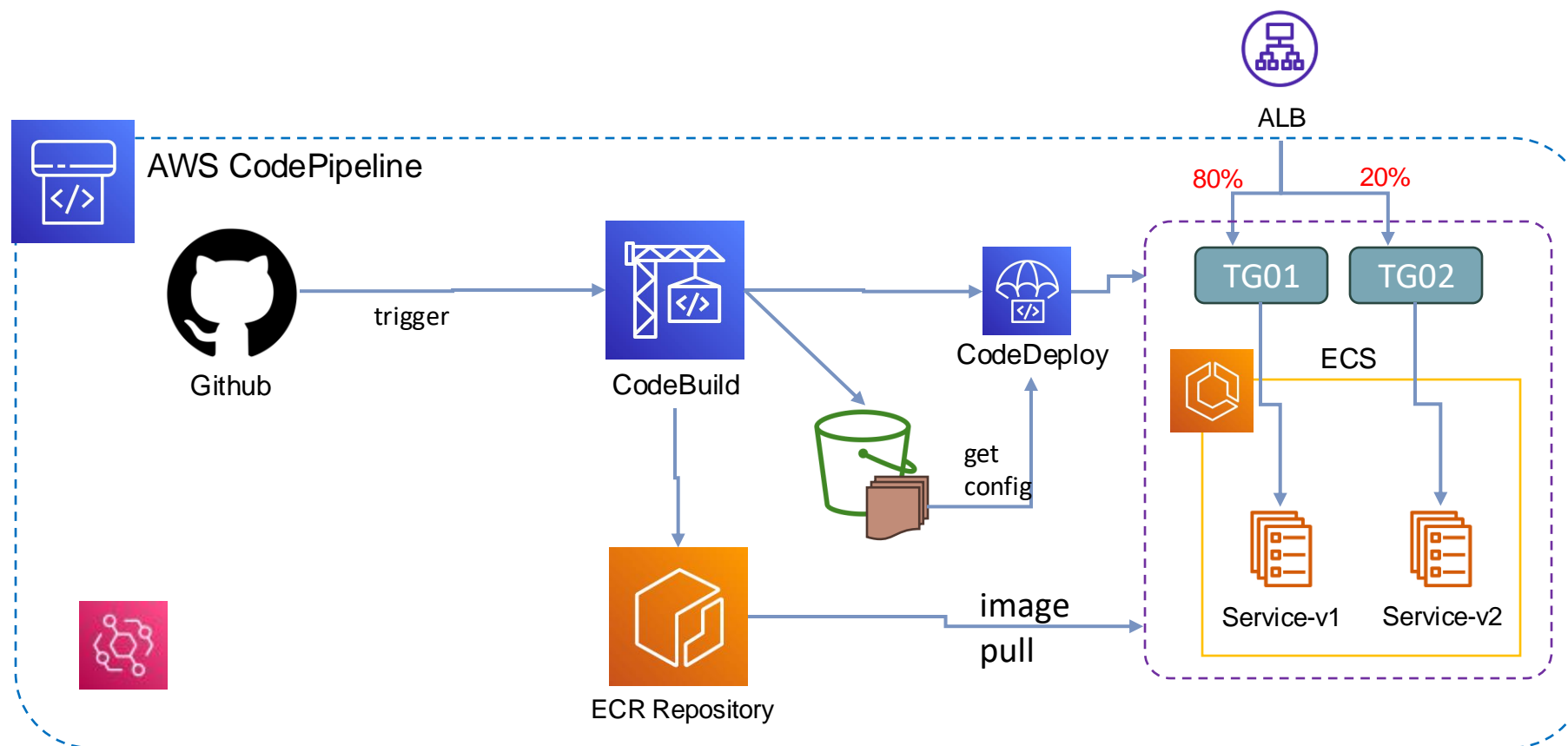
Sơ đồ hệ thống của bài lab



*Lưu ý: Canary trong trường hợp này là service level (không phải cluster level)

