CI/CD tools

Các công cụ triển khai CI/CD

trong phát triển dự án phần mềm

NỘI DUNG CHÍNH

01 **Giới thiệu về CI/CD**

Một số thông tin cơ bản về CI/CD

02 **SonarQube**

Công cụ phân tích chất lượng code (SAST)

03 **Gitlab**

Công cụ quản lý source code, files cấu hình triển khai sản và thiết lập pipline CICD

04 **ArgoCD**

Công cụ phục vụ việc chuyển giao/triển khai liên tục sản phẩm lên Kubernetes.

05 **Kubernetes**

Công cụ quản lý triển khai ứng dụng dưới dạng Container

06 **Terraform**

Công cụ hỗ trợ thực hiện Infrastructure As Code (IaC)

Theo nguồn: Coursera *(reviewed by VinhNI)*

01.

Giới thiệu về CI/CD

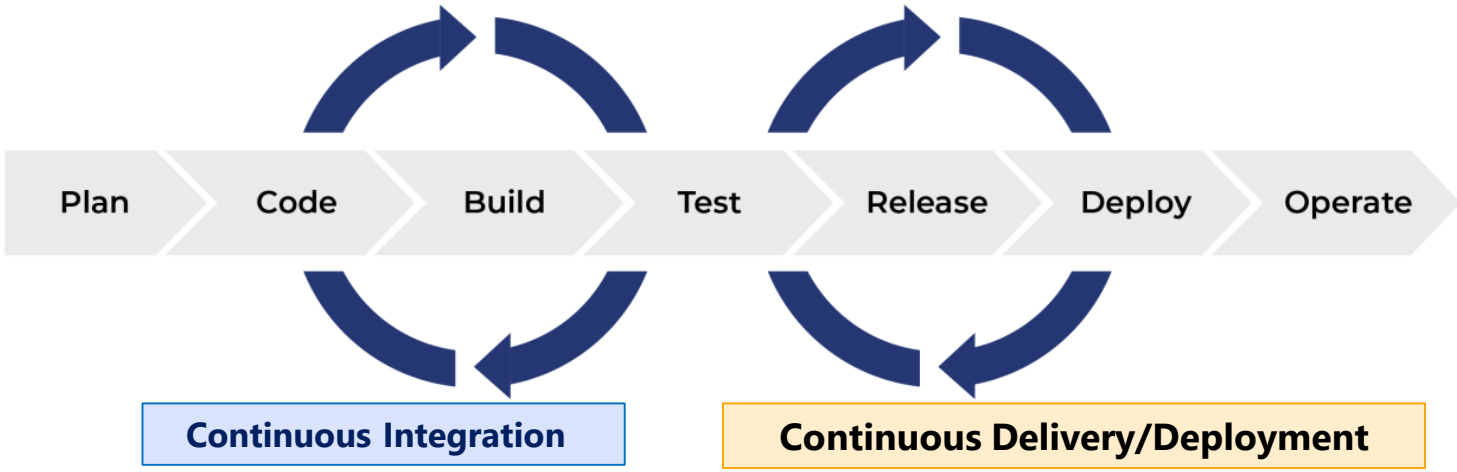
01. Giới thiệu về CI/CD

CI/CD là gì ?

CI/CD là quá trình làm việc liên tục và tự động hóa trong quy trình phát triển phần mềm và chuyển giao sản phẩm

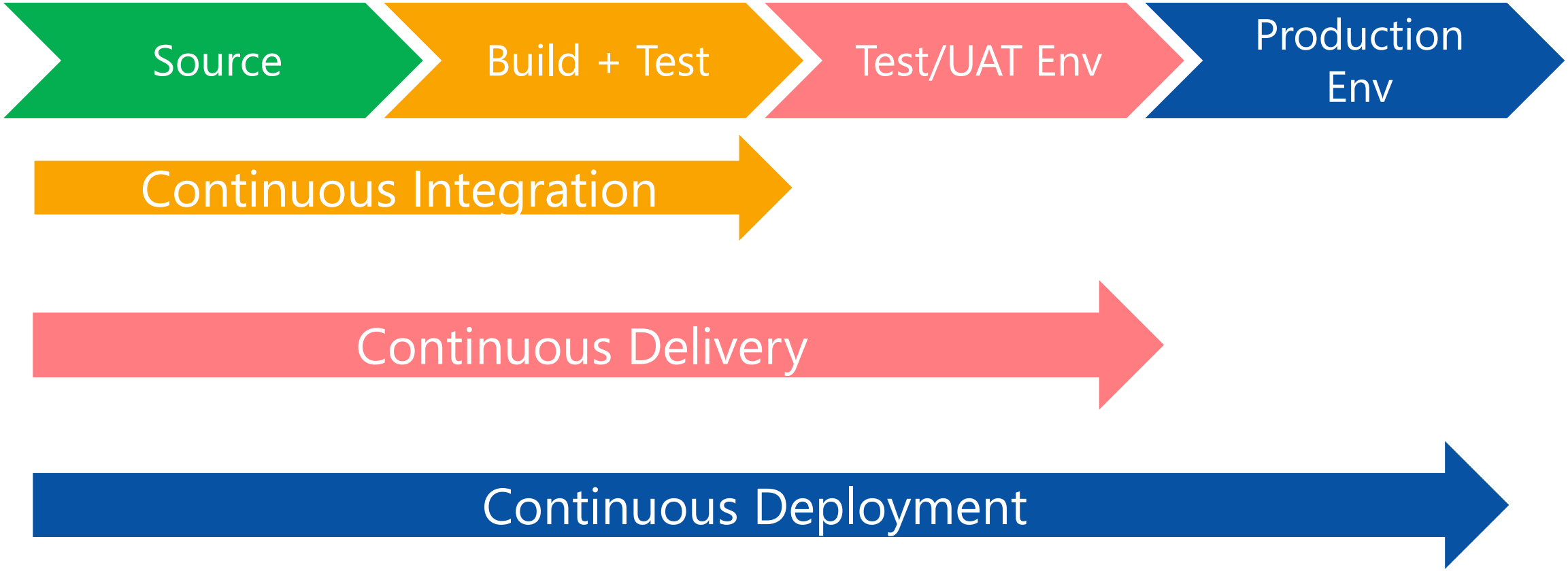
CI - Continuous Integration (tích hợp liên tục). CI giúp các thành viên dự án liên tục tích hợp code vào một kho lưu trữ chung (Gitlab, Github,...) và kiểm tra chất lượng mã nguồn bằng các công cụ tự động (SonarQube,...).

