

The Saigon International
University



Khóa luận
tốt nghiệp

Thành phố Hồ Chí Minh - 2024

KHÓA LUẬN TỐT NGHIỆP

Ngành

KỸ THUẬT PHẦN MỀM

Đề tài

**XÂY DỰNG PHẦN MỀM QUẢN LÝ HỌC VỤ
TẠI SIU DÙNG KIẾN TRÚC MICROSERVICE**

Giảng viên hướng dẫn

Th.S Trần Hàm Dương

Sinh viên

Vũ Nhật Chương

Mã sinh viên: **71012001858**



**The Saigon
International
University**

Lewis Campus

Email: admission@siu.edu.vn

Website: www.siu.edu.vn

LỜI CAM ĐOAN

Tôi xin cam đoan rằng đề tài tốt nghiệp “Xây dựng phần mềm quản lý học vụ tại SIU dùng kiến trúc microservice” là quá trình nghiên cứu và triển khai của riêng bản thân mình dưới sự hướng dẫn của Th.S Trần Hàm Dương. Đề tài được tiến hành một cách minh bạch, công khai. Những nhận định được trình bày trong khóa luận cũng là kết quả từ việc tự nghiên cứu trực tiếp, độc lập dựa vào các cơ sở tìm kiếm, hiểu biết tài liệu đã được công bố. Khóa luận sẽ giúp đảm bảo được tính khách quan, trung thực, đổi mới sáng tạo và khoa học.

Các số liệu, kết quả nghiên cứu được đưa ra trong khóa luận không sao chép hay sử dụng kết quả của bất kỳ đề tài nghiên cứu nào tương tự. Nếu như sai bản thân tôi xin chịu hoàn toàn trách nhiệm.

Tp. Hồ Chí Minh, ngày ... tháng ... năm 2024
Sinh viên

Vũ Nhật Chương

LỜI CẢM ƠN

Em xin được gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của trường Đại học Quốc Tế Sài Gòn nói chung, các thầy cô trong khoa Kỹ thuật & Khoa học máy tính nói riêng. Đặc biệt, để hoàn thành khóa luận tốt nghiệp này, em xin tỏ lòng biết ơn đến thầy Trần Hàm Dương, người đã tận tình hướng dẫn em trong suốt quá trình hoàn thành khóa luận tốt nghiệp để em có thể đạt kết quả tốt nhất.

Em cũng xin gửi lời cảm ơn đến các anh chị, thầy cô trong đơn vị thực tập đã tạo điều kiện cho em trang bị những kiến thức chuyên môn, có cơ hội được học hỏi, hiểu rõ hơn về cách tạo ra một sản phẩm hoàn chỉnh trong suốt 4 năm học tập và nghiên cứu. Từ đó, tạo cho em điều kiện thuận lợi và làm bước phát triển cho khóa luận tốt nghiệp trở nên chính chu hơn.

Em cũng vô cùng biết ơn, vì là một thành viên của Trung tâm Dữ liệu và Trí tuệ nhân tạo (SIU-AILAB), nơi nhận được sự quan tâm, hỗ trợ giúp đỡ của tất cả mọi người trong suốt thời gian làm việc và hoàn thiện luận văn.

Dù đã có cố gắng và nỗ lực trong quá trình thực hiện, nhưng một số kiến thức vẫn còn thiếu và trình độ chuyên môn cũng như kinh nghiệm thực tế của em vẫn chưa có nhiều nên sẽ không tránh khỏi những hạn chế và thiếu sót. Vì vậy, em kính mong nhận được sự góp ý và nhận xét của quý thầy, cô để từ đó em có thể hoàn thiện khóa luận tốt nghiệp của mình.

Em xin chân thành cảm ơn.

Tp. Hồ Chí Minh, ngày ... tháng ... năm 2024
Sinh viên

Vũ Nhật Chương

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày ... tháng ... năm 2024
GIẢNG VIÊN HƯỚNG DẪN

