

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng công cụ quản lý tham số DevOps và cấu hình CI/CD để tự động triển khai các dự án phần mềm

NGUYỄN DUY CHIẾN

chien.nd205059@sis.hust.edu.vn

Ngành Công Nghệ Thông Tin

Giảng viên hướng dẫn: TS. Vũ Thị Hương Giang

Chữ kí GVHD

ThS. Nguyễn Mạnh Tuấn

Chữ kí GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 06/2024

LỜI CẢM ƠN

Trong suốt quá trình thực hiện đồ án tốt nghiệp, em đã nhận được rất nhiều sự giúp đỡ, quan tâm, động viên của gia đình, thầy cô, bạn bè.

Trước hết, em xin gửi lời cảm ơn chân thành nhất đến cô Vũ Thị Hương Giang, người đã tận tình chỉ bảo, hướng dẫn em trong suốt quá trình thực hiện đồ án. Cô đã dành rất nhiều thời gian, công sức, chuyên môn kiến thức để ân tình, sâu sắc giúp em hoàn thành đồ án một cách tốt nhất.

Em cũng xin gửi lời cảm ơn chân thành đến các thầy cô giáo trong khoa Công Nghệ Thông Tin Việt Nhật, đã truyền đạt cho em những kiến thức quý báu, giúp em có được nền tảng vững chắc để thực hiện đồ án.

Em xin cảm ơn gia đình, những người thân yêu, câu lạc bộ Hesdpi Nichibu, những người bạn, người anh chị luôn dành cho em sự quan tâm, động viên, ủng hộ trong suốt quá trình không chỉ là học tập và thực hiện đồ án, mà còn về tinh thần, kinh nghiệm.

Cuối cùng, em xin cảm ơn chính bản thân mình đã luôn cố gắng, nỗ lực hết mình để hoàn thành đồ án. Với điều kiện thời gian thực hiện cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao kiến thức chuyên môn của mình và phục vụ tốt hơn trong công việc thực tế.

Một lần nữa, em xin chân thành cảm ơn tất cả mọi người đã giúp đỡ em trong suốt quá trình thực hiện đồ án.

Em xin chân thành!

LỜI CAM KẾT

Họ và tên sinh viên: Nguyễn Duy Chiến
Điện thoại liên lạc: 0967002666
Email: chien.nd205059@sis.hust.edu.vn
Lớp: CNTT Việt Nhật 04 - K65
Hệ đào tạo: Cử nhân CNTT Việt - Nhật

Tôi – *Nguyễn Duy Chiến* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Vũ Thị Hương Giang* và *ThS. Nguyễn Mạnh Tuấn*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 30 tháng 6 năm 2024
Tác giả ĐATN

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong môi trường CNTT hiện nay, mỗi phần mềm có kiến trúc khác nhau thì được triển khai bởi các công cụ, quy trình khác nhau ví dụ mô hình Monolithic sử dụng Docker Image, chạy thành Container đơn lẻ, hoặc cũng có thể chạy trực tiếp mã nguồn trên môi trường triển khai; hay như mô hình Microservice sử dụng Kubernetes, Docker swarm, tách biệt từng quy trình triển khai cho từng service. Hệ thống phần mềm sẽ được triển khai trên nhiều môi trường triển khai, chia thành các môi trường Dev/Test, Staging, Production nhằm kiểm thử, kiểm tra độ hài lòng của người dùng, độ bao phủ của Test Case, độ tích hợp với hệ thống của mã nguồn mới. Quy trình tích hợp tự động và triển khai tự động (CI/CD) sẽ được thực hiện liên tục, phân chia thành các giai đoạn triển khai. Việc đóng gói mã nguồn và truyền vào các tham số theo từng giai đoạn triển khai để thay đổi mã nguồn phù hợp với môi trường triển khai, bởi mỗi môi trường triển khai sẽ có mục đích triển khai, tài nguyên và lưu lượng sử dụng khác nhau. Thông thường việc cập nhật các tham số sẽ do các lập trình viên thao tác thủ công, phải thực hiện nhiều thao tác mới truy cập tới vị trí của các tham số được lưu trữ theo dạng file, từ đó thực hiện khởi động lại quy trình CI/CD nhằm áp dụng các tham số mới đó. Hơn nữa, trong trường hợp lập trình viên cập nhật tham số nhằm giữa các môi trường có thể ảnh hưởng trực tiếp đến hiệu suất sản phẩm phần mềm.