CD - Continuous Delivery (chuyển giao liên tục) hoặc Continuous Deployment (triển khai liên tục). Giúp dự án tự động chuyển sản phẩm phần mềm lên các môi trường như Test, UAT, Stagging, Production.



01. Giới thiệu về CI/CD

CI/CD là gì ?



Continuous Deployment (triển khai liên tục) là mức độ tự động hóa cao hơn, trong đó quá trình xây dựng/triển khai diễn ra tự động bất cứ khi nào có thay đổi đối với code.

01. Giới thiệu về CI/CD

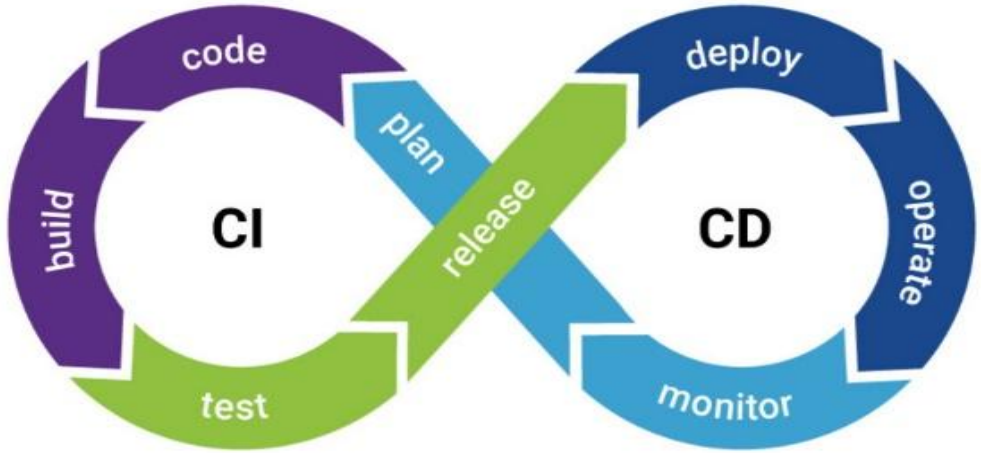
Mục đích của CI/CD

CI - Continuous Integration (tích hợp liên tục)

Mục đích chính giúp phát hiện sớm các lỗi tích hợp, đồng thời tạo ra sự hợp tác tốt trong công việc.

CD - Continuous Delivery (chuyển giao liên tục) hoặc Continuous Deployment (triển khai liên tục).

Giúp giảm thiểu các cản trở vốn có trong quá trình chuyển giao hoặc triển khai. Các bước trong quá trình chuyển giao/triển khai được cấu hình tự động để có thể chuyển giao/triển khai bất cứ lúc nào.



01. Giới thiệu về CI/CD

Lợi ích của CI/CD

Nhờ vào tính năng tự động build, release, deploy, và kiểm tra chất lượng code khi có thay đổi source code, CI/CD giúp tăng tốc độ phát triển và bàn giao sản phẩm, nhận diện lỗi sớm, giúp giảm thiểu lỗi, nâng cao chất lượng sản phẩm.

