TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

MÔN MẠNG MÁY TÍNH NÂNG CAO

**BÁO CÁO**

TÌM HIỂU VÀ KHAI TRIỂN CI/CD CƠ BẢN

GVLT: **ThS. Lê Ngọc Sơn**

Thành viên:

**20424047 – Nguyễn Ngọc Linh**

**20424048 – Nguyễn Vũ Linh**

**20424049 – Ngô Thị Mĩ Loan**

**20424054 – Trương Ngọc Mỵ**

***TP. Hồ Chí Minh***, ngày 10 tháng 05 năm 2021

Mục lục

[1.](#_gjdgxs) Các khái niệm: 3

[2.](#_30j0zll) Motivations of CI/CD: 3

[3.](#_1fob9te) CI/CD pipelines: 4

[4.](#_3znysh7) Ví dụ về đường ống (Pipelines): 4

[5.](#_2et92p0) Cách thiết kế đường ống CI/CD: 6

[6.](#_tyjcwt) Ưu, nhược điểm 7

[7.](#_3dy6vkm) Thách thức: 8

[8.](#_1t3h5sf) Ứng dụng: 9

[9.](#_4d34og8) Demo: 9

[TÀI LIỆU THAM KHẢO: 9](#_2s8eyo1)

| **STT** | **Họ tên** | **Phân công** | **Mức độ hoàn thành** |
| --- | --- | --- | --- |
| 1 | 20424047 Nguyễn Ngọc Linh | * Tìm kiếm nội dung | 100% |
| 2 | 20424048 Nguyễn Vũ Linh | * Tìm kiếm nội dung * Code demo | 100% |
| 3 | 20424049 Ngô Thị Mỹ Loan | * Thiết kế slide | 100% |
| 4 | 20424054 Trương Ngọc Mỵ | * Tổng hợp nội dung, báo cáo | 100% |

# Các khái niệm:

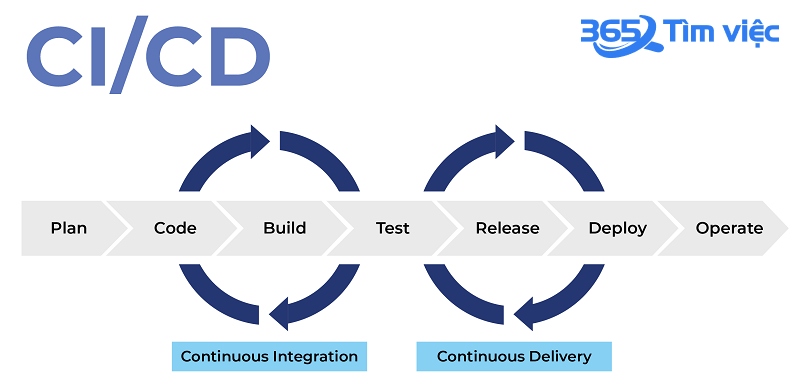
**CI - Continuous Integration:**

Tích hợp liên tục là một quy trình nhằm mục đích tích hợp các thay đổi code mới và giữ cơ sở mã nguồn trong một trạng thái tốt. Một cách để đạt được nó có thể tích hợp các thay đổi mới và tự động kiểm thử mã tổng thể.

**CD - Continuous Delivery vs. Continuous Deployment:**

Mặt khác, phân phối liên tục nhằm mục đích thường xuyên tạo ra các bản phát hành sẵn sàng được triển khai, không nhất thiết trong sản xuất. Bản phát hành có thể được sử dụng trong các giai đoạn beta hoặc trong môi trường sản xuất. Bản phát hành phải nhất quán và miễn phí.

Triển khai liên tục là thực tiễn để phát hành thường xuyên trong sản xuất (cho người dùng cuối). Bản phát hành được tạo ra bởi quá trình phân phối liên tục.

**CI/CD** - Là quá trình tích hợp (integration) thường xuyên, nhanh chóng hơn khi code cũng như thường xuyên cập nhật phiên bản mới (delivery).  


# Motivations of CI/CD:

1. Tăng tốc đưa sản phẩm ra thị trường: xây dựng một bản phát hành tốt trong thời gian ngắn là điểm mấu chốt để giảm thời gian đưa ra thị trường;
2. Tăng độ tin cậy: tự động hóa các bài kiểm tử và có môi trường tiền giai đoạn cho phép bạn giảm thiểu các lỗi;
3. Khắc phục sự cố nhanh chóng: tạo ra bản phát hành thường xuyên có nghĩa là có ít sự khác biệt giữa các bản phát hành, vì vậy làm giảm đi các khâu xử lý sự cố;
4. Tránh căng thẳng khi phát hành: CD cho phép bạn triển khai ít, nhưng thường xuyên, vì vậy bạn ít sợ hãi về các lỗi mỗi khi có bản phát hành.

# CI/CD pipelines:

Quy trình CI/CD được biên dịch trong các đường ống. Bên trong các đường ống là một loạt các hành động / giai đoạn tiếp theo thường bắt đầu từ các thay đổi mã nguồn và dừng chuỗi nếu có lỗi trong một tác vụ.

