



**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN MÔN HỌC
HỆ TÍNH TOÁN PHÂN BỐ NÂNG CAO**

**TÊN ĐỀ TÀI: AMAZON WEB SERVICES: KIẾN TRÚC VÀ
ỨNG DỤNG**

**TÊN ĐỀ TÀI TIẾNG ANH: AMAZON WEB SERVICES:
ARCHITECTURES AND APPLICATIONS DEVELOPMENT**

**GIẢNG VIÊN HƯỚNG DẪN
TS. LÊ DUY TÂN**

**HỌC VIÊN THỰC HIỆN
TÔ TRỌNG NGHĨA
NGUYỄN HỒNG SƠN
TẠ VIỆT HOÀNG**

1 Tóm tắt đề tài

Trong hàng thập kỷ phát triển của ngành Công nghệ Thông tin, máy tính luôn là một trong những điểm khởi đầu cũng như nút thắt cuối cùng trong sự phát triển của công nghệ. Rào cản về điện tử và vật lý một mặt thúc đẩy, mặt khác cũng kìm hãm những giới hạn tư duy của các nhà khoa học.

Từ khởi đầu với kích thước lớn như tòa nhà, đến thời điểm vi xử lý chỉ còn vài centimet, từ những trung tâm dữ liệu được xây dựng đầu tiên, tới việc mỗi công ty, tập đoàn, trường đại học đều có một trung tâm dữ liệu riêng, với những cách quản trị khác nhau. Những phát triển nhanh chóng về cơ sở hạ tầng tiếp tục thúc đẩy những nghiên cứu mới vượt qua mọi rào cản kỹ thuật, ứng dụng được nhiều thuật toán, giải pháp trên nền tảng máy tính mới, với tốc độ xử lý, khả năng tổng hợp kho dữ liệu lớn cùng đường truyền tốc độ cao. Thông qua internet, người dùng kết nối thiết bị của mình đến các thiết bị ảo hóa có sức mạnh xử lý khổng lồ ở bất kỳ nơi nào trên thế giới [1].

Hệ thống của các nhà cung cấp điện toán đám mây phân bố khắp thế giới, sử dụng nhiều thuật toán, chương trình, thiết bị, vật liệu và hàng ngàn kỹ sư để duy trì kết nối với nhau, cung cấp dịch vụ liên tục cho chính hệ thống của họ và thương mại hóa.

Điện toán đám mây là sự tổng hợp của Distributed Computing, Parallel Computing, Utility Computing along with Network Storage, Virtualization, Load Balance, High Available và các công nghệ liên quan khác. Có nhiều định nghĩa khác nhau về Điện toán đám mây, một trong số định nghĩa phổ biến thường được sử dụng đó là: “Điện toán đám mây là một mô hình phổ biến, thuận tiện, cho phép truy cập từ mạng cục bộ tới các tài nguyên tính toán có thể tùy chỉnh cấu hình, được cung cấp và có thể sử dụng nhanh chóng với những tác động tối thiểu từ nhà cung cấp dịch vụ” [2].

Trong khuôn khổ đề án này, nhóm học viên giới thiệu một số giải pháp được cung cấp của dịch vụ Amazon Web Services, trình bày khả năng của nó trong việc đảm bảo cung cấp dịch vụ liên tục một cách phân tán, hệ thống được cung cấp cho người dùng mà họ không hay biết nó được triển khai trên các thành phần khác nhau.

Phần còn lại của báo cáo được trình bày như sau:

- Phần hai trình bày lịch sử hình thành và một số khái niệm chính của điện toán đám mây nói chung cũng như của AWS nói riêng.
- Phần ba trình bày chi tiết các dịch vụ được sử dụng trong báo cáo, các loại hình dịch vụ, những điểm mạnh và các phần còn hạn chế.
- Phần bốn trình bày mô hình, ngữ cảnh ứng dụng. Các thành phần hệ thống, phương pháp triển khai và biểu diễn kết quả
- Phần năm trình bày kết luận và các nội dung liên quan.

2 Giới thiệu

2.1 Điện toán đám mây

2.2 Amazon Web Services

2.3 Các nghiên cứu và nhà cung cấp chủ yếu

3 Một số dịch vụ sử dụng

3.1 Amazon RDS

3.2 Route 53

3.3 Amazon Elastic Compute Cloud

4 Hiện thực ứng dụng

4.1 Yêu cầu chung của bài toán

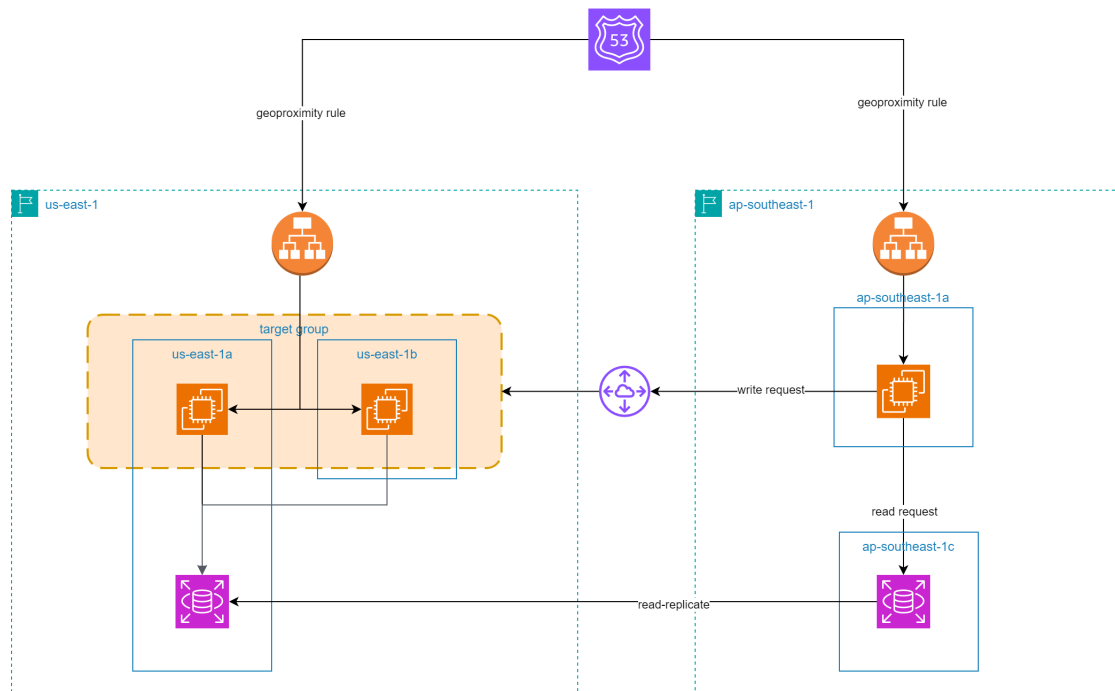
Hệ thống được triển khai là một hệ thống đơn giản, chỉ bao gồm backend và database. Người dùng sử dụng một domain duy nhất truy cập vào ứng dụng. Tùy vị trí của người dùng, ứng dụng được điều hướng đến các server khác nhau. Ứng dụng có sử dụng cân bằng tải. Database có khả năng chịu lỗi cao, thời gian đọc nhanh.

4.2 Phân tích thiết kế hệ thống