MỤC LỤC

LỜI CAM ĐOAN	i
LỜI CẢM ƠN	ii
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	iii
MỤC LỤC	iv
DANH MỤC HÌNH ẢNH	vi
Chương 1: TỔNG QUAN	1
1.1. Đặt vấn đề	1
1.1.1. Vấn đề hiện tại	1
1.1.2. Phương pháp giải quyết	2
1.2. Nhiệm vụ khóa luận	3
1.3. Phạm vi đề tài	4
1.4. Đối tượng sử dụng	5
1.5. Mục tiêu của ứng dụng	5
1.6. Các bước xây dựng khóa luận	6
1.7. Các công nghệ được sử dụng trong khoá luận	6
Chương 2: CƠ SỞ LÝ THUYẾT	7
2.1. Giới thiệu về phần mềm quản lý học vụ	7
2.2. Công nghệ và công cụ hỗ trợ trong khoá luận	8
2.2.1. Microservice	8
2.2.2. Hệ thống quản lý dịch vụ (Service management systems)	10
2.2.3. Giám sát (Observability)	13
2.2.4. Vì sao cần kết hợp microservice và giám sát	18
2.2.5. Công nghệ xây dựng Front-end: NextJS	18
2.2.6. Framework Back-end chính: Spring Boot	20
2.2.7. Cơ sở dữ liệu MySQL	22
2.2.8. Chạy dịch vụ phụ trợ với Docker	23
Chương 3: Xây dựng hệ thống microservice	24
3.1. Sơ đồ khái quát mô hình microservice	24
3.2. Consul	25
3.3. Config Server và Repository chứa config	25
3.3.1. Config first	26
3.3.2. Discovery first	26
3.3.3. Ứng dụng vào chương trình	26
3.4. Sơ đồ quản lý học vụ	28
3.5. Website quản lý đào tạo	35
3.5.1. Màn hình đăng nhập	35

3.5.2. Màn hình chức năng	36
Chương 4: Xây dựng hệ thống giám sát (Observability).....	39
4.1. Tổng quan về giám sát	39
4.2. Phần mềm trực quan Grafana Open Source Software.....	41
4.3. Loki	44
4.4. Mimir.....	45
4.5. Tempo	47
Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	49
5.1. Kết luận	49
5.2. Hướng phát triển	49
5.3. Tầm nhìn dài hạn.....	50
TÀI LIỆU THAM KHẢO	51

DANH MỤC HÌNH ẢNH

Hình 1: Ví dụ về microservice	9
Hình 2: Logo Consul	11
Hình 3: Ví dụ về Service Discovery	12
Hình 4: Ba cột mốc của giám sát trong microservice.....	14
Hình 5: Metrics trong microservice	15
Hình 6: Logs trong microservice.	16
Hình 7: Tracing trong microservice	17
Hình 8: Grafana Dashboard.....	18
Hình 9: Logo NextJS.....	19
Hình 10: Logo Spring Boot.....	21
Hình 11: Logo MySQL	22
Hình 12: Sơ đồ mô hình microservice	24
Hình 13: Ví dụ về Spring Cloud Config Server.	25
Hình 14: Chạy Consul.....	25
Hình 15: Ví dụ cài đặt cho Server Config	27
Hình 16: Server config cho config server.....	27
Hình 17: Đồ thị quản lý học vụ khái quát.....	28
Hình 18: Đồ thị quản lý sinh viên	29
Hình 19: Đồ thị quản lý đào tạo 1.....	30
Hình 20: Đồ thị quản lý đào tạo 2.....	31
Hình 21: Đồ thị quản lý đào tạo rút gọn	32
Hình 22: Model của dịch vụ quản lý lớp học.....	33
Hình 23: Dùng FeignClient để giao tiếp giữa với dịch vụ Subject.	34
Hình 24: Thêm một lớp học.	35
Hình 25: Màn hình đăng nhập.....	36
Hình 26: Màn hình chức năng.....	37
Hình 27: Màn hình chỉnh sửa thông tin chi tiết	38
Hình 28: Màn hình tạo mới chuyên ngành	38
Hình 29: Sơ đồ này minh họa cấu hình hệ thống theo dõi.	40
Hình 30: Cài đặt Grafana Agent cho dịch vụ.....	40
Hình 31: Cấu trúc folder observability	41
Hình 32: Dashboard Grafana hiện thông tin chung của chương trình	42
Hình 33: Dashboard Grafana hiện thông tin dịch vụ Classroom.....	43
Hình 34: Xem trace của ID 16a075... ..	43
Hình 35: Xem trace của dịch vụ classroom dưới dạng Node graph.....	44
Hình 36: Xem trace của dịch vụ classroom dưới dạng tường tự.....	44
Hình 37: Loki Logging stack	45
Hình 38: Docker chạy Loki.....	45
Hình 39: Cài đặt cho Mimir	46
Hình 40: Docker chạy Mimir	46
Hình 41: Trace	47
Hình 42: Cài đặt cho Tempo	48
Hình 43: Docker chạy Tempo.....	48

Chương 1: TỔNG QUAN

1.1. Đặt vấn đề

1.1.1. Vấn đề hiện tại

Trong bối cảnh môi trường giáo dục hiện tại, đặc biệt là tại trường Đại học Quốc tế Sài Gòn, việc quản lý hiệu quả các dữ liệu có cấu trúc từ nhiều nguồn khác nhau với mối quan hệ phức tạp như thông tin sinh viên, điểm số, học kỳ, lớp học và phòng học trở nên ngày càng khó khăn. Sử dụng các phương pháp truyền thống như giấy tờ và sổ sách không còn đáp ứng được yêu cầu ngày càng cao của việc quản lý dữ liệu. Điều này dẫn đến việc xử lý và quản lý dữ liệu trở nên ngày càng phức tạp và dễ xảy ra sai sót.