**Các thành phần của các đường ống CI/CD?**

* Các ràng buộc của kho lưu trữ: thông thường khi hợp nhất một tính năng mới;
* Xây dựng/hành động: biên dịch mã hoặc các hành động khác nhau như gửi email hoặc thực thi tập lệnh;
* Kiểm tra: xác minh xem bản phát hành mới có hoạt động như mong đợi hay không, hành động kiểm tra này có thể là tự động hoặc thủ công;
* Phê duyệt, thủ công hoặc kịp thời: một số loại phê duyệt cần có để tiếp tục quá trình;
* Deploys: hành động cài đặt bản phát hành trong môi trường trực tiếp.

Icon

Description automatically generated

# Ví dụ về đường ống (Pipelines):

Shape

Description automatically generated with medium confidence

Sau khi đẩy mã mới vào kho lưu trữ (Repository), nó được xây dựng để tạo đơn vị phát hành. Giai đoạn xây dựng là thử nghiệm đầu tiên, trên thực tế, nó kiểm tra các lỗi biên dịch.

A picture containing icon

Description automatically generated

Đơn vị phát hành được kiểm tra bằng các bài kiểm tra tích hợp. Các bài kiểm tra phải được chuẩn bị bởi nhóm phát triển hoặc nhóm Đảm bảo chất lượng.

Chúng được tạo ra để đảm bảo rằng mọi tính năng trong quá khứ không bị ảnh hưởng bởi các thay đổi mới và hành vi của các thay đổi mới như mong đợi. Vượt qua các bài kiểm tra, bản phát hành được triển khai ở giai đoạn beta để con người hoặc các công cụ kiểm tra khác kiểm tra.

A picture containing shape

Description automatically generated

Sau khi triển khai ở giai đoạn beta, một email sẽ được gửi đến Chủ sở hữu dịch vụ (hoặc nhóm Đảm bảo chất lượng) để phê duyệt và bắt đầu sản xuất.

Icon

Description automatically generated

Việc triển khai trong sản xuất bao gồm 2 bước: thay đổi DB và cài đặt trong môi trường sản xuất trực tiếp.

Thay đổi DB không phải lúc nào cũng bắt buộc, nó phụ thuộc vào các thay đổi code. Để hoàn chỉnh, hành động này được tách biệt với "Triển khai trên sản xuất".

# Cách thiết kế đường ống CI/CD:

Để thiết kế một đường dẫn, bạn nên hiểu các thành phần liên quan của dịch vụ mà bạn đang phát hành là gì.

Thông thường, bản phát hành bao gồm:

* Mã nhị phân
* Cấu trúc cơ sở dữ liệu hoặc dữ liệu
* Tài sản tĩnh

Mỗi thành phần được đề cập này đều dẫn đến sự thay đổi trong cơ sở hạ tầng mà bạn sử dụng để cung cấp dịch vụ.

Các bước để thiết kế đường ống bao gồm:

1. Xác định những thay đổi phổ biến, tìm những thay đổi có thể ảnh hưởng đến dịch vụ;
2. Xác định các thành phần được cập nhật, bạn phải lặp lại cho mỗi thay đổi và giải quyết tác động của các thay đổi đối với các thành phần là gì;
3. Xác định hành động để thực hiện triển khai / cập nhật
4. Mô hình hóa quy trình, bạn phải mô hình hóa quy trình xem xét các nhu cầu kinh doanh.

# Ưu, nhược điểm

**a. Ưu điểm:**

* Các thay đổi nhỏ hơn, sẽ đơn giản hơn và có ít hậu quả không mong muốn hơn
* Cách ly lỗi đơn giản và nhanh chóng hơn
* Thời gian trung bình để giải quyết (MTTR) ngắn hơn do mã thay đổi nhỏ hơn và cách ly lỗi nhanh hơn.
* Khả năng kiểm tra được cải thiện do những thay đổi nhỏ hơn, cụ thể. Những thay đổi nhỏ hơn này cho phép các xét nghiệm tích cực và tiêu cực chính xác hơn.
* Thời gian đã trôi qua để phát hiện và sửa lỗi thoát sản xuất ngắn hơn với tốc độ phát hành nhanh hơn.
* Việc backlog các khuyết tật không nghiêm trọng sẽ thấp hơn vì các khuyết tật thường được sửa trước khi các áp lực về tính năng khác phát sinh.
* Sản phẩm cải tiến nhanh chóng thông qua việc giới thiệu tính năng nhanh chóng và thay đổi tính năng nhanh chóng.
* Các bản nâng cấp giới thiệu các đơn vị thay đổi nhỏ hơn và ít gây gián đoạn hơn.
* Tốc độ tính năng của sản phẩm CI-CD cao. Tốc độ cao giúp cải thiện thời gian điều tra và vá lỗi.
* Tính năng bật tắt và triển khai blue-green lá cây cho phép giới thiệu liền mạch, có mục tiêu các tính năng sản xuất mới.
* Có thể giới thiệu những thay đổi quan trọng trong những giờ không quan trọng (khu vực). Việc giới thiệu thay đổi giờ không quan trọng này hạn chế tác động tiềm ẩn của sự cố triển khai.
* Chu kỳ phát hành ngắn hơn với các bản phát hành được nhắm mục tiêu và điều này chặn ít tính năng chưa sẵn sàng phát hành hơn.
* Sự tham gia và phản hồi của người dùng cuối trong quá trình phát triển liên tục dẫn đến cải tiến khả năng sử dụng. Chúng ta có thể thêm các yêu cầu mới dựa trên nhu cầu của khách hàng hàng ngày.

