**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**ĐỒ ÁN CUỐI KỲ**

**----- 🙡 🕮 🙣 -----**



CÔNG NGHỆ ĐÁM MÂY

ĐỀ TÀI: ***CI/CD Pipeline using AWS***

Lớp: 19BITV02

Giảng viên : LÊ HOÀNG VIỆT TUẤN

Nhóm :

Nguyễn Công Huy - 1911548384

Lê Thành Cơ - 1900007999

Bùi Duy Cương - 1911547215

Nguyễn Thiên Tuấn Mai - 1911547105

Phan Thanh Thiên - 1900009380

Nguyễn Thị Thúy Vy - 1911548481

Mục Lục

[**MỞ ĐẦU** 3](#_Toc113735439)

[**CHƯƠNG I : KHÁI NIỆM** 4](#_Toc113735440)

[**1.** **Amazon Web Services là gì?** 4](#_Toc113735441)

[**2.** **Continuous Integration (CI):** 4](#_Toc113735442)

[**3.** **Continuous Delivery (CD):** 4](#_Toc113735443)

[**CHƯƠNG II: QUY TRÌNH** 5](#_Toc113735444)

[**1.** **QUI TRÌNH CI/CD:** 5](#_Toc113735445)

[**CHƯƠNG III: CHI PHÍ VÀ LỢI ÍCH** 6](#_Toc113735446)

[**1.** **Chi phí:** 6](#_Toc113735447)

[**2.** **Lợi ích :** 6](#_Toc113735448)

[**3.** **Ưu điểm và nhược điểm :** 6](#_Toc113735449)

**MỞ ĐẦU**

Hiện nay có rất nhiều những mô hình phát triển phần mềm khác nhau như mô hình thác nước, chữ V, Agile,... Mỗi một mô hình đều có những ưu và nhược điểm riêng. Tuy nhiên, đối mặt với nhu cầu của con người ngày càng tăng như hiện nay thì SDLC lại dần bộc lộ những điểm yếu của mình. Nhiều người đã cho rằng SDLC đã không còn phù hợp như trước đây và đã bị thay thế bởi những mô hình mới có những lợi ích về tốc độ, tự động hóa trong công việc trong đó có CI/CD.Vậy CI/CD là gì? vai trò và cách thức hoạt động của nó như thế nào? Đây là 2 chương trình cơ bản được dùng để tối ưu hóa các thao tác test và build một phần mềm. CI và CD giúp tiết kiệm rất nhiều thời gian trong quy trình tạo lập mã code. Chính vì thế, không khó hiểu khi 2 chương trình này được hầu hết các lập trình viên sử dụng. Để nâng cao tay nghề lập trình của mình, bạn nên chắc chắn đã nắm được thông tin cũng như đặc điểm cơ bản về CI và CD.

**CHƯƠNG I : KHÁI NIỆM**

1. **Amazon Web Services là gì?**

Là một nền tảng điện toán đám mây phát triển toàn diện được cung cấp bởi Amazon.com. Dịch vụ Web đôi khi được gọi là dịch vụ đám mây hoặc các dịch vụ điện toán từ xa. Các dịch vụ AWS đầu tiên đã được đưa ra vào năm 2006 để cung cấp các dịch vụ trực tuyến cho các trang web và các ứng dụng phía máy khách. Để giảm thiểu việc bị mất điện đột ngột và đảm bảo tính mạnh mẽ của hệ thống, AWS đa dạng về địa lý theo khu vực. Các khu vực này có trung tâm ở Đông Mỹ, Tây Mỹ (hai địa điểm), Brazil, Ireland, Singapore, Nhật Bản và Úc. Mỗi khu vực bao gồm nhiều khu vực địa lý nhỏ hơn được gọi là vùng sẵn có.

1. **CI/CD là gì?**

CI/CD là một phương pháp giúp việc cung cấp ứng dụng cho khách hàng được liên tục, bằng cách áp dụng tự động hóa vào các giai đoạn phát triển ứng dụng.

CI / CD là giải pháp cho các vấn đề tích hợp mã mới có thể gây khó khăn cho đội ngũ phát triển và vận hành (hay còn gọi là “địa ngục tích hợp”).

Cụ thể, CI / CD áp dụng tính năng tự động hóa liên tục và giám sát liên tục trong suốt vòng đời của ứng dụng, từ giai đoạn tích hợp và kiểm thử đến phân phối và triển khai.