❖ **Tối ưu hóa thời gian phát triển của dự án:**

- Tự động hóa các thao tác lặp lại thường xuyên
- DEV có thể chủ động biết được đánh giá về source code của mình, không phải chờ đợi đến giai đoạn test thủ công.
- Giảm bớt được thời gian build & deploy source code qua từng giai đoạn
- Giảm bớt chi phí sửa lỗi

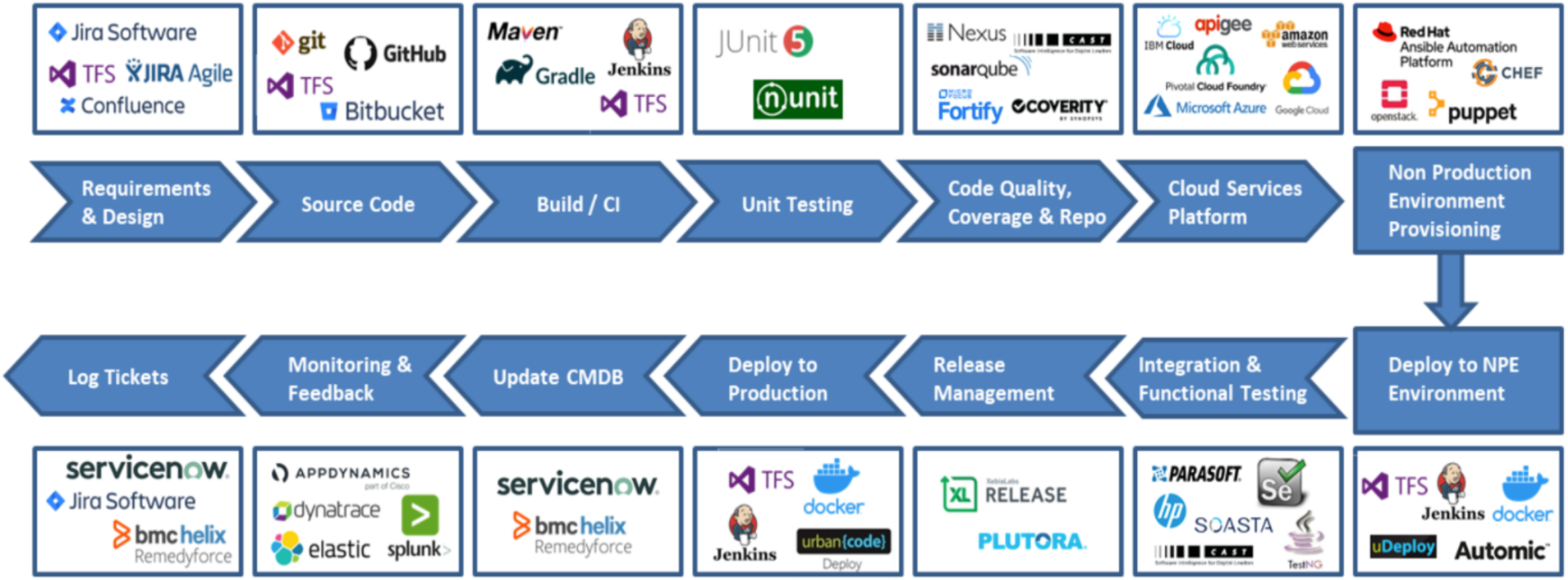
❖ **Tăng chất lượng của sản phẩm đầu ra:**

- Source code được test sớm, phát hiện ra lỗi sớm => Rút kinh nghiệm và tránh các lỗi tương tự trong thời gian sắp tới
- Giảm rủi ro trong quá trình chuyển giao/triển khai vì những lỗi sai do yếu tố con người