**b. Nhược điểm:**

* Đòi hỏi sự tỉnh táo liên tục từ người dùng. Tránh sai quy định trình tự động hóa được thực hiện trước và hết sức thận trọng trong việc chọn đúng thứ tự quy trình
* Code base phải sẵn sàng và được đưa vào sản xuất ngay lập tức khi kết quả hiện tại thành công. Điều này tứ thời có thể dẫn đến sự hoảng loạn cho người dùng
* Các team có thể lập một dashboard mà không phải thành viên nào cũng biết trước. Điều này dẫn đến việc trở thành con mồi của ngụy biện logic
* VÌ CI/CD được điều phối, chúng phải được triển khai đồng bộ với nhau. Cần phải chú ý nhiều và chi tiết đến yếu tố con người để giúp chúng hoạt động
* **Lợi ích:**
* Triển khai code thường xuyên nên dễ phát hiện lỗi => Giảm rủi ro cho dự án
* Khi CI được tích hợp với CD việc chia sẻ code dễ dàng hơn, tăng cường sự hợp tác giữa các thành viên trong đội ngũ phát triển => thân thiện với người dùng
* Khi việc phát hành code trở nên thường xuyên, khoảng cách giữa mã code trong sản xuất và code trong phát triển nhỏ hơn nhiều => vòng tuần hoàn làm việc nhanh hơn
* Toàn bộ quá trình được tự động hóa => giảm các quá trình thủ công
* Khi quá trình phát hành trở nên nhanh hơn, người quản lý dễ dàng thấy được sự thay đổi và đưa ra quyết định kinh doanh nhanh chóng => Phản hồi nhanh hơn

# Thách thức:

Trong đường dẫn CI / CD, có một số vấn đề phổ biến. Một trong số chúng gồm:

**a. Database migration: (Di chuyển CSDL):** cấu trúc của dữ liệu có thể thay đổi trong quá trình phát triển của ứng dụng. Hành động thay đổi cấu trúc được gọi là di chuyển cơ sở dữ liệu. Để cho bản phát hành có thể lặp lại và nhất quán, bản phát hành của bạn nên bao gồm các tập lệnh để thao tác với cơ sở dữ liệu.

Có rất nhiều thư viện tạo các tệp Ngôn ngữ Định nghĩa Dữ liệu / Ngôn ngữ Thao tác Dữ liệu cho phép tiến và lùi với cơ sở dữ liệu của bạn tương ứng với phiên bản mã nguồn.

**b. Monitoring the application (Giám sát ứng dụng):** để kích hoạt rollback sau một bản phát hành bị hỏng hoặc để xác minh rằng một thay đổi mới không làm hỏng phần mềm. Nếu chúng ta đang nói về một ứng dụng web, các kiểm tra thông thường là ping HTTP, xác minh 200 OK của một trang thử nghiệm; và ping HTTP sâu nơi bạn kiểm tra nội dung bên trong trang.

Các bài kiểm tra khác có thể là kiểm tra chức năng, nơi bạn “thẩm vấn” ứng dụng của mình và xác minh phản hồi. Nó có thể được thực hiện mô phỏng các hành động trên Giao diện người dùng hoặc đơn giản là thực hiện gọi một số REST API.

Ngoài ra, các bài kiểm tra phổ biến khác có thể liên quan đến các chỉ số hệ thống, như sử dụng CPU, RAM và hoán đổi giữa chúng.

# Ứng dụng:

Một số phần mềm đã tích hợp CI/CD: Jenkins, Bamboo, TeamCity, Microsoft VSTS, Travis CI, Gitlab CI, Circleci, Codeship, AWS Codepipeline, Go CD, Urbancode, deploy

# Demo:

* ***Video demo:*** <https://drive.google.com/file/d/1ojOW-cQGWvvwYWCSwrdwrLtxpep0Zh-A/view?usp=sharing>

***Hoặc:*** <https://youtu.be/fqMIRiRCh50>

* ***Link‌ ‌github repo:‌*** ‌<https://github.com/nvlinh99/cicd-02>
* ***Link‌ ‌app‌ ‌demo:***‌ ‌<https://demo-cicd-02.herokuapp.com/>

# TÀI LIỆU THAM KHẢO:

1. <https://thenewstack.io/understanding-the-difference-between-ci-and-cd/>
2. <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html>