Với tỷ lệ sinh viên ngày càng tăng tại trường Đại học Quốc tế Sài Gòn, quá trình lưu trữ thông tin gặp nhiều thách thức, có thể dẫn đến các sai sót hoặc mất mát dữ liệu – điều này cần phải tránh bằng mọi giá. Khi số lượng dữ liệu tăng lên, việc quản lý trở nên đòi hỏi nhiều nhân lực hơn, dễ dẫn đến những sai sót không mong muốn. Do đó, trường cần có một giải pháp hiện đại, có khả năng hỗ trợ tốt việc lưu trữ và mở rộng dữ liệu một cách hiệu quả và bền vững.

Hiện nay, Đại học Quốc tế Sài Gòn, cùng nhiều trường khác, đang phụ thuộc vào phần mềm mã nguồn đóng được phát triển bởi công ty AQTech. Việc sử dụng phần mềm được thiết kế tổng thể cho nhiều trường học dẫn đến những khó khăn trong việc chỉnh sửa hay bổ sung các chức năng mới. Hơn nữa, việc mở rộng phần mềm ra các nền tảng như di động hoặc tích hợp với các ứng dụng khác trở nên rất khó khăn và tốn kém. Kiến trúc phần mềm phức tạp với nhiều nghiệp vụ, một số chức năng và luồng tích hợp không phù hợp với đặc điểm riêng của trường

Do sử dụng phần mềm từ bên thứ ba và không phải do trường tự phát triển, việc nắm bắt nhu cầu về tài nguyên triển khai trở nên khó khăn. Điều này ảnh hưởng đến khả năng lập

kế hoạch tối ưu hóa, tiết kiệm chi phí và mở rộng trong tương lai. Nếu xảy ra các sự cố vận hành hoặc lỗi kỹ thuật, trường phải liên hệ trực tiếp với AQTech, gây ra bất tiện và có thể làm gián đoạn hệ thống.

Trường Đại học Quốc tế Sài Gòn cần các hệ thống có kiến trúc linh hoạt, dễ dàng mở rộng và có khả năng phát triển theo tốc độ tăng trưởng của trường. Hệ thống cần có khả năng thêm và thay đổi các chức năng một cách nhanh chóng, đáp ứng nhu cầu về cả chức năng và công nghệ. Điều này sẽ giúp trường có thể quản lý dữ liệu hiệu quả, giảm thiểu sai sót và nâng cao chất lượng dịch vụ giáo dục.

1.1.2. Phương pháp giải quyết

Để giải quyết vấn đề quản lý dữ liệu một cách hiện đại và đáp ứng tiêu chuẩn hiện nay, trường Đại học Quốc tế Sài Gòn cần triển khai một hệ thống phần mềm mã nguồn mở do chính nhà trường kiểm soát. Hệ thống này sẽ hỗ trợ quản lý đào tạo thông qua cơ sở dữ liệu tập trung, giúp lưu trữ thông tin một cách thuận lợi và dễ dàng truy vấn. Bên cạnh đó, việc phát triển một website đơn giản, dễ sử dụng, cho phép mọi người trong trường truy cập mọi lúc mọi nơi để tra cứu thông tin, là một yếu tố quan trọng.

Để đáp ứng nhu cầu ngày càng phát triển của nhà trường, cơ sở hạ tầng phải được xây dựng dựa trên kiến trúc hỗ trợ quá trình mở rộng phần mềm một cách linh hoạt, với tính an toàn cao và khả năng quản lý chặt chẽ. Kiến trúc phù hợp nhất hiện nay và được nhiều công ty lớn cấp quốc gia ưa chuộng là kiến trúc Microservice.

Kiến trúc Microservice giúp nhà trường dễ dàng nâng cấp và mở rộng các chức năng mà không ảnh hưởng đến toàn bộ hệ thống. Mỗi dịch vụ trong kiến trúc này được triển khai độc lập, giúp tăng cường tính linh hoạt và khả năng mở rộng. Điều này đặc biệt quan trọng trong việc quản lý dữ liệu lớn và phức tạp của trường, đảm bảo tính liên tục và hiệu quả của các dịch vụ cung cấp.