❖ **Tối ưu sự hợp tác giữa các thành viên, các công việc**

01. Giới thiệu về CI/CD

CICD Sample toolchains



01. Giới thiệu về CI/CD

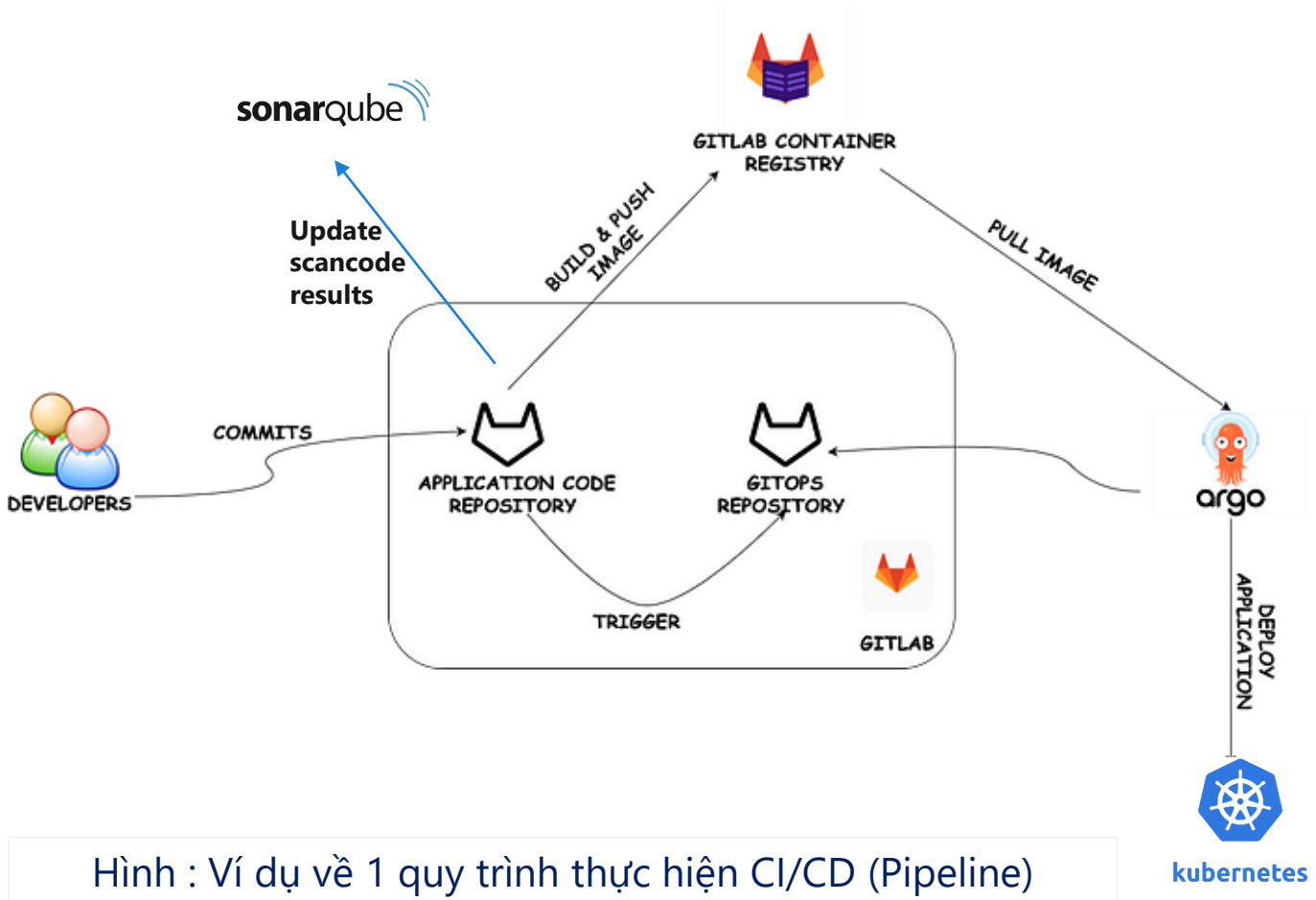
CICD Sample toolchains

Tools	Description	AWS option	Alternative	Note
Code Repository	Code and configuration source control for both application and infrastructure	CodeCommit	GitLab, GitHub, BitBucket, …	
Package management	Control binary and artifacts of software package	CodeArtifact S3	Nexus, Artifactory, …	
Container Registry	Store container images	ECR	Nexus, Harbor, …	
Secret store	Store and manage access to credentials, keys, …	Secret Manager	Vault, Kubernetes secrets, …	
Job Runner	Help to automate the parts of development, operation or security	CodePipeline CodeBuild	Jenkins, Gitlab, CircleCI	
Automation build		N/A	Ant, Maven, Gradle, Helm, …	Programming language
Automated deployment	Help to automate the parts of deployment	CodeDeploy CodePipeline	ArgoCD, FluxCD	

01. Giới thiệu về CI/CD

Quy trình CI/CD

Tùy theo tính chất của từng dự án, trình tự thực hiện CI/CD có thể khác nhau.



Hình : Ví dụ về 1 quy trình thực hiện CI/CD (Pipeline)

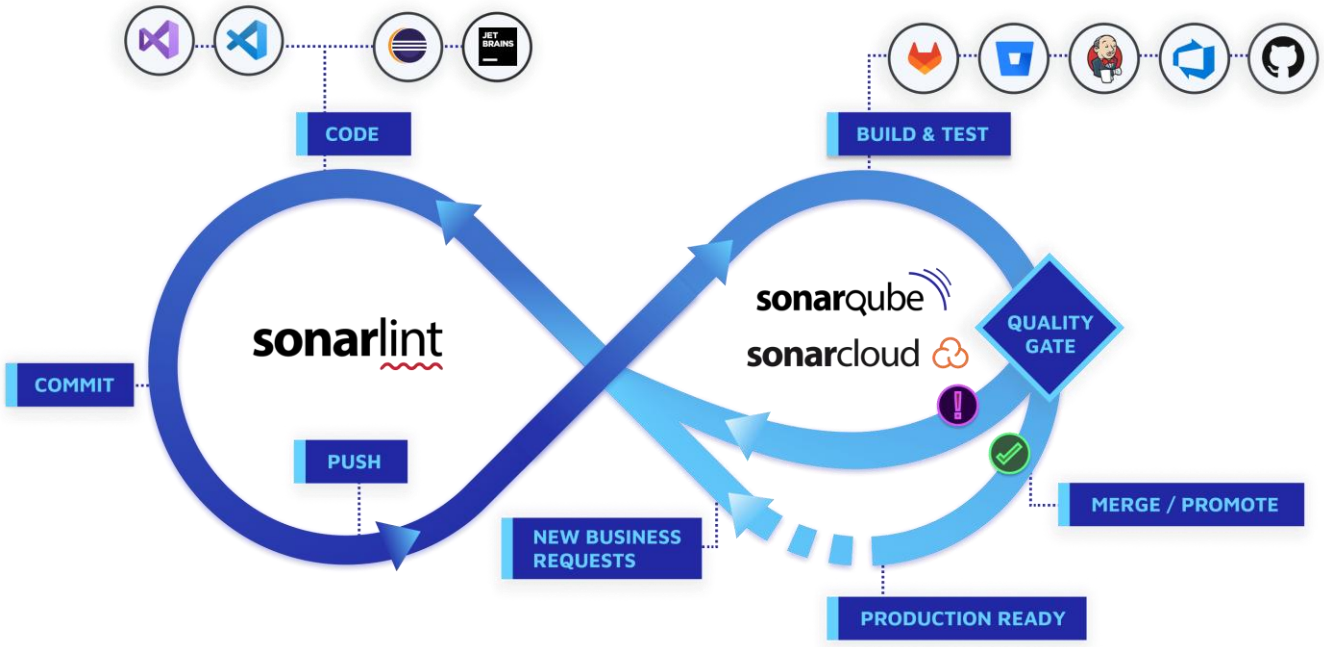
02.

