ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

A blue logo with a planet and a flower

Description automatically generated

**Chủ đề:**

**Kubernetes Based on CI/CD**

**Môn học: Công nghệ DevOps và ứng dụng**

**Giáo viên hướng dẫn: ThS Lê Anh Tuấn**

|  |  |  |
| --- | --- | --- |
| STT | HỌ VÀ TÊN | MSSV |
| 1 | Tô Lý Tiến Đạt | 21520712 |
| 2 | Nguyễn Minh Nhàn | 21520374 |

*Hồ Chí Minh, ngày ......... tháng ......... năm 2024*

**Mục Lục**

[LỜI CÁM ƠN 4](#_Toc186029432)

[CHƯƠNG 1: GIỚI THIỆU 5](#_Toc186029433)

[1. Bối cảnh 5](#_Toc186029434)

[2. Giải pháp 5](#_Toc186029435)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 5](#_Toc186029436)

[1. Kubernetes 5](#_Toc186029437)

[a. Giới thiệu 5](#_Toc186029438)

[b. Chức năng 5](#_Toc186029439)

[2. Jenkins. 6](#_Toc186029440)

[**a.** **Giới thiệu.** 6](#_Toc186029441)

[**b.** **Lợi ích.** 6](#_Toc186029442)

[**c.** **Ứng dụng.** 6](#_Toc186029443)

[3. Github 7](#_Toc186029444)

[**a.** **Giới thiệu.** 7](#_Toc186029445)

[**b.** **Lợi ích.** 7](#_Toc186029446)

[**c.** **Ứng dụng.** 7](#_Toc186029447)

[4. Docker 7](#_Toc186029448)

[**a.** **Giới thiệu.** 7](#_Toc186029449)

[**b.** **Lợi ích.** 7](#_Toc186029450)

[**c.** **Ứng dụng.** 7](#_Toc186029451)

[5. Maven 7](#_Toc186029452)

[**a.** **Giới thiệu.** 7](#_Toc186029453)

[**b.** **Lợi ích** 7](#_Toc186029454)

[**c.** **Ứng dụng** 7](#_Toc186029455)

[6. Sonarqube 7](#_Toc186029456)

[**a.** **Giới thiệu.** 7](#_Toc186029457)

[**b.** **Lợi ích** 8](#_Toc186029458)

[**c.** **Ứng dụng** 8](#_Toc186029459)

[7. ArgoCD 8](#_Toc186029460)

[**a.** **Giới thiệu** 8](#_Toc186029461)

[**b.** **Lợi ích** 8](#_Toc186029462)

[**c.** **Ứng dụng** 8](#_Toc186029463)

[8. Aqua Trivy 8](#_Toc186029464)

[**a.** **Giới thiệu** 8](#_Toc186029465)

[**b.** **Lợi ích** 8](#_Toc186029466)

[**c.** **Ứng dụng** 8](#_Toc186029467)

[CHƯƠNG III: KIẾN TRÚC HỆ THỐNG 9](#_Toc186029468)

[CHƯƠNG IV: TRIỂN KHAI 10](#_Toc186029469)

[1. Demo 10](#_Toc186029470)

[Nguồn Tham Khảo 11](#_Toc186029471)

# LỜI CÁM ƠN

Trước tiên, em xin gửi lời cảm ơn sâu sắc đến thầy Lê Anh Tuấn, người đã tận tình hướng dẫn và cung cấp những kiến thức quý báu trong quá trình tôi thực hiện bài tiểu luận này. Những chỉ dẫn tận tâm từ thầy không chỉ giúp em định hướng đúng đắn mà còn tạo động lực để em hoàn thành bài báo cáo này một cách tốt nhất.

Không thể thiếu là sự hỗ trợ từ các bạn đồng hành, đội nhóm trong suốt quá trình làm đồ án. Sự đoàn kết, chia sẻ kinh nghiệm, và tinh thần trách nhiệm của các bạn là yếu tố quan trọng góp phần vào thành công của đồ án này.

Dù đã cố gắng hết sức, đồ án không thể tránh khỏi những thiếu sót. Chúng tôi rất mong nhận được sự góp ý của thầy cô và các bạn để tiếp tục hoàn thiện hơn nữa.