Mặc dù kiến trúc Microservice mang lại nhiều lợi ích, nó không phải là một kiến trúc hoàn hảo. Để đảm bảo tính ổn định, an toàn và liên mạch của tất cả các dịch vụ, Microservice cần sự hỗ trợ của các phần mềm quản lý hệ thống bên thứ ba và các công cụ giám sát, kiểm soát lỗi. Những công cụ này giúp nhà trường nắm bắt được tình hình hệ thống và nhanh chóng khắc phục các sự cố kỹ thuật. Do đó, việc sử dụng các framework giám sát và điều phối là cần thiết để duy trì sự ổn định và hiệu quả của hệ thống.

Việc triển khai một hệ thống phần mềm mã nguồn mở kết hợp với kiến trúc Microservice và các công cụ giám sát, kiểm soát lỗi sẽ giúp trường Đại học Quốc tế Sài Gòn nâng cao hiệu quả quản lý dữ liệu. Điều này không chỉ đảm bảo tính linh hoạt và khả năng mở rộng mà còn đáp ứng các yêu cầu khắt khe về an toàn và quản lý chặt chẽ. Hệ thống này sẽ là nền tảng vững chắc cho sự phát triển bền vững và hiệu quả của nhà trường trong tương lai.

1.2. Nhiệm vụ khóa luận

Dựa trên các vấn đề đã nêu, khóa luận này tập trung vào việc xây dựng một website để quản lý đào tạo, sử dụng kiến trúc Microservice kết hợp với các công cụ giám sát và quản lý hệ thống. Mục tiêu là hỗ trợ nhà trường quản lý, lưu trữ thông tin một cách dễ dàng, thuận lợi, và đáp ứng nhu cầu ngày càng phát triển.

Khóa luận sẽ tập trung vào việc áp dụng kiến trúc Microservice, một phương pháp hiện đại và linh hoạt, vào hệ thống quản lý đào tạo. Điều này bao gồm việc triển khai các dịch vụ độc lập, cho phép dễ dàng nâng cấp và mở rộng các chức năng mà không ảnh hưởng đến toàn bộ hệ thống. Kiến trúc Microservice sẽ được triển khai kèm với các công cụ hỗ trợ quản lý và điều phối, nhằm đảm bảo tính ổn định và hiệu quả của hệ thống.

Một trong những trọng tâm của khóa luận là việc giám sát và quản lý các dịch vụ một cách hiệu quả. Điều này bao gồm việc sử dụng các framework và công cụ giám sát để theo dõi tình trạng hoạt động của hệ thống, phát hiện và xử lý kịp thời các sự cố kỹ thuật.

Các công cụ này sẽ giúp đảm bảo tính ổn định và liên tục của hệ thống, giảm thiểu nguy cơ gián đoạn và tăng cường tính an toàn.

Khóa luận sẽ mang lại một giải pháp quản lý đào tạo hiệu quả, giúp nhà trường dễ dàng quản lý và lưu trữ thông tin. Hệ thống này không chỉ đáp ứng nhu cầu hiện tại mà còn có khả năng mở rộng và phát triển trong tương lai. Việc áp dụng kiến trúc Microservice cùng với các công cụ giám sát và quản lý hệ thống sẽ giúp đảm bảo tính linh hoạt, an toàn và hiệu quả của hệ thống, đáp ứng đầy đủ các yêu cầu của một môi trường giáo dục hiện đại.

Khóa luận này không chỉ cung cấp một giải pháp kỹ thuật cho việc quản lý đào tạo mà còn đóng góp vào việc nghiên cứu và ứng dụng kiến trúc Microservice trong lĩnh vực giáo dục. Các kết quả và kinh nghiệm từ khóa luận sẽ là tài liệu tham khảo quý giá cho các nghiên cứu và triển khai hệ thống quản lý đào tạo trong tương lai.

Tóm lại, khóa luận nhằm xây dựng một hệ thống quản lý đào tạo hiện đại, sử dụng kiến trúc Microservice kết hợp với các công cụ giám sát và quản lý hệ thống, đáp ứng nhu cầu phát triển của nhà trường và đảm bảo tính linh hoạt, an toàn và hiệu quả. Đây là bước quan trọng trong việc nâng cao chất lượng quản lý và dịch vụ giáo dục tại trường Đại học Quốc tế Sài Gòn.