SonarQube

02. Giới thiệu về SonarQube

SonarQube là một nền tảng mã nguồn mở được phát triển bởi SonarSource với mục đích kiểm tra liên tục chất lượng code, review tự động bằng cách scan phân tích code để phát hiện lỗi, đoạn code không tốt, hoặc lỗ hổng bảo mật của rất nhiều ngôn ngữ lập trình.

Sonarqube hiện hỗ trợ nhiều ngôn ngữ lập trình thông dụng, như Java (including Android), C#, PHP, JavaScript, C/C++, COBOL, PL/SQL, PL/I, ABAP, VB.NET, VB6, Python, RPG, Flex, Objective-C, Swift, XML, ...



02. Giới thiệu về SonarQube

Lợi ích của Sonarqube

Sonarqube có khả năng quét (scan) tất cả các dòng code có trong dự án và dựa trên các coding standard tương ứng để đánh giá code của mọi ngôn ngữ trong dự án.

- **Kiểm tra chất lượng mã một cách liên tục** với các báo cáo chi tiết về các bài kiểm tra đơn vị, tiêu chuẩn mã hoá, lỗ hổng bảo mật, độ phủ mã.
- **Giúp các lập trình viên có thể dễ dàng kiểm tra mức độ hiệu quả của mã cod**e để nhanh chóng cảnh báo và đưa ra gợi ý chỉnh sửa ngay khi phát hiện lỗi vi phạm hay nguy cơ vi phạm những nguyên tắc chuẩn chung.
- **Giúp hình thành thói quen cho các lập trình viên và hạn chế tối đa các lỗi có thể xảy** ra trong quá trình code.
- **Giúp giải quyết các vấn đề về quản lý chất lượng code,** bug, code smell, hay những lỗ hổng bảo mật cho source code

02. Giới thiệu về SonarQube

Bộ tiêu chí đánh giá

Ví dụ về 1 bộ tiêu chí cơ bản		
Metric	**Operator**	**Error**
Blocker Issues	is greater than	20
Critical Issues	is greater than	20
Duplicated Lines (%)	is greater than	30.00%
Maintainability Rating	is worse than	C
Reliability Rating	is worse than	C
Security Rating	is worse than	C

- **Blocker Issues:** số lượng lỗi xếp loại blocker
- **Critical Issues:** số lượng lỗi xếp loại critical
- **Duplicated Lines:** tỉ lệ số dòng code lặp lại (các đoạn code giống nhau không đưa vào hàm, thư viện,...)

- **Maintainability Rating: tính dựa trên tỉ lệ của tổng effort ước tính cho việc fix các issue loại "Code Smell" trên tổng effort phát triển tính theo tổng số dòng code.**
  - E: >50%
  - D: từ 21% đến 50%
  - C: từ 11% đến 20%
  - B: từ 6% đến 10%
  - A: <=5%
- **Reliability Rating: tính dựa trên số lượng issue loại "Bug"**
  - E: có từ 1 Blocker Bug trở lên
  - D: có từ 1 Critical Bug trở lên nhưng chưa đạt mức E
  - C: có từ 1 Major Bug trở lên nhưng chưa đạt mức D
  - B: có từ 1 Minor Bug trở lên nhưng chưa đạt mức C
  - A: 0 bug hoặc toàn bug loại Info

- **Security Rating: tính dựa trên số lượng issue loại "Vulnerability"**
  - E: có từ 1 Blocker Vulnerability trở lên
  - D: có từ 1 Critical Vulnerability trở lên nhưng chưa đạt mức E
  - C: có từ 1 Major Vulnerability trở lên nhưng chưa đạt mức D
  - B: có từ 1 Minor Vulnerability trở lên nhưng chưa đạt mức C
  - A: 0 Vulnerability hoặc toàn vulnerability loại Info

02. Giới thiệu về SonarQube

Bộ tiêu chí đánh giá

Các loại Issues

- **Bug:** Lỗi code có thể khiến lỗi tiềm ẩn cho chương trình hoặc lỗi ngừng chương trình cần sửa ngay
- **Vulnerability:** Lỗ hổng bảo mật dễ bị tấn công
- **Code smell:** Code kém chất lượng gây khó khăn hoặc xung đột khi sửa chữa hoặc phát triển thêm code sau này.