Lab8 - Tạo một CodePipeline – Deploy lên ECS sử dụng Canary

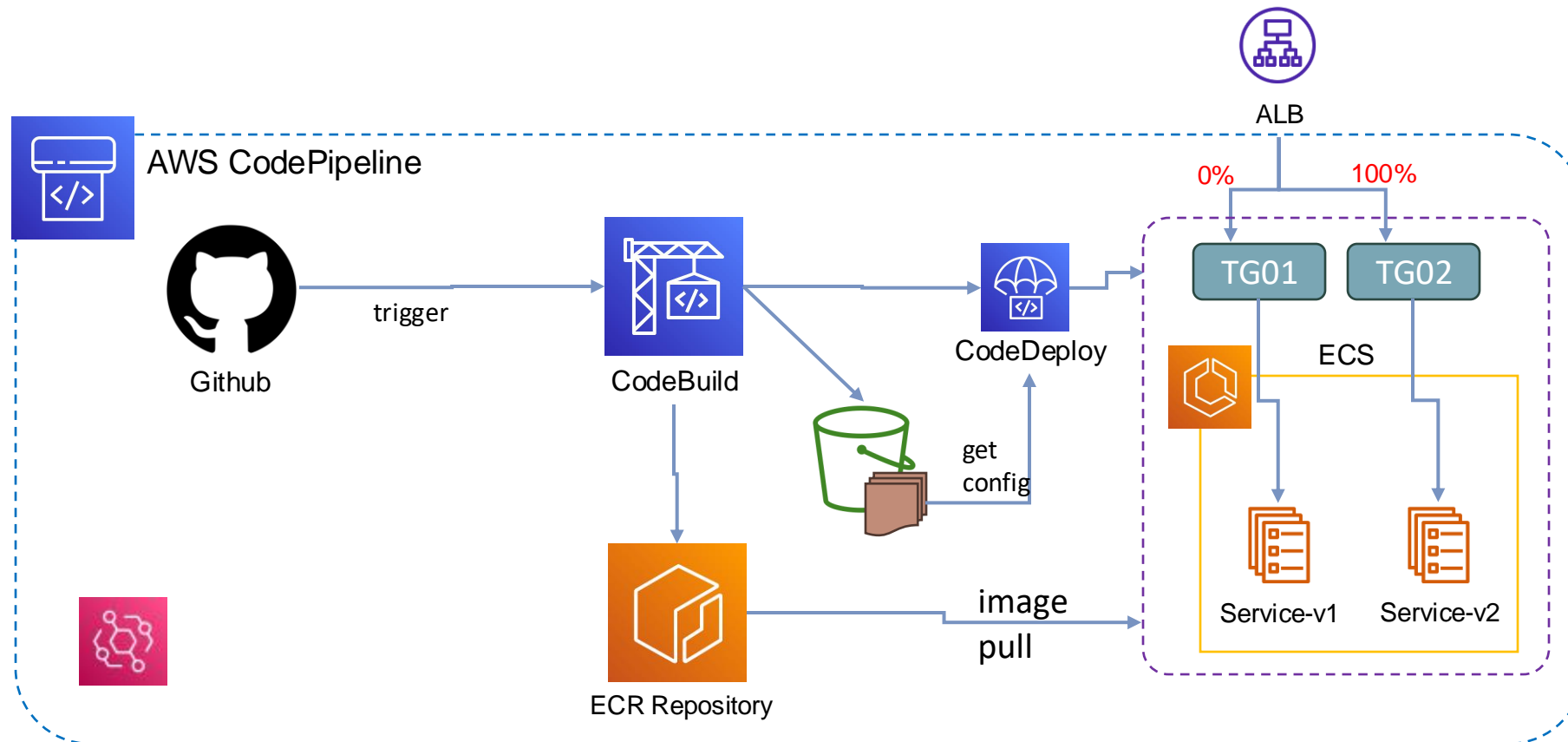
Sơ đồ hệ thống của bài lab



*Lưu ý: Canary trong trường hợp này là service level (không phải cluster level)

Lab8 - Tạo một CodePipeline – Deploy lên ECS sử dụng Canary

Sơ đồ hệ thống của bài lab



*Lưu ý: Canary trong trường hợp này là service level (không phải cluster level)

Lab9 - Tạo bước manual approve trong CodeDeploy.

Yêu cầu: Update Job để thêm các action sau:

- Thêm notification trong các event của CodeDeploy
- Add thêm bước approval trước CodeDeploy job trong Pipeline.
- Thực hiện push code lên Github, chờ đến phase approve.
- Tiến hành Approve và xem kết quả.

Tổng kết & Clear resources

- Terminate EC2 instance (nếu có).
- Xoá beanstalk environments.
- Xoá ECS Cluster (hoặc terminate sử dụng **terraform destroy**).
- Kiểm tra lại NAT Gateway.
- Kiểm tra lại Elastic IP.
- Xoá các image trên ECR repository.
- Github repository, CodeBuild, CodeDeploy, CodePipeline: Giữ lại để tham khảo trong tương lai.

Thanks you and see you in the next chapter!