Các phương pháp kết nối này thường được gọi là quy trình CI/CD (pipeline) và được hỗ trợ bởi các đội ngũ phát triển và vận hành làm việc cùng nhau một cách linh hoạt với một DevOps hoặc một SRE (Kỹ sư quản lý độ tin cậy)

1. **CI là gì? :**

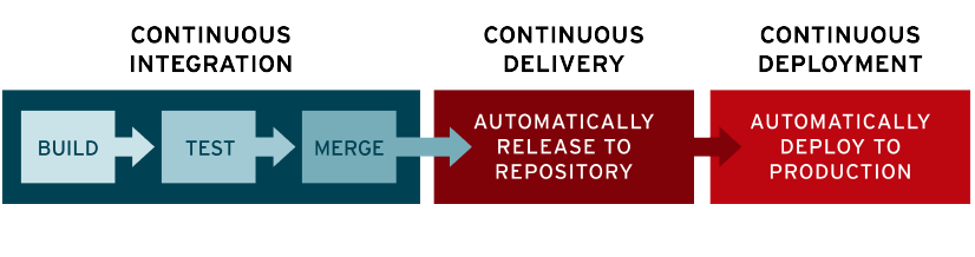
Là tích hợp liên tục (continuous integration), là một quá trình tự động hóa cho các nhà phát triển.CI thành công có nghĩa là các thay đổi mã của một ứng dụng thường xuyên được xây dựng, kiểm tra và tích hợp vào một kho lưu trữ được chia sẻ (shared repository). Đó là giải pháp cho vấn đề có quá nhiều nhánh của một ứng dụng được phát triển cùng lúc và có thể xung đột với nhau.

1. **CD là gì?:**

Là phân phối liên tục (continuous delivery) và / hoặc triển khai liên tục (continuous deployment), là những khái niệm liên quan đôi khi được sử dụng thay thế cho nhau. Cả hai đều nhằm mục đích tự động hóa các giai đoạn tiếp theo của quy trình, nhưng đôi khi chúng được sử dụng riêng biệt để minh họa mức độ tự động hóa đang diễn ra.

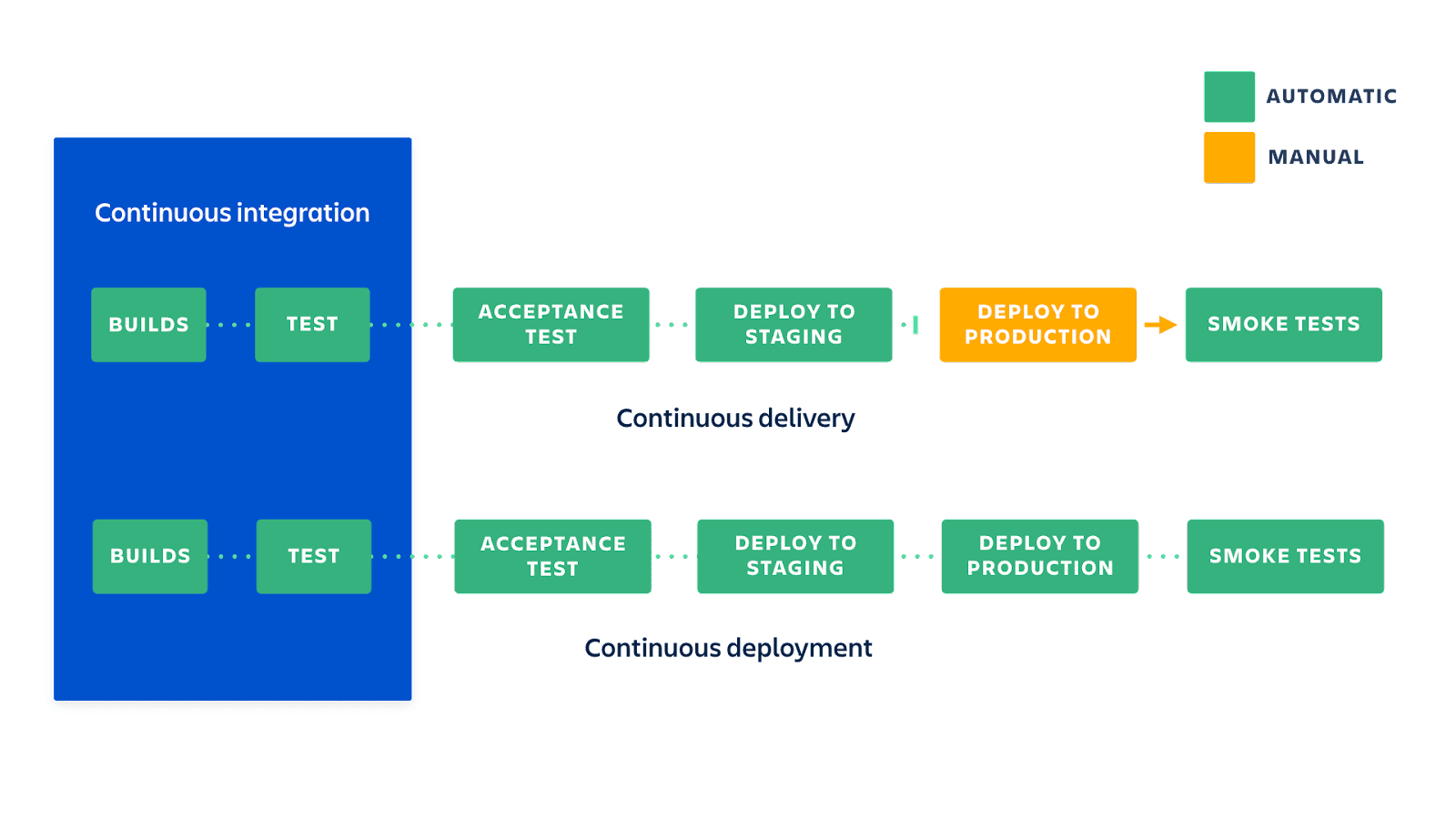
Phân phối liên tục thường nghĩa là các thay đổi của nhà phát triển đối với ứng dụng sẽ tự động được kiểm lỗi và tải lên kho lưu trữ (như GitHub hoặc container registry), nơi đội ngũ vận hành đưa chúng vào môi trường sản xuất trực tiếp (live production). Vì vậy, mục đích của phân phối liên tục là đảm bảo rằng việc triển khai mã mới sẽ tốn ít công sức nhất.