Các mức độ nghiêm trọng

- **Blocker:** lỗi gây ảnh hưởng nghiêm trọng đến hoạt động của ứng dụng như "memory leak, unclosed JDBC connection, ..."; hoặc lỗ hổng bảo mật nghiêm trọng cần sửa ngay lập tức
- **Critical:** lỗi gây ảnh hưởng thấp đến hoạt động của ứng dụng hoặc một số lổ hổng bảo mật có độ nghiêm trọng thấp hơn như empty catch block, SQL injection, ... Các lỗi này cần được xem xét ngay để có kế hoạch sửa chữa
- **Major:** Các lỗi hoặc lỗ hổng bảo mật có ảnh hưởng nghiêm trọng đến năng suất của lập trình viên như duplicate code, đoạn code chưa unit test, tham số không sử dụng
- **Minor:** các lỗi hoặc lỗ hổng bảo mật chủ yếu ở mức độ convention như dòng code quá dài, switch với nhỏ hơn 3 cases,...
- **Info:** Chỉ là các phát hiện, không phải lỗi hay lỗ hổng bảo mật

03.

Gitlab

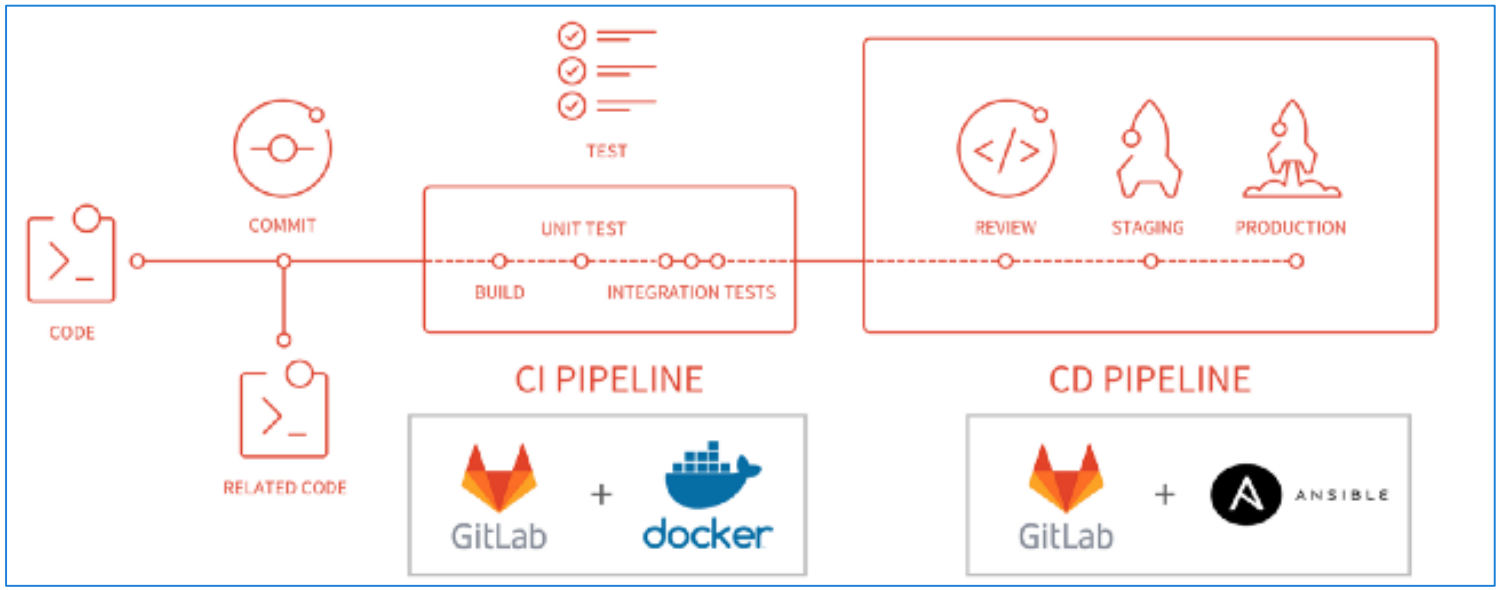
03. Gitlab

Giới thiệu

GitLab là hệ thống mã nguồn mở dựa trên hệ thống máy chủ Git dùng để quản lý source code

Gitlab CI/CD là một giải pháp tích hợp có tính năng kết hợp hệ thống quản lý source code kèm theo cơ chế triển khai CI/CD cho dự án.

GitOps là phương pháp tận dụng tính năng của Git giúp kiểm soát các thay đổi về Infrastructure-as-Code và cấu hình triển khai CD, với quy trình tự động đảm bảo trạng thái hiện tại của hệ thống phản ánh trạng thái trong repo.



04.

ArgoCD

04. ArgoCD

ArgoCD là một CD tools phục vụ việc chuyển giao/triển khai liên tục sản phẩm lên Kubernetes.

ArgoCD có thể lấy code đã được cập nhật từ Git và triển khai trực tiếp lên Kubernetes.

Nó cho phép Dev quản lý cả cấu hình cơ sở hạ tầng và cập nhật ứng dụng trong một hệ thống.