**Trân trọng cảm ơn**

# CHƯƠNG 1: GIỚI THIỆU

## Bối cảnh

Trong thời đại công nghệ phát triển vượt bậc, các doanh nghiệp ngày càng phụ thuộc vào các ứng dụng phần mềm để tối ưu hóa hoạt động kinh doanh và cung cấp trải nghiệm tốt hơn cho khách hàng. Việc phát triển, kiểm thử và triển khai ứng dụng nhanh chóng, đáng tin cậy trở thành một yêu cầu cấp thiết. Tuy nhiên, nhiều tổ chức gặp khó khăn trong việc quản lý vòng đời ứng dụng do sự phức tạp của các quy trình thủ công, thiếu sự tự động hóa, và khó khăn trong việc đảm bảo tính ổn định khi triển khai ứng dụng trên quy mô lớn.

Kubernetes, một nền tảng mạnh mẽ để quản lý container, đã nổi lên như một giải pháp hiệu quả giúp giải quyết các thách thức về triển khai và mở rộng hệ thống. Khi kết hợp Kubernetes với các pipeline CI/CD (Continuous Integration/Continuous Deployment), các tổ chức có thể tự động hóa quy trình phát triển phần mềm, từ việc viết mã nguồn, kiểm thử, đến triển khai, đảm bảo tính liên tục và độ tin cậy cao của ứng dụng.

## Giải pháp

Đồ án này tập trung vào việc xây dựng một hệ thống CI/CD dựa trên Kubernetes, nhằm:

* Tự động hóa toàn bộ quy trình phát triển phần mềm: Tích hợp các công cụ CI/CD như Jenkins hoặc ArgoCD với Kubernetes để tự động hóa từ kiểm thử đến triển khai.
* Đảm bảo tính sẵn sàng cao và khả năng mở rộng: Sử dụng Kubernetes để triển khai ứng dụng trên các cluster với khả năng mở rộng linh hoạt và duy trì tính ổn định.
* Đơn giản hóa quy trình triển khai: Cung cấp các pipeline dễ dàng quản lý, giúp nhóm phát triển nhanh chóng triển khai các tính năng mới mà không làm gián đoạn hệ thống.
* Tăng cường bảo mật và chất lượng phần mềm: Tích hợp các công cụ quét mã nguồn, kiểm thử bảo mật trong pipeline CI/CD để đảm bảo chất lượng sản phẩm.

Hệ thống này sẽ được triển khai trên một mô hình cluster Kubernetes với các ứng dụng mẫu minh họa quy trình CI/CD từ đầu đến cuối. Kết quả không chỉ giúp tăng cường hiệu quả trong việc phát triển phần mềm mà còn cung cấp một giải pháp thực tế và mạnh mẽ cho các tổ chức muốn ứng dụng DevOps vào quy trình làm việc của mình.

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT

## Kubernetes

### Giới thiệu

Kubernetes là một nền tảng mã nguồn mở, linh hoạt và có khả năng mở rộng, dùng để quản lý Các khối lượng công việc và dịch vụ container hóa, hỗ trợ cấu hình khai báo và tự động hóa. Hệ sinh thái của Kubernetes rất lớn và đang phát triển nhanh chóng. Các dịch vụ, hỗ trợ và công cụ dành cho Kubernetes có sẵn rộng rãi.

