

# Tìm hiểu lý thuyết về DevOps

Phan Minh Anh Tuấn

[tuam.pma205227@sis.hust.edu.vn](mailto:tuam.pma205227@sis.hust.edu.vn)

Trường Công nghệ Thông tin và Truyền thông  
Đại học Bách Khoa Hà Nội

Ngày 28 tháng 10 năm 2022

# Mục lục

---

## 1. DevOps là gì

## 2. CI/CD

- 2.1 Continuous integration (CI)
- 2.2 Continuous delivery (CD)
- 2.3 Continuous deployment

# DevOps là gì

---

- Được ghép từ Developement (Dev) và Operations (Ops), nghĩa là sự kết hợp của bên phía phát triển sản phẩm và bên vận hành hệ thống rút ngắn quá trình phát triển sản phẩm

# DevOps là gì

---

- Được ghép từ Developement (Dev) và Operations (Ops), nghĩa là sự kết hợp của bên phía phát triển sản phẩm và bên vận hành hệ thống rút ngắn quá trình phát triển sản phẩm
- Một số yếu tố chính của quy trình:
  1. Quá trình tích hợp (Integration) và phân phối liên tục (continuous delivery) hay được gọi là CI/CD
  2. Triển khai và kiểm tra tự động với quy trình tập trung vào Behavior-Driven Design (BDD) hoặc Test-Driven Design (TDD)
  3. Việc triển khai dựa trên thu thập phản hồi từ khách hàng
  4. Giám sát ứng dụng và cơ sở hạ tầng

# CI/CD

---

Có 3 practice chính:

1. Continuous integration (CI)
2. Continuous delivery (CD)
3. Continuous deployment

# Continuous integration (CI)

---

- Là một kĩ thuật phát triển phần mềm, trong đó các thành viên trong một team thường xuyên tích hợp code của họ với nhau thông qua GitHub, GitLab...

# Continuous integration (CI)

---

- Là một kĩ thuật phát triển phần mềm, trong đó các thành viên trong một team thường xuyên tích hợp code của họ với nhau thông qua GitHub, GitLab...
- Mỗi lần tích hợp đều được build và test tự động để phát hiện và sửa lỗi một cách nhanh chóng

# Continuous integration (CI)

---

- Là một kĩ thuật phát triển phần mềm, trong đó các thành viên trong một team thường xuyên tích hợp code của họ với nhau thông qua GitHub, GitLab...
- Mỗi lần tích hợp đều được build và test tự động để phát hiện và sửa lỗi một cách nhanh chóng

⇒ Continuous integration là một quá trình tự động cho phép kiểm tra độ ổn định và hoàn thiện của code mỗi khi thành viên của nhóm thực hiện thêm/sửa đổi



# Triển khai CI

---

- Ta cần có một **Source Code Manager (SCM)** cho phép đồng bộ code của toàn bộ thành viên trong nhóm.  
Phổ biến nhất là **Git**

# Triển khai CI

---

- Ta cần có một **Source Code Manager (SCM)** cho phép đồng bộ code của toàn bộ thành viên trong nhóm.  
Phổ biến nhất là **Git**
- Một **Automatic Build Manager (CI server)** hỗ trợ continuous integration như GitLab CI, Azure Pipelines, GitHub Actions ...

# Triển khai CI

---

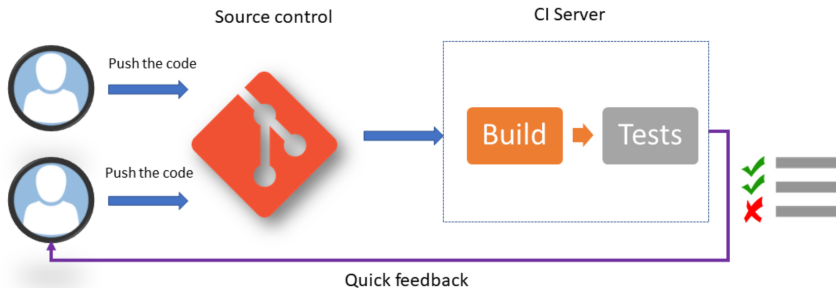
- Ta cần có một **Source Code Manager (SCM)** cho phép đồng bộ code của toàn bộ thành viên trong nhóm.  
Phổ biến nhất là **Git**
- Một **Automatic Build Manager (CI server)** hỗ trợ continuous integration như GitLab CI, Azure Pipelines, GitHub Actions ...