Trong đồ án này, em đã thực hiện khảo sát công việc quản lý tham số cho quy trình CI/CD của các lập trình viên. Hơn nữa có thể cập nhật các tham số đó vào quy trình CI/CD ngay sau khi cập nhật mà không cần thêm nhiều thao tác thủ công. Em đã phân tích yêu cầu sử dụng tham số cho các giai đoạn và môi trường triển khai của quy trình CI/CD. Từ đó kết hợp với nền tảng thực hiện CI/CD nhằm cập nhật các tham số mới lên quy trình CI/CD mỗi khi có thay đổi.

Từ kết quả phân tích và khảo sát trên, em đã thực hiện phân tích, thiết kế hệ thống, xây dựng và phát triển ứng dụng web quản lý tham số truyền vào trong quy trình CI/CD. Ứng dụng được xây dựng và hoạt động trên `https://param-store.datn.live` có đầy đủ chức năng chính như: quản lý Parameter theo từng Project, tự động thực hiện lại quy trình CI/CD mà tham số mới được cập nhật, quản lý giai đoạn, môi trường triển khai cho quy trình CI/CD của từng Project, sử dụng file script nhằm lấy tham số đúng giai đoạn, môi trường triển khai cho quy trình CI/CD, quản lý các dự án, quản lý người dùng và các chức năng khác: phân quyền người dùng vào các dự án, thống kê số lần cập nhật tham số, số lần truy vấn tham số, lưu lịch sử thay đổi tham số trong dự án.

Từ khoá: DevOps, CI/CD, parameter management, automatic deployment

ABSTRACT

In the current IT environment, each software has different architectures and is deployed using various tools and processes. For example, the Monolithic model utilizes Docker Images, running as standalone containers, or can also run source code directly on the deployment environment. On the other hand, the Microservice model uses Kubernetes, Docker Swarm, and separates deployment processes for each service. Software systems are deployed across multiple environments, divided into Dev/Test, Staging, and Production environments to test, check user satisfaction, coverage of test cases, and integration of new source code with the system. Continuous Integration and Continuous Deployment (CI/CD) processes are continuously executed, divided into deployment stages. The source code is packaged, and parameters are passed in according to each deployment stage to adjust the source code to fit the deployment environment. Each deployment environment has different deployment purposes, resources, and usage loads. Typically, updating parameters is manually done by developers, requiring many steps to access the location where the parameters are stored as files, then restarting the CI/CD process to apply the new parameters. Moreover, if a developer mistakenly updates parameters between environments, it can directly affect the software product's performance.

In this project, I have conducted a survey on the parameter management work for the CI/CD process of developers. Additionally, the parameters can be updated into the CI/CD process immediately after being updated without many manual steps. I analyzed the parameter usage requirements for the stages and deployment environments of the CI/CD process. From there, I combined with the CI/CD platform to update new parameters into the CI/CD process whenever there is a change.

Based on the analysis and survey results, I have analyzed, designed the system, built, and developed a web application for managing parameters passed into the CI/CD process. The application is now online at <https://param-store.datn.live/> is built with all the main features such as: managing Parameters by Project, automatically re-executing the CI/CD process when new parameters are updated, managing stages and deployment environments for each Project's CI/CD process, using script files to fetch parameters for the correct stage and deployment environment for the CI/CD process, managing projects, managing users, and other functions: user role management in projects, statistics on the number of parameter updates, parameter query counts, and storing parameter change history within the project.

Keywords: DevOps, CI/CD, parameter management, automatic deployment

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	3
1.4 Bố cục đồ án	4
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	6
2.1 Khảo sát hiện trạng	6
2.1.1 Khảo sát một số công cụ.....	6
2.1.2 Tổng kết.....	7
2.2 Tổng quan chức năng	8
2.2.1 Biểu đồ use case tổng quát	9
2.2.2 Biểu đồ phân rã use case Quản lý tổ chức	10
2.2.3 Biểu đồ phân rã use case Quản lý dự án.....	11
2.2.4 Biểu đồ phân rã use case Quản lý tham số	12
2.2.5 Quy trình nghiệp vụ	13
2.3 Đặc tả chức năng	16
2.3.1 Đặc tả use case thêm tham số	16
2.3.2 Đặc tả use case sửa tham số	19
2.3.3 Đặc tả use case thêm Agent.....	21
2.3.4 Đặc tả use case sửa Agent	22
2.3.5 Đặc tả use case sửa thông tin dự án.....	24
2.3.6 Đặc tả use case tạo người dùng trong tổ chức.....	26
2.3.7 Đặc tả use case thêm người dùng vào dự án	27