### Chức năng

* **Khám phá dịch vụ và cân bằng tải**: Kubernetes có thể truy cập một container thông qua tên DNS hoặc địa chỉ IP riêng của nó. Nếu lưu lượng truy cập vào container cao, Kubernetes có thể cân bằng tải và phân phối lưu lượng mạng để đảm bảo triển khai ổn định.
* **Điều phối lưu trữ**: Kubernetes cho phép bạn tự động gắn kết một hệ thống lưu trữ tùy chọn, chẳng hạn như lưu trữ cục bộ, nhà cung cấp dịch vụ đám mây công cộng, và nhiều loại khác.
* **Tự động triển khai và hoàn tác**: Bạn có thể mô tả trạng thái mong muốn cho các container được triển khai bằng Kubernetes. Hệ thống sẽ điều chỉnh trạng thái thực tế để khớp với trạng thái mong muốn ở tốc độ được kiểm soát. Ví dụ: Kubernetes có thể tự động tạo container mới, loại bỏ container cũ, và chuyển tài nguyên sang container mới.
* **Đóng gói tự động**: Bạn cung cấp cho Kubernetes một cụm các node để chạy các tác vụ container hóa. Bạn chỉ định lượng CPU và bộ nhớ (RAM) mà mỗi container cần. Kubernetes sẽ tối ưu hóa cách phân bổ container vào các node để tận dụng tốt nhất tài nguyên.
* **Tự phục hồi**: Kubernetes khởi động lại các container bị lỗi, thay thế container, tiêu diệt container không phản hồi kiểm tra sức khỏe (health check) mà bạn định nghĩa, và chỉ quảng bá chúng tới client khi chúng sẵn sàng phục vụ.
* **Quản lý bí mật và cấu hình**: Kubernetes cho phép bạn lưu trữ và quản lý thông tin nhạy cảm như mật khẩu, token OAuth, và khóa SSH. Bạn có thể triển khai và cập nhật các bí mật hoặc cấu hình ứng dụng mà không cần phải xây dựng lại hình ảnh container hay để lộ bí mật trong cấu hình hệ thống.
* **Thực thi tác vụ hàng loạt**: Ngoài việc quản lý dịch vụ, Kubernetes có thể xử lý các tác vụ hàng loạt và CI, thay thế container khi chúng gặp lỗi nếu cần.
* **Mở rộng ngang**: Dễ dàng mở rộng hoặc thu nhỏ ứng dụng bằng một lệnh đơn giản, giao diện người dùng, hoặc tự động dựa trên mức sử dụng CPU.
* **Hỗ trợ IPv4/IPv6 kép**: Phân bổ địa chỉ IPv4 và IPv6 cho Pods và Services.
* **Thiết kế để mở rộng**: Thêm tính năng vào cụm Kubernetes của bạn mà không cần thay đổi mã nguồn gốc.

## Jenkins.

1. **Giới thiệu.**

Jenkins Jlà nền tảng hàng đầu trong lĩnh vực CI/CD, giúp tự động hóa các công đoạn trong quy trình phát triển phần mềm như build, kiểm thử, và triển khai. Được xây dựng dựa trên nền tảng mã nguồn mở, Jenkins có thể dễ dàng tích hợp với nhiều công cụ DevOps khác, từ quản lý mã nguồn, containerization đến quản lý chất lượng.

1. **Lợi ích.**

* Tích hợp linh hoạt: Với hơn 1.000 plugin, Jenkins có thể tích hợp với nhiều công cụ khác nhau, từ GitHub, Docker, Maven đến các nền tảng triển khai như Kubernetes.
* Pipeline dưới dạng mã: Jenkinsfile cung cấp cách định nghĩa pipeline như một đoạn mã, giúp dễ dàng quản lý và tái sử dụng.
* Phân tán công việc: Jenkins hỗ trợ phân tán khối lượng công việc qua nhiều agent, tăng khả năng mở rộng và xử lý đồng thời nhiều tác vụ.

1. **Ứng dụng.**

* Jenkins thường được sử dụng để tự động hóa quy trình phát triển phần mềm cho các đội ngũ DevOps.
* Trong dự án CI/CD, Jenkins giúp thực hiện chuỗi công việc: lấy mã từ GitHub, xây dựng ứng dụng bằng Maven, kiểm thử tự động với SonarQube và đóng gói container bằng Docker.
* Ngoài ra, Jenkins có thể triển khai các ứng dụng đã được build lên Kubernetes hoặc các dịch vụ đám mây.

## Github

1. **Giới thiệu.**

GitHub là nền tảng quản lý mã nguồn dựa trên hệ thống kiểm soát phiên bản Git, được thiết kế để hỗ trợ cộng tác giữa các lập trình viên và đội phát triển phần mềm. GitHub không chỉ là nơi lưu trữ mã nguồn mà còn cung cấp một hệ sinh thái phát triển toàn diện với khả năng cộng tác, quản lý version control và tự động hóa quy trình thông qua GitHub Actions.