04. ArgoCD

Flow mẫu:

1. Dev thực hiện các thay đổi đối với ứng dụng, commit sự thay đổi trên Git.
2. Tích hợp liên tục được kích hoạt, kết quả là docker image mới được lưu vào registry.
3. Dev tạo pull request về sự thay đổi ở các manifests Kubernetes. Việc này có thể thực hiện tự động.
4. Pull request được xem xét và các thay đổi được merge vào nhánh chính. Điều này kích hoạt một webhook thông báo cho Argo CD rằng đã có thay đổi được thực hiện.
5. Argo CD sao chép repo và so sánh ứng dụng mới được cập nhật với trạng thái ứng dụng hiện tại trên Kubernetes để thực hiện việc đồng bộ
6. Kubernetes sử dụng bộ điều khiển của mình để điều chỉnh các thay đổi cần thiết.
7. Argo CD theo dõi tiến trình và sẽ thông báo khi dụng đã được đồng bộ hóa xong.
8. ArgoCD cũng theo dõi các thay đổi trong Kubernetes và loại bỏ chúng nếu chúng không khớp với cấu hình hiện tại trong Git.

04. ArgoCD

Một số lợi ích của ArgoCD

- Tự động triển khai các ứng dụng tới các môi trường được chỉ định
- Hỗ trợ nhiều công cụ quản lý cấu hình/templates (Kustomize, Helm, plain-YAML)
- Khả năng quản lý và triển khai trên nhiều cụm K8s
- Tích hợp SSO (OIDC, OAuth2, LDAP, SAML 2.0, GitHub, GitLab, Microsoft, LinkedIn)
- Rollback/Roll-anywhere cho bất kỳ cấu hình ứng dụng nào đã committed trên Git
- Tự động phát hiện và hiển thị sai lệch cấu hình
- Đồng bộ hóa tự động hoặc thủ công các ứng dụng với trạng thái mong muốn

05.

Kubernetes

05. Kubernetes

Kubernetes là một nền tảng mã nguồn mở, có thể giúp tự động hóa việc quản lý, mở rộng và triển khai ứng dụng dưới dạng Container. Kubernetes là một hệ sinh thái lớn và có tốc độ phát triển nhanh chóng. Các dịch vụ, sự hỗ trợ và công cụ được tích hợp có sẵn rộng rãi.

05. Kubernetes

Kubernetes cung cấp

Service discovery và load balancing

Kubernetes có thể expose một container sử dụng DNS hoặc địa chỉ IP của riêng nó. Nếu lượng traffic truy cập đến một container cao, Kubernetes có thể load balance (cân bằng tải) và phân phối lưu lượng mạng (network traffic) để việc triển khai được ổn định.

Điều phối bộ nhớ

Kubernetes cho phép bạn tự động mount một hệ thống lưu trữ mà bạn chọn, như local storages, public cloud providers, v.v.

Tự động rollouts và rollbacks

05. Kubernetes

Kubernetes cung cấp

Tự phục hồi

Kubernetes khởi động lại các containers bị lỗi, thay thế các container, xoá các container không phản hồi lại cấu hình health check do người dùng xác định

Quản lý cấu hình và bảo mật

Kubernetes cho phép bạn lưu trữ và quản lý các thông tin nhạy cảm như: password, OAuth token và SSH key.

06.

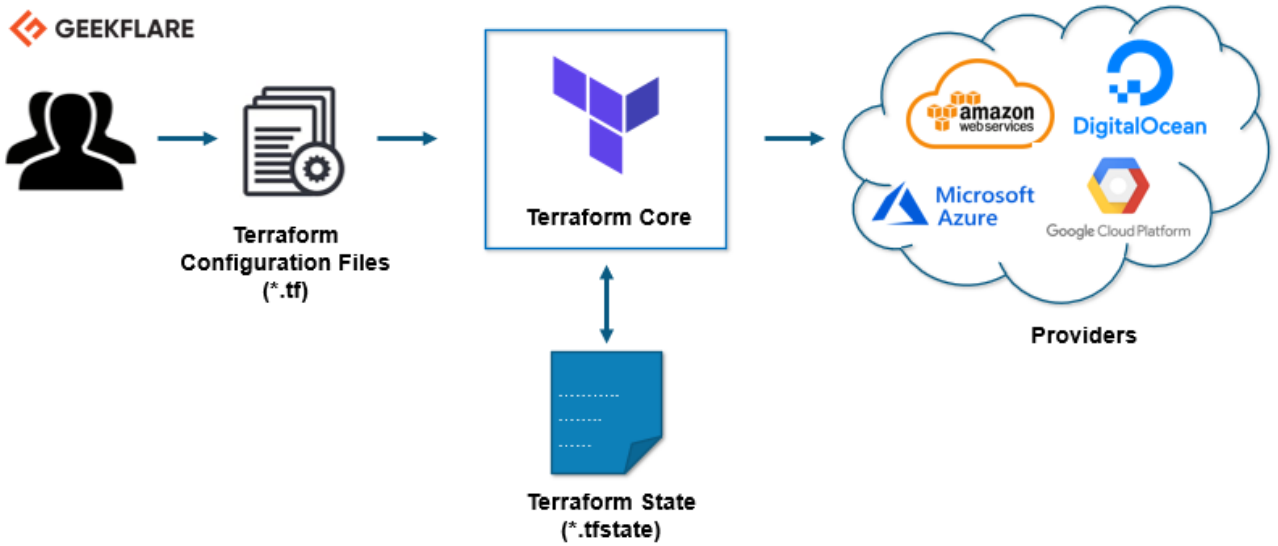
Terraform

06. Terraform

HashiCorp Terraform

Terraform là một công cụ mã nguồn mở giúp người dùng định nghĩa và lưu trữ thông tin tài nguyên bên trong hạ tầng hệ thống thông qua các file code.