1.3. Phạm vi đề tài

Đề tài tập trung vào việc xây dựng một hệ thống gồm các dịch vụ tách rời thông qua kiến trúc microservice có tích hợp những phần mềm quản lý các dịch vụ đó nhằm đáp ứng nhu cầu mở rộng của trường Đại học Quốc tế Sài Gòn. Bên cạnh đó là một website đơn giản cho nhân viên đào tạo, sinh viên sử dụng.

- Phạm vi chức năng: Phần mềm quản lý đào tạo gồm các chức năng:
 - Quản lý cơ bản các dữ liệu như thêm, xóa, sửa và truy vấn thông qua giao diện web đơn giản.

- Phân chia các bộ phận chức năng thành từng dịch vụ đơn nhỏ theo microservice.
- Quản lý và giám sát các dịch vụ đó bằng cách chương trình bên thứ ba.

Đơn vị sử dụng: Trường đại học Quốc tế Sài Gòn (SIU) nói chung cho cả hệ thống microservice và các nhân viên phòng đào tạo, sinh viên cho website quản lý đào tạo. Nhưng cách thiết kế hệ thống này về mặt chung có thể áp dụng cho các trường khác hay thậm chí những công ty khác.

1.4. Đối tượng sử dụng

- Nhân viên phòng đào tạo: Người chịu trách nhiệm lên kế hoạch giảng dạy, sắp xếp lớp cho các sinh viên.
- Giảng viên: Người sử dụng chương trình để xem danh sách học sinh trong lớp, những thông tin về môn học.
- Sinh viên: Người dùng để xem điểm, các thông tin như phòng học, hệ đào tạo.
- Quản trị viên, nhà phát triển: Sử dụng màn hình trực quan của giám sát để thu thập, phân tích thông tin về nhật ký, lỗi và đo lường.

1.5. Mục tiêu của ứng dụng

- Xây dựng một cơ sở hạ tầng có kiến trúc linh hoạt, dễ dàng thay đổi hay nâng cấp bất kì lúc nào.
- Tích hợp hệ thống giám sát tốt, có thể thu thập và ghi chú đầy đủ các đo lường, nhật ký mà các dịch vụ xuất xuyên suốt.
- Sử dụng hệ thống quản lý dịch vụ để kiểm soát các dịch vụ.
- Website quản lý có giao diện và chức năng đơn giản, dễ sử dụng.
- Thiết kế được cấu trúc cơ sở dữ liệu phù hợp đạt nhu cầu.
- Giúp cải thiện cơ sở hạ tầng của trường.

1.6. Các bước xây dựng khóa luận

- Khảo sát các chức năng chính nhà trường cần có.
- Phân tích và thiết kế hệ thống.
- Xây dựng các dịch vụ microservice theo chức năng một cách hợp lý
- Tìm hiểu và cài đặt công nghệ quản lý dịch vụ.
- Tìm hiểu và cài đặt công nghệ giám sát microservice.
- Thu thập dữ liệu được quan sát.

1.7. Các công nghệ được sử dụng trong khoá luận

- Framework Front-end: NextJS để xây dựng giao diện web cơ bản để hình dung và thao tác.
- Framework Back-end: Spring Boot để xây dựng các dịch vụ API cho kiến trúc microservice.
- Cơ sở dữ liệu: MySQL để lưu trữ, truy vấn và quản lý dữ liệu.
- Hệ thống quản lý dịch vụ: Consul để quản lý các dịch vụ trong microservice.
- Stack cho giám sát (Observability): Loki, Grafana, Tempo, Mimir, Grafana Agent giúp dựng lên hệ thống giám sát phù hợp cho microservice.
- Chạy dịch vụ phụ trợ: Docker dùng để chạy các dịch vụ hỗ trợ microservice như Consul, stack giám sát.

Chương 2: CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu về phần mềm quản lý học vụ

Phần mềm quản lý học vụ là một hệ thống được thiết kế nhằm tạo ra một cổng thông tin tập trung, hỗ trợ quản lý hiệu quả từng chức năng của học vụ như quản lý số lượng lớp học, các hệ đào tạo, và nhiều khía cạnh liên quan khác. Hệ thống này đóng vai trò quan trọng trong việc tối ưu hóa quy trình quản lý thông tin học vụ của nhà trường.

Nhu cầu xây dựng một phần mềm quản lý học vụ xuất phát từ những khó khăn trong việc lưu trữ và quản lý thông tin học vụ theo phương pháp truyền thống. Các phương pháp này thường không tối ưu, đặc biệt khi trường học ngày càng phát triển. Điều này dẫn đến việc tìm kiếm và truy vấn dữ liệu trở nên tốn nhân lực và có khả năng xảy ra sai sót cao.