1. **Lợi ích.**

* Hỗ trợ cộng tác: Cho phép nhiều lập trình viên làm việc trên cùng một dự án thông qua các nhánh (branches) và pull request.
* GitHub Actions: Hỗ trợ tích hợp CI/CD trực tiếp trên GitHub, giúp tự động hóa các quy trình phát triển.
* Bảo mật mã nguồn: Tích hợp quét bảo mật, kiểm tra lỗi phụ thuộc, và bảo vệ các nhánh chính.

1. **Ứng dụng.**

GitHub là nơi lưu trữ mã nguồn cho các dự án phần mềm, đồng thời là nền tảng để tích hợp với các công cụ như Jenkins, SonarQube, và ArgoCD. Kết hợp với Jenkins để kéo mã tự động từ các repository và triển khai ứng dụng mới. GitHub Actions giúp thực hiện các công việc tự động hóa như kiểm thử, build, và triển khai ngay trên nền tảng.

## Docker

1. **Giới thiệu.**

Docker là một nền tảng containerization, cho phép đóng gói ứng dụng và tất cả các phụ thuộc của nó vào các container, đảm bảo tính nhất quán khi triển khai trên nhiều môi trường khác nhau.

1. **Lợi ích.**

* Nhẹ và nhanh: Container chỉ chứa các phần cần thiết của ứng dụng, giảm dung lượng và tăng tốc độ triển khai.
* Khả năng tái sử dụng: Dockerfile định nghĩa cách tạo container, giúp dễ dàng tái tạo và chia sẻ container.
* Triển khai dễ dàng: Hỗ trợ tích hợp với Kubernetes để quản lý container trên quy mô lớn.

1. **Ứng dụng.**

Docker được sử dụng để đóng gói ứng dụng trong môi trường phát triển, kiểm thử và sản xuất.

## Maven

1. **Giới thiệu.**

Maven là một công cụ quản lý dự án mạnh mẽ, thường được sử dụng cho các dự án Java để tự động hóa quy trình build, kiểm thử, và quản lý thư viện phụ thuộc.

1. **Lợi ích**

* Quản lý phụ thuộc tự động: Maven tự động tải xuống và quản lý các thư viện cần thiết cho dự án, giảm thiểu lỗi do thiếu thư viện.
* Dễ dàng mở rộng: Hỗ trợ nhiều plugin cho các tác vụ như biên dịch, kiểm thử, và triển khai.
* Cấu hình tập trung: File pom.xml cho phép định nghĩa tất cả các thông tin cần thiết cho dự án, từ phụ thuộc, build đến kiểm thử.

1. **Ứng dụng**

Maven được sử dụng để tự động hóa quá trình xây dựng phần mềm trong các dự án Java, thường kết hợp với Jenkins để tích hợp vào pipeline CI/CD. Maven được sử dụng để chuẩn bị ứng dụng cho việc đóng gói container với Docker. Ngoài ra, Maven cũng đóng vai trò quan trọng trong việc chạy các unit test hoặc tích hợp với SonarQube để phân tích chất lượng mã.

## Sonarqube

1. **Giới thiệu.**

SonarQube là một công cụ phân tích mã nguồn mạnh mẽ, giúp phát hiện lỗi, cải thiện cấu trúc mã, và đảm bảo các tiêu chuẩn chất lượng phần mềm.

1. **Lợi ích**

* Phân tích đa ngôn ngữ: Hỗ trợ hơn 25 ngôn ngữ lập trình như Java, Python, JavaScript.
* Báo cáo chi tiết: Phát hiện các vấn đề về code smell, nợ kỹ thuật, lỗ hổng bảo mật và đưa ra khuyến nghị cải thiện.
* Tích hợp với CI/CD: Tích hợp dễ dàng với Jenkins, GitHub Actions để tự động hóa việc phân tích mã trong pipeline.