Còn Triển khai liên tục đề cập đến việc tự động phát hành các thay đổi của nhà phát triển từ kho lưu trữ sang production, nơi khách hàng có thể sử dụng được. Nó giải quyết vấn đề quá tải của đội ngũ vận hành với các quy trình thủ công làm chậm quá trình phân phối ứng dụng. Nó được xây dựng dựa trên những lợi ích của việc phân phối liên tục bằng cách tự động hóa giai đoạn tiếp theo trong quy trình.

1. **Quá trình CI/CD**

**CHƯƠNG II: QUY TRÌNH**

1. **QUI TRÌNH CI/CD:**

Trong kỹ thuật phần mềm, tiến trình CI/CD làm cho quá trình phát triển phần mềm nhanh hơn, thời gian release (ra đời) một sản phẩm được rút ngắn rất nhiều, nâng cao hiệu năng làm việc của developer.

* Liên tục tích hợp (CI) : là quá trình diễn ra một cách liên tục không đứt quản trong phần việc develop --> testing --> build --> deploy.
* Liên tục chuyển giao(CD): là quá trình ra đời sản phầm một cách liên tục theo từng phiên bản của phần mềm. (versioning) làm phần mềm dần hoàn thiện hơn cho tới bản release chính thức.
* Quá trình liên tục tích hợp(CD), cần có các thành phần :

Nơi chứa source code của nhà phát triển có thể là các phần mềm như: github, gutlab, bitbucket, ....

Tiến trình testing phần mềm : chạy các unit test cũng như các thông số quét mã nguồn để tìm các lỗi bảo mật, lỗi code lặp lại, ..v.v..

Môi trường build sản phẩm; là môi trường mà khi ra đời một sản phẩm phần mềm, sản phẩm đó chạy trên môi trường đó, ví dụ như môi trường Windows, Linux, iOS và Android.

Môi trường triển khai sản phẩm:

+ Môi trường develop - môi trường dành cho nhà phát triển,

+ Môi trường qc - môi trường dành cho bộ phận kiểm thử phần mềm,

+ Môi trường staging - môi trường dành cho khách hàng; người thuê viết phần mềm

+ Môi trường production - môi trường dành cho công đồng người dùng

* Khái niệm Continuous trong kỹ thuật này là tính liên tục của quá trình và xảy ra một các tự động.
* Để tối ưu quá trình làm việc của một developer thì khi developer merge code lên nhánh ấn định sẵn (ví dụ nhánh develop) thì tiến trình test và build xảy ra ngay tức thì và tự động.
* Nếu build và deploy thành công thì ta có một phiên bản phầm mềm ngay lúc đó (version). Các version được tạo ra và tăng lên tương ứng với qúa trình pháp triển phầm mềm của các developer.