Trong phạm vi môn học, em sử dụng:

- **Git** với vai trò một Source Code Manager
- **GitLab / Azure Pipelines** với vai trò một Automatic Build Manager

# CI pipeline

---



Hình 1: Continuous intergration pipeline

# Continuous delivery (CD)

---

- Mỗi một lần CI hoàn thành, bước tiếp theo là deploy sản phẩm **tự động** tại các môi trường testing hoặc staging và deploy **thủ công** lên môi trường production. Những bài test này bao gồm UI testing, load testing, integration testing, API testing... Tự động hoàn toàn quy trình release phần mềm  
Đó được gọi là Continuous delivery (CD)

# Continuous delivery (CD)

---

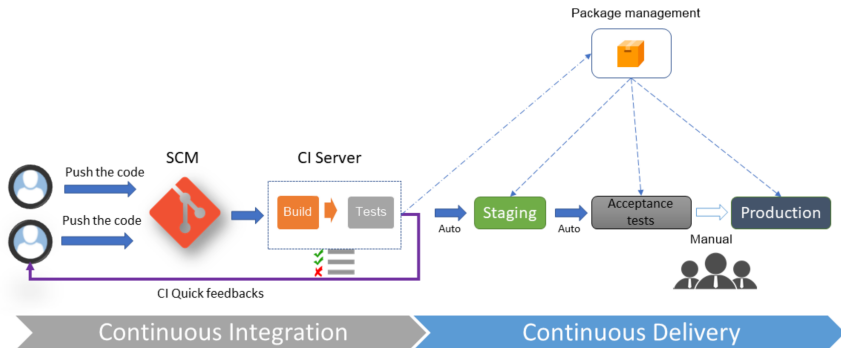
- Mỗi một lần CI hoàn thành, bước tiếp theo là deploy sản phẩm **tự động** tại các môi trường testing hoặc staging và deploy **thủ công** lên môi trường production. Những bài test này bao gồm UI testing, load testing, integration testing, API testing... Tự động hoàn toàn quy trình release phần mềm  
Đó được gọi là Continuous delivery (CD)
- Mục tiêu của CD là test toàn bộ sản phẩm tạo ra  
≠  
CI chỉ test từng thành phần

# Continuous delivery (CD)

---

- Mỗi một lần CI hoàn thành, bước tiếp theo là deploy sản phẩm **tự động** tại các môi trường testing hoặc staging và deploy **thủ công** lên môi trường production. Những bài test này bao gồm UI testing, load testing, integration testing, API testing... Tự động hoàn toàn quy trình release phần mềm  
Đó được gọi là Continuous delivery (CD)
- Mục tiêu của CD là test toàn bộ sản phẩm tạo ra  
≠  
CI chỉ test từng thành phần  
**Ví dụ:** Phần mềm phát triển theo kiến trúc Microservice

# CD pipeline



Hình 2: Continuous delivery pipeline

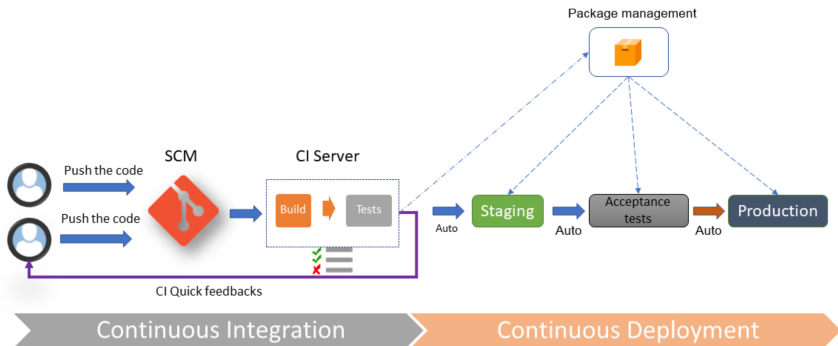


# Continuous deployment

---

- Điểm khác duy nhất giữa Continuous deployment và Continuous delivery là Continuous deployment tự động hóa việc deploy lên môi trường production

# Continuous deployment pipeline



Hình 3: Continuous deployment pipeline

# Continuous delivery may Continuous deployment ?

---

# Continuous delivery hay Continuous deployment ?

---

Câu trả lời: cả hai. Tùy thuộc vào mục tiêu và yêu cầu để chọn practice phù hợp. Trong đó:

# Continuous delivery hay Continuous deployment ?

---

Câu trả lời: cả hai. Tùy thuộc vào mục tiêu và yêu cầu để chọn practice phù hợp. Trong đó:

- Continuous delivery phù hợp với các công ty muốn kiểm soát sản phẩm cuối đến với người dùng
- Continuous deployment sẽ nhanh hơn do tự động hóa cả quy trình cuối.

Chủ đề tuần sau

# **Infrastructure as Code**

# References

---



Mikael Krief (2019) Learning DevOps, First Edition.

**The End**