

BAN CƠ YẾU CHÍNH PHỦ
HỌC VIỆN KỸ THUẬT MẬT MÃ



ĐỀ CƯƠNG CHUYÊN ĐỀ AN TOÀN HỆ THỐNG THÔNG TIN
Nghiên cứu giải pháp đảm bảo an toàn cho Microservices trên Kubernetes

Ngành: An toàn thông tin

Sinh viên thực hiện:

Phương Văn Sơn

Mã sinh viên: AT160258

Lê Huy Dũng

Mã sinh viên: AT160211

Người hướng dẫn:

KS. Nguyễn Mạnh Thắng

Khoa An toàn thông tin - Học viện Kỹ thuật mật mã

Hà Nội, 2022

Mục lục

Lời cảm ơn	3
Lời mở đầu	4
1 Giới thiệu về công nghệ Container và kiến trúc Microservices	5
1.1 Giới thiệu về công nghệ Container	5
1.2 Giới thiệu về kiến trúc Microservices	5
1.3 Giới thiệu về Kubernetes	5
1.4 Một số vấn đề bảo mật Microservices trên Kubernetes	5
2 Tổng quan về Istio	6
2.1 Tổng quan về Istio	6
2.1.1 Tổng quan về Service Mesh	6
2.1.2 Kiến trúc của Istio	6
2.1.3 Tổng quan về Envoy Proxy	6
2.2 Quản lý mạng giữa các Microservices với Istio	6
2.2.1 Tổng quan về Istio Ingress Gateway	6
2.2.2 Định tuyến trong Istio	6
2.2.3 Giải quyết các vấn đề về mạng trong Microservices	6
2.3 Giám sát các Microservices với Istio	6
2.3.1 Một số Metrics quan trọng của Istio	6
2.3.2 Giám sát các lưu lượng mạng qua Jaeger và Kiali	6
2.4 Bảo mật các Microservices bằng Istio	6
2.4.1 Xác thực giữa các Microservices với Istio	6
2.4.2 Phân quyền cho các Microservices với Istio	6
3 Triển khai Istio trên Kubernetes	7
3.1 Mô hình triển khai	7
3.2 Kịch bản triển khai	7
3.3 Thực nghiệm	7
3.4 Kết luận	7

Lời cảm ơn

Nhóm chúng em xin chân thành cảm ơn các thầy cô trường Học viện Kỹ thuật Mật mã nói chung, quý thầy cô của khoa An toàn thông tin nói riêng đã tận tình dạy bảo, truyền đạt kiến thức cho chúng em trong suốt quá trình học.

Kính gửi đến Thầy Nguyễn Mạnh Thắng lời cảm ơn chân thành và sâu sắc nhất, cảm ơn thầy đã tận tình theo sát, chỉ bảo và hướng dẫn cho nhóm em trong quá trình thực hiện đề tài này. Thầy không chỉ hướng dẫn chúng em những kiến thức chuyên ngành, mà còn giúp chúng em học thêm những kĩ năng mềm, tinh thần học hỏi, thái độ khi làm việc nhóm.

Trong quá trình tìm hiểu nhóm chúng em xin cảm ơn các bạn sinh viên đã góp ý, giúp đỡ và hỗ trợ nhóm em rất nhiều trong quá trình tìm hiểu và làm đề tài.

Do kiến thức còn nhiều hạn chế nên không thể tránh khỏi những thiếu sót trong quá trình làm đề tài. Chúng em rất mong nhận được sự đóng góp ý kiến của quý thầy cô để đề tài của chúng em đạt được kết quả tốt hơn.

Chúng em xin chân thành cảm ơn!

Lời mở đầu

Nhiều năm trước, hầu hết các ứng dụng phần mềm đều được xây dựng với kiến trúc monolith hay còn gọi là kiến trúc 1 khối là mẫu thiết kế được dùng nhiều nhất trong giới lập trình web hiện nay bởi tính đơn giản của nó khi phát triển và khi triển khai. Các ứng dụng này chạy dưới dạng một tiến trình đơn lẻ hoặc số lượng nhỏ các tiến trình trên một số ít máy chủ. Chúng có khả năng cập nhật và nâng cấp chậm và yêu cầu nâng cấp thường xuyên. Trong trường hợp có sự cố như lỗi phần cứng hệ thống phần mềm này sẽ phải được di chuyển một cách thủ công sang các máy chủ còn hoạt động tốt.

Ngày nay các ứng dụng được xây dựng với kiến trúc lớn và phức tạp đang dần được chia thành các thành phần nhỏ hơn, có khả năng hoạt động độc lập được gọi là microservices. Vì các Microservices tách biệt với nhau nên chúng có thể được phát triển, triển khai hay cập nhật và mở rộng quy mô một cách riêng lẻ. Nhờ khả năng này cho phép ta thay đổi các thành phần nhanh chóng và thường xuyên khi cần thiết để theo kịp với các yêu cầu thay đổi nhanh chóng thời nay.

Nhưng với số lượng lớn các thành phần cũng như cơ sở dữ liệu việc cấu hình, quản lý và giữ hệ thống hoạt động trơn tru ngày càng trở nên khó khăn đặc biệt trong việc tối ưu hiệu quả sử dụng tài nguyên. Kubernetes ra đời để đáp ứng nhu cầu tự động hoá như lập lịch tự động, cấu hình tự động hay giám sát và xử lý lỗi.

Kubernetes cung cấp cho các nhà phát triển khả năng triển khai các ứng dụng một cách thường xuyên mà không cần thông qua nhóm vận hành. Không chỉ dừng lại ở đó Kubernetes cũng giúp nhóm vận hành tự động theo dõi và khắc phục sự cố.

Đi cùng với sự phát triển lớn mạnh của kiến trúc Microservices cũng như Kubernetes đó là nhu cầu về việc đảm bảo tính an toàn cho các hệ thống này. Trong bài báo cáo này chúng em sẽ giới thiệu về giải pháp đảm bảo an toàn cho Microservices bằng Istio Service Mesh

Chương 1

Giới thiệu về công nghệ Container và kiến trúc Microservices

1.1 Giới thiệu về công nghệ Container

1.2 Giới thiệu về kiến trúc Microservices

1.3 Giới thiệu về Kubernetes

1.4 Một số vấn đề bảo mật Microservices trên Kubernetes

Kết luận Chương 1

Chương 2

Tổng quan về Istio

2.1 Tổng quan về Istio

2.1.1 Tổng quan về Service Mesh

2.1.2 Kiến trúc của Istio

2.1.3 Tổng quan về Envoy Proxy

2.2 Quản lý mạng giữa các Microservices với Istio

2.2.1 Tổng quan về Istio Ingress Gateway

2.2.2 Định tuyến trong Istio

2.2.3 Giải quyết các vấn đề về mạng trong Microservices

2.3 Giám sát các Microservices với Istio

2.3.1 Một số Metrics quan trọng của Istio

2.3.2 Giám sát các lưu lượng mạng qua Jaeger và Kiali

2.4 Bảo mật các Microservices bằng Istio

2.4.1 Xác thực giữa các Microservices với Istio

2.4.2 Phân quyền cho các Microservices với Istio

Kết luận chương 2

Chương 3

Triển khai Istio trên Kubernetes

3.1 Mô hình triển khai

3.2 Kịch bản triển khai

3.3 Thực nghiệm

3.4 Kết luận