Đây cũng là một trong những công cụ IaC (Infrastructure as code) phổ biến nhất tính tới thời điểm hiện tại



06. Terraform

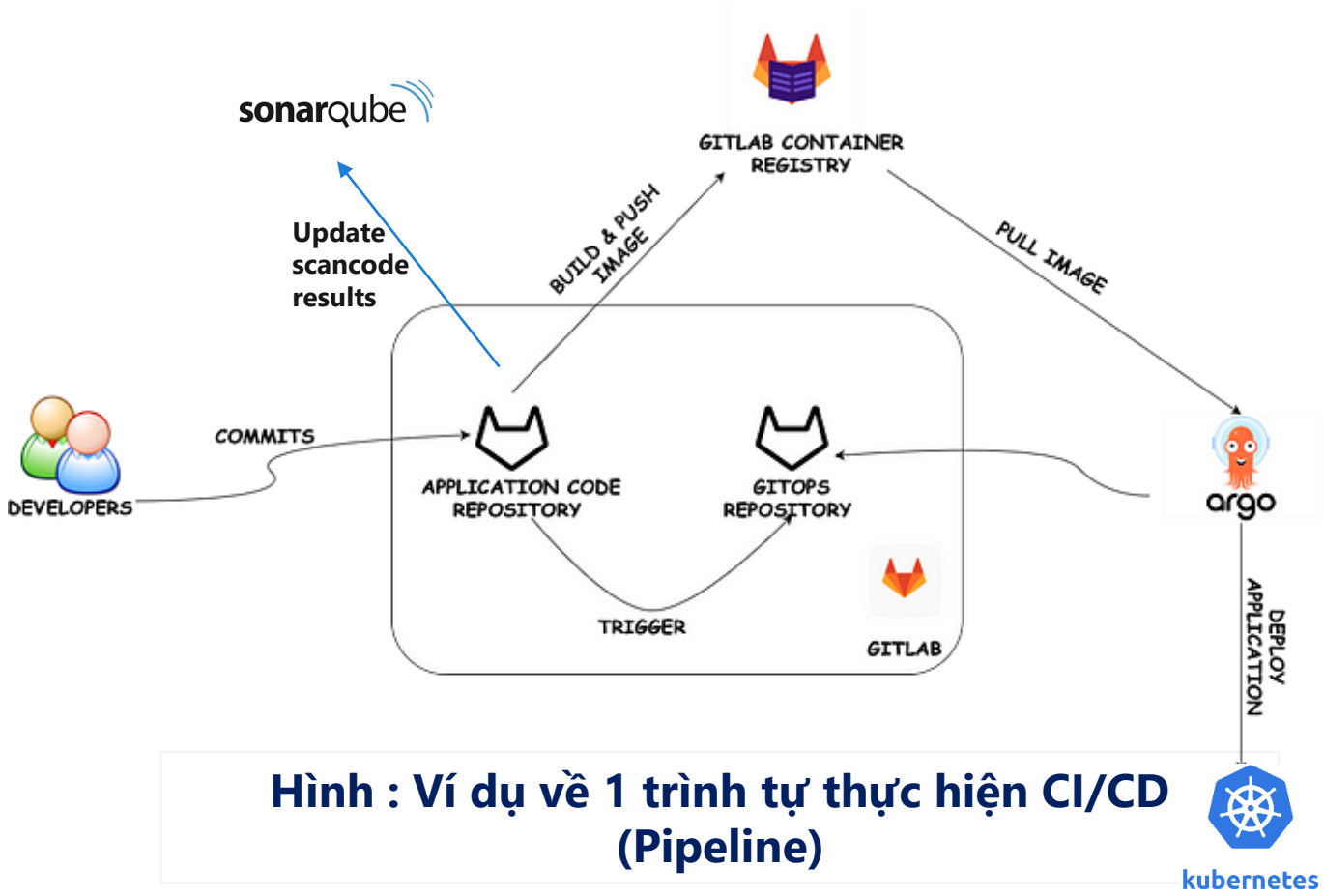
Ưu điểm

- Mã nguồn mở và miễn phí
- Dễ sử dụng
- Tối ưu thời gian triển khai tài nguyên
- Hỗ trợ nhiều cloud
- Dễ dàng tích hợp CI/CD

01. Giới thiệu về CI/CD

Quy trình CI/CD

Tùy theo tính chất của từng dự án, trình tự thực hiện CI/CD có thể khác nhau.



Hình : Ví dụ về 1 trình tự thực hiện CI/CD (Pipeline)

Thank you!