Trong môi trường giáo dục ngày càng phát triển, việc quản lý thông tin học vụ như danh sách lớp học, môn học, và các hệ đào tạo theo cách truyền thống gặp nhiều thách thức. Sự gia tăng về số lượng sinh viên và khối lượng thông tin cần quản lý làm cho việc sử dụng giấy tờ và sổ sách trở nên không còn hiệu quả. Các phương pháp này không chỉ làm tốn thời gian và công sức của nhân viên quản lý, mà còn có nguy cơ dẫn đến những sai sót trong quá trình lưu trữ và truy xuất dữ liệu.

Phần mềm quản lý học vụ được phát triển nhằm giải quyết các vấn đề trên bằng cách cung cấp một giải pháp công nghệ hiện đại. Hệ thống này sẽ giúp nhà trường quản lý thông tin học vụ một cách tập trung, giảm thiểu công sức và thời gian cần thiết cho việc xử lý thông tin. Đồng thời, phần mềm cũng giúp tăng cường tính chính xác và hiệu quả trong quá trình quản lý, đáp ứng nhu cầu ngày càng tăng của nhà trường.

Tóm lại, việc xây dựng và triển khai phần mềm quản lý học vụ là một bước quan trọng trong việc hiện đại hóa quy trình quản lý thông tin học vụ tại các trường học. Hệ thống

này không chỉ giúp tối ưu hóa quy trình quản lý, mà còn đảm bảo tính chính xác và hiệu quả, đáp ứng tốt nhu cầu phát triển của môi trường giáo dục hiện đại.

2.2. Công nghệ và công cụ hỗ trợ trong khoá luận

2.2.1. Microservice

2.2.1.1. Định Nghĩa

Kiến trúc Microservice là một phương pháp tiếp cận mang tính kiến trúc và tổ chức để phát triển phần mềm, trong đó phần mềm được chia thành các dịch vụ nhỏ, độc lập. Mỗi dịch vụ thực hiện một chức năng cụ thể và giao tiếp với các dịch vụ khác thông qua các API được xác định rõ ràng. Các dịch vụ này thường được sở hữu và quản lý bởi các nhóm nhỏ, độc lập, có quyền tự chủ trong việc phát triển, triển khai và vận hành dịch vụ của mình. Kiến trúc Microservice cho phép các nhóm phát triển phần mềm hoạt động một cách linh hoạt và nhanh chóng, đáp ứng tốt các yêu cầu thay đổi và mở rộng hệ thống.

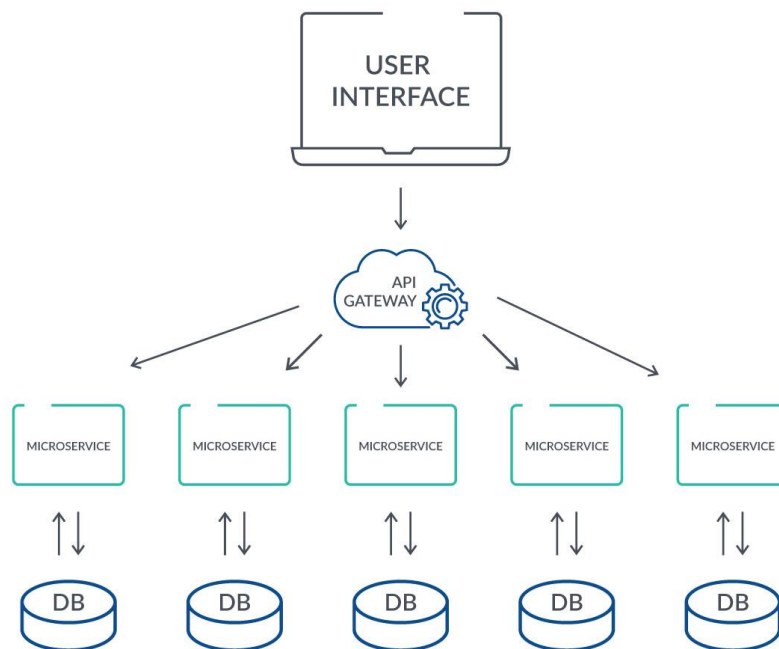
2.2.1.2. Vì sao phải dùng microservice

Việc sử dụng kiến trúc Microservice mang lại nhiều lợi ích đáng kể trong quá trình phát triển và vận hành hệ thống phần mềm:

- **Khả năng nhân bản dễ dàng:** Một trong những điểm mạnh lớn nhất của Microservice là khả năng nhân bản các dịch vụ một cách dễ dàng. Mỗi dịch vụ có thể được triển khai trên một máy chủ riêng (có thể sử dụng các dịch vụ đám mây như AWS hoặc Azure), hoặc nhiều dịch vụ có thể được nhân đôi và đặt trên một máy chủ chung. Điều này giúp hệ thống dễ dàng mở rộng và phân bổ tải công việc một cách hiệu quả.
- **Quản lý và sửa lỗi hiệu quả:** Kiến trúc Microservice giúp việc tìm và sửa lỗi trở nên dễ dàng hơn, bởi các dịch vụ được tách nhỏ và độc lập. Khi xảy ra sự cố, việc xác

định và khắc phục lỗi chỉ ảnh hưởng đến dịch vụ cụ thể, không gây gián đoạn cho toàn bộ hệ thống.