**CHƯƠNG III: CHI PHÍ VÀ LỢI ÍCH**

1. **Chi phí:**

|  |  |  |
| --- | --- | --- |
| Continuous Integration (CI) | Continuous Delivery (CD) | Continuous Deployment |
| - Viết test tự động  - CI server để monitor repository chính  - Devs phải merge code thường xuyên | - CD phải cấu hình đúng và phải bao gồm đầy đủ các mã nguồn  - Quá trình Deployment cần phải được tự động hóa. Một số quy trình đặc biệt có thể chạy manually.  - Team cần phải rõ ràng trong việc features chưa hoàn chỉnh để không ảnh hưởng đến khách hàng. | - Chất lượng testing sẽ quyết định chất lượng relaese.  - Doucoument cần phải luôn theo kịp tốc độ triển khai.  - Các tính năng (features) cần phải được theo dõi tiến độ để đảm bảo sự phối hợp với các bộ phận khác ( Support, Marketing,…) |

1. **Lợi ích :**

|  |  |  |
| --- | --- | --- |
| Continuous Integration (CI) | Continuous Delivery (CD) | Continuous Deployment. |
| - ít bugs và kiểm soát bugs tốt hơn  - các bản release sẽ chất lượng hơn  - chi phí test giảm – CI sever có thể chạy test tự động cực nhanh  - Team có nhiều thời gian hơn để tập trung vào nghiệp vụ. | - Việc Deploy sản phẩm sẽ rất nhanh .  - Tiến độ release sản phẩm sẽ thường xuyên hơn, thúc đẩy khách hàng feedback.  - Đối với những thay đổi nhỏ sẽ có ít áp lực hơn. | - Quá Trình development sẽ nhanh hơn vì không cần phải tạm dừng cho việc Deploy.  - Các bản Relaese sẽ ít rủi ro và dễ khắc phục hơn.  - Khách hàng sẽ thấy việc cải tiến liên tục thường xuyên hơn, và chất lượng được tăng lên mỗi ngày , |

1. **Ưu điểm và nhược điểm :**

\* Một số ưu điểm mà quy trình CI/CD mang lại cho Developer bao gồm:

* Tránh được những lỗi không đáng có: chẳng hạn như lỗi compile (khi đẩy code lên) hoặc các lỗi phát sinh liên quan đến môi trường build sản phẩm.
* Đảm bảo logic (vì quy trình CI/CD có phần automation test), khi Developer xây dựng tính năng mới sẽ không gây ảnh hưởng đến tính năng cũ.
* Giúp tập trung vào công việc bởi quy trình CI/CD mang tính tự động cao nên Developer không cần phải thực hiện việc build và deploy phần mềm/ứng dụng trên máy cá nhân nữa.
* Nâng cao chất lượng code thông qua quy trình, Developer có thể cài đặt những ràng buộc ngay từ đầu.
* hát triển kỹ năng unit test cho Developer thông qua các chỉ số ràng buộc về code coverage (% code đã được cover) được cài đặt trong quy trình CI/CD. Nghĩa là khi phát triển tính năng mới, để không làm giảm chỉ số code coverage, Developer phải ý thức được tầm quan trọng của unit test và chủ động học hỏi, nâng cao các kỹ năng liên quan.
* Tối ưu tốc độ phát triển của sản phẩm thông qua việc theo dõi thời gian build pipeline (các bước chạy test, build, chạy static code analytics (lint check).

\* Bên cạnh những ưu điểm giúp quy trình CI/CD đáng được cân nhắc để áp dụng trong tổ chức thì CI/CD vẫn có những hạn chế cần phải lưu ý như:

* Trong một dự án nếu có quá nhiều Developer cùng tham gia phát triển sản phẩm, sẽ có thời điểm phát sinh nhiều pull request cần được merge vào branch, phải chờ pull request của người trước được merge hoàn tất, sau đó thực hiện update (cập nhật) lại source code (trong trường hợp có thông báo conflict từ Git repository) và phải trải qua các bước test lại từ đầu. Hệ quả là làm gián đoạn thời gian phát triển sản phẩm.
* Vì sử dụng dịch vụ CI/CD của bên service thứ 3 nên nếu service đó gặp vấn đề và bị crash, bị khai tử thì những dự án áp dụng CI/CD cũng bị ảnh hưởng khá nghiêm trọng.