1. **Ứng dụng**

SonarQube thường được sử dụng trong giai đoạn kiểm thử mã nguồn của pipeline CI/CD để đảm bảo chất lượng phần mềm trước khi triển khai. Trong các dự án DevOps, SonarQube giúp đảm bảo mã nguồn đáp ứng các tiêu chuẩn chất lượng và không chứa các lỗ hổng bảo mật nghiêm trọng. Ngoài ra, SonarQube còn cung cấp thông tin về "nợ kỹ thuật" (technical debt), giúp các đội ngũ ưu tiên sửa chữa và tối ưu hóa mã.

## ArgoCD

1. **Giới thiệu**

ArgoCD là một công cụ GitOps mã nguồn mở, giúp triển khai ứng dụng lên Kubernetes một cách tự động và đảm bảo trạng thái của ứng dụng được đồng bộ với trạng thái mong muốn.

1. **Lợi ích**

* GitOps: Trạng thái mong muốn của ứng dụng được lưu trữ trong Git, giúp dễ dàng quản lý và theo dõi thay đổi.
* Tự động đồng bộ: Giám sát và tự động cập nhật cluster Kubernetes để khớp với cấu hình trong Git.
* Dễ sử dụng: Cung cấp giao diện web và dòng lệnh để quản lý ứng dụng.

1. **Ứng dụng**

ArgoCD được sử dụng để triển khai các ứng dụng Kubernetes trong mô hình GitOps, đảm bảo triển khai nhanh chóng và đáng tin cậy. Trong pipeline CI/CD, ArgoCD tự động triển khai container đã được build lên các cluster Kubernetes. Nó cũng giúp các đội DevOps phát hiện sự khác biệt giữa trạng thái thực tế và trạng thái mong muốn của ứng dụng, từ đó đồng bộ hóa tự động hoặc cảnh báo kịp thời.

## Aqua Trivy

1. **Giới thiệu**

Aqua Trivy là một công cụ mã nguồn mở dùng để quét và phát hiện các lỗ hổng bảo mật trong container, mã nguồn, và các cấu hình cơ sở hạ tầng.

1. **Lợi ích**

* Quét nhanh và chính xác: Phát hiện các lỗ hổng trong container image, Git repository và các tệp cấu hình Kubernetes.
* Tích hợp dễ dàng: Hoạt động tốt với các pipeline CI/CD, Docker, Kubernetes.
* Báo cáo theo mức độ nguy hiểm: Phân loại lỗ hổng thành các cấp độ như Critical, High, Medium, và Low.

1. **Ứng dụng**

Aqua Trivy được tích hợp vào pipeline CI/CD để quét các lỗ hổng bảo mật trong các container trước khi triển khai lên môi trường sản xuất. Trong DevOps, Trivy giúp đảm bảo các ứng dụng được xây dựng trên cơ sở hạ tầng an toàn, không chứa các lỗ hổng nghiêm trọng. Ngoài ra, Trivy còn được sử dụng để kiểm tra bảo mật các repository Git hoặc file cấu hình, đảm bảo hệ thống tuân thủ các tiêu chuẩn bảo mật.

# CHƯƠNG III: KIẾN TRÚC HỆ THỐNG

A diagram of a software company

AI-generated content may be incorrect.

Hình 1. Mô hình kiến trúc Kubernetes based on CI/CD

# CHƯƠNG IV: TRIỂN KHAI

## Demo

Link demo: [Link Demo](https://drive.google.com/drive/folders/1b4ui5SdN3v1ce6kIv3HCKKBSUmNvFdwe?usp=sharing)

# Nguồn Tham Khảo

[1] <https://kubernetes.io/docs/home/>

[2] <https://www.jenkins.io/doc/>

[3] <https://docs.github.com/en>

[4] <https://docs.docker.com/>

[5] <https://docs.sonarsource.com/sonarqube-server/latest/>

[6] <https://argo-cd.readthedocs.io/en/stable/>

[7] <https://www.aquasec.com/products/trivy/>

[8] <https://docs.dman.cloud/tutorial-documentation/>

[9] <https://github.com/dmancloud/complete-prodcution-e2e-pipeline>