- **Độc lập về công nghệ:** Mỗi dịch vụ trong kiến trúc Microservice có thể được phát triển bằng các ngôn ngữ lập trình khác nhau và sử dụng cơ sở dữ liệu riêng. Điều này cho phép các nhóm phát triển lựa chọn công nghệ phù hợp nhất với yêu cầu của từng dịch vụ, tối ưu hóa hiệu suất và khả năng mở rộng.
- **Giao tiếp qua API:** Các dịch vụ giao tiếp với nhau thông qua các API được xác định rõ ràng, giúp đảm bảo tính nhất quán và dễ dàng tích hợp giữa các dịch vụ. Việc thống nhất cấu trúc dữ liệu truyền đi và nhận về là yếu tố quan trọng trong việc đảm bảo tính tương thích và hoạt động trơn tru của hệ thống.
- **Khả năng mở rộng và phát triển linh hoạt:** Kiến trúc Microservice cho phép dễ dàng mở rộng và thêm mới các dịch vụ mà không ảnh hưởng đến toàn bộ hệ thống. Điều này đặc biệt quan trọng trong môi trường phát triển phần mềm hiện đại, nơi yêu cầu thay đổi và mở rộng liên tục.



Hình 1: Ví dụ về microservice

2.2.2. Hệ thống quản lý dịch vụ (Service management systems)

2.2.2.1. Hệ thống quản lý dịch vụ là gì?

Hệ thống quản lý dịch vụ trong kiến trúc microservice là công cụ cấp cao giúp theo dõi, quản lý, và điều phối các dịch vụ trong microservice. Loại công cụ này thường có chức năng hỗ trợ nhận dạng và giao tiếp lẫn nhau giữa các dịch vụ trong nội bộ môi trường microservice.

Đôi khi, các công cụ này còn được gọi là hệ thống quản lý dịch vụ (service management systems) hoặc nền tảng điều phối dịch vụ (service orchestration platforms). Chúng thuộc về một lĩnh vực rộng hơn gọi là quản lý cơ sở hạ tầng (infrastructure management).

2.2.2.2. Vì sao cần hệ thống quản lý dịch vụ?

Hệ thống quản lý dịch vụ giúp tự động hóa quá trình khám phá và đăng ký dịch vụ (Service Discovery), cho phép các thành phần trong hệ thống dễ dàng tìm và định dạng nhau để kết nối với nhau mà không cần cấu hình thủ công phức tạp hay biết rõ thông tin chi tiết địa chỉ của dịch vụ, ta không cần phải ghi chú lại dịch vụ nào nằm ở địa chỉ nào, chỉ cần sử dụng tên ta đặt ra là chúng có thể giao tiếp với nhau.

Công cụ này còn hỗ trợ giám sát sức khỏe của các dịch vụ, đảm bảo rằng chỉ những dịch vụ hoạt động tốt mới được sử dụng, giúp tăng độ tin cậy và ổn định của hệ thống. Nó còn cung cấp khả năng quản lý cấu hình phân tán, cho phép thay đổi cấu hình một cách linh hoạt và đồng bộ mà không cần khởi động lại hệ thống.

Cuối cùng, chúng cải thiện bảo mật bằng cách kiểm soát và phân đoạn dịch vụ, đảm bảo rằng chỉ những dịch vụ được phép mới có thể giao tiếp với nhau. Những tính năng này không chỉ giúp giảm thiểu sai sót và tăng hiệu quả vận hành mà còn tối ưu hóa việc quản lý và mở rộng hệ thống, đáp ứng nhanh chóng nhu cầu thay đổi và mở rộng của doanh nghiệp.

Trong thị trường hiện nay, hai sản phẩm nổi tiếng cho việc này là Eureka tạo bởi Netflix và Consul do CNCF (Cloud Native Computing Foundation).