2.4 Yêu cầu phi chức năng	29
2.4.1 Hiệu năng.....	29
2.4.2 Tính dễ sử dụng	29
2.4.3 Độ tin cậy.....	29
2.4.4 Giao diện	29
2.4.5 Các yêu cầu khác	29
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	31
3.1 Quy trình CICD.....	31
3.2 Github Actions	32
3.3 Github API	32
3.4 Công nghệ, công cụ phát triển và cài đặt ứng dụng	32
3.4.1 FrontEnd.....	32
3.4.2 Backend	33
3.4.3 Shell Script.....	34
3.4.4 PostgreSQL	34
3.4.5 Công cụ hỗ trợ lập trình.....	34
3.5 Công cụ, công nghệ triển khai.....	35
3.5.1 Nginx.....	35
3.5.2 Docker	36
3.5.3 Triển khai phần mềm lên Internet	36
3.6 Kết chương.....	36
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG	37
4.1 Tổng quan	37
4.2 Thiết kế kiến trúc.....	37
4.2.1 Lựa chọn kiến trúc phần mềm	37
4.2.2 Thiết kế tổng quan.....	39

4.2.3 Thiết kế chi tiết gói	42
4.3 Thiết kế chi tiết.....	46
4.3.1 Thiết kế giao diện	46
4.3.2 Thiết kế lớp	49
4.3.3 Thiết kế cơ sở dữ liệu	56
4.4 Xây dựng ứng dụng.....	62
4.4.1 Thư viện và công cụ sử dụng.....	62
4.4.2 Kết quả đạt được	63
4.4.3 Minh họa các chức năng chính	64
4.5 Kiểm thử.....	69
4.6 Triển khai	71
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	72
5.1 Xác định Parameter trong quy trình CICD	72
5.1.1 Vấn đề.....	72
5.1.2 Giải pháp	72
5.1.3 Kết quả	72
5.2 Phát triển script nhằm lấy các Parameter chính xác	74
5.2.1 Vấn đề.....	74
5.2.2 Giải pháp	74
5.2.3 Kết quả	75
5.3 Tự động ánh xạ giữa tham số và mã nguồn	75
5.3.1 Vấn đề.....	75
5.3.2 Giải pháp	76
5.3.3 Kết quả	76
5.4 Tự động triển khai tham số cho quy trình CICD mỗi khi có thay đổi.....	78
5.4.1 Vấn đề.....	78

5.4.2 Giải pháp	78
5.4.3 Kết quả	79
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	80
6.1 Kết luận	80
6.2 Hướng phát triển.....	81

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quát	9
Hình 2.2	Use case Quản lý tổ chức	10
Hình 2.3	Use case Quản lý người dùng và vai trò	10
Hình 2.4	Use case Quản lý dự án	11
Hình 2.5	Use case Quản lý tham số	12
Hình 2.6	Quy trình nghiệp vụ tạo và cài đặt dự án	13
Hình 2.7	Quy trình nghiệp vụ tạo và cấu hình Agent	14
Hình 2.8	Quy trình nghiệp vụ thực thi file script yêu cầu truy vấn tham số	15
Hình 2.9	Quy trình nghiệp vụ cập nhật và tự động triển khai tham số	16
Hình 3.1	Quy trình CI/CD áp dụng cho phát triển sản phẩm đồ án	31
Hình 4.1	Kiến trúc tổng quan của phần mềm	37
Hình 4.2	Phân cấp tầng mã nguồn máy chủ	39
Hình 4.3	Biểu đồ gói phía Client	40
Hình 4.4	Biểu đồ gói phía Server	41
Hình 4.5	Thiết kế chi tiết gói routes	42
Hình 4.6	Thiết kế chi tiết gói middlewares	43
Hình 4.7	Thiết kế chi tiết gói controllers	44
Hình 4.8	Thiết kế chi tiết gói modules	45
Hình 4.9	Thiết kế chi tiết gói models	46
Hình 4.10	Sitemap giao diện của ứng dụng	47
Hình 4.11	Thiết kế giao diện màn danh sách các Parameter	48
Hình 4.12	Thiết kế giao diện màn danh sách các Agent	48
Hình 4.13	Thiết kế giao diện màn Overview của Project	49
Hình 4.14	Thiết kế giao diện màn Tracking của Project	50
Hình 4.15	Biểu đồ trình tự cập nhật Parameter	55
Hình 4.16	Biểu đồ trình tự cập nhật thông tin Project	56
Hình 4.17	Thiết kế cơ sở dữ liệu	57
Hình 4.18	Giao diện đăng nhập	64
Hình 4.19	Giao diện danh sách Project	64
Hình 4.20	Giao diện danh sách Parameter trong Project	65
Hình 4.21	Giao diện danh sách Agent trong Project	65
Hình 4.22	Giao diện lịch sử cập nhật của User và Agent Project	66
Hình 4.23	Giao diện thông tin chung của Project (1)	66