2.2.2.3. Quản lý dịch vụ bằng Consul

Consul là một giải pháp quản lý dịch vụ mạnh mẽ và linh hoạt cho các hệ thống phân tán và microservice. Được phát triển bởi HashiCorp, Consul cung cấp một bộ công cụ phong phú gồm service discovery và service configuration.

Với Consul, các dịch vụ có thể dễ dàng tìm và kết nối với nhau thông qua một hệ thống đăng ký và tìm kiếm tự động, giúp giảm thiểu sự phức tạp trong việc quản lý kết nối giữa các dịch vụ.

Ngoài ra, Consul còn hỗ trợ giám sát sức khỏe của các dịch vụ (health checking), đảm bảo rằng chỉ những dịch vụ hoạt động tốt mới được đưa vào sử dụng. Tính năng phân đoạn dịch vụ của Consul cũng tăng cường bảo mật bằng cách kiểm soát truy cập giữa các dịch vụ. Với những khả năng nêu trên, khả năng mở rộng và tích hợp tốt với nhiều công cụ khác, Consul trở thành một lựa chọn hàng đầu cho các hệ thống microservices hiện đại.



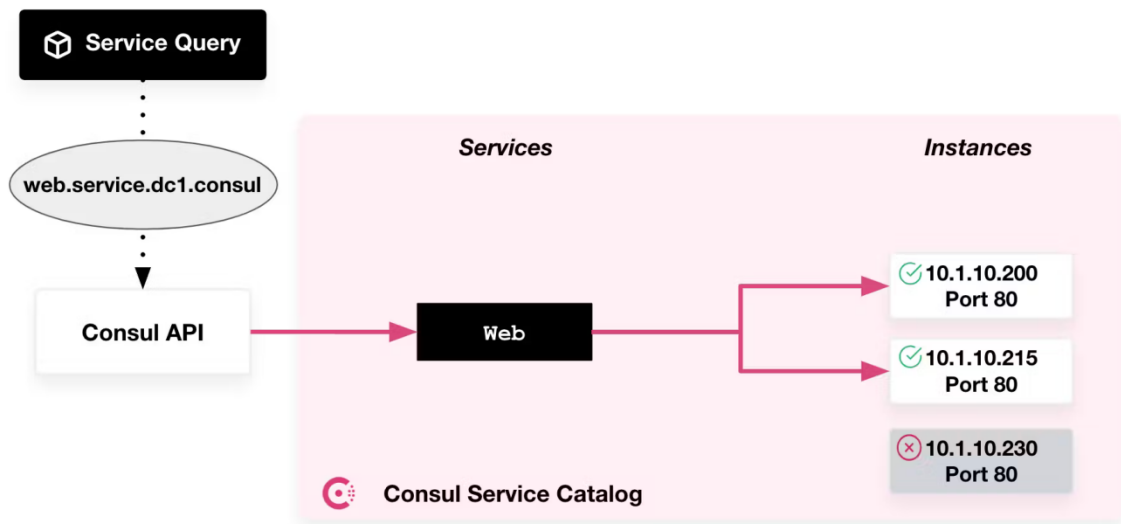
Hình 2: Logo Consul

2.2.2.4. Khám phá dịch vụ (Service Discovery)

Service discovery là quá trình các dịch vụ tự động tìm và kết nối với nhau mà không cần phải cấu hình thủ công. Khi một dịch vụ mới được bắt đầu, nó sẽ tự đăng ký thông tin của mình như địa chỉ IP và Port vào Consul (service registry). Các dịch vụ khác có thể truy vấn Consul để tìm thông tin về các dịch vụ mà chúng cần giao tiếp thông qua những thông tin dễ nhớ và nhận dạng hơn như tên hoặc nhãn.

Điều này giúp đảm bảo rằng các dịch vụ luôn có thể kết nối với nhau ngay cả khi có sự thay đổi về địa chỉ hoặc cấu hình, vì các dịch vụ giao tiếp với nhau thông qua tên thay vì IP và Port, thêm vào đó ta có thể nhân bản các dịch vụ và đặt nó ở một nơi khác, Consul sẽ tự động điều chỉnh giao thông để ta có thể sử dụng hai dịch vụ như một.

Service discovery giúp tăng cường tính linh hoạt, khả năng mở rộng và độ tin cậy của hệ thống bằng cách tự động hóa quá trình quản lý kết nối giữa các dịch vụ, giảm thiểu lỗi và đảm bảo tính nhất quán trong giao tiếp.



Hình 3: Ví dụ về Service Discovery

2.2.2.5. Cấu hình dịch vụ (Service Configuration)

Service configuration là dịch vụ lưu trữ, quản lý và duy trì các cấu hình và cài đặt để các dịch vụ có thể dễ dàng lấy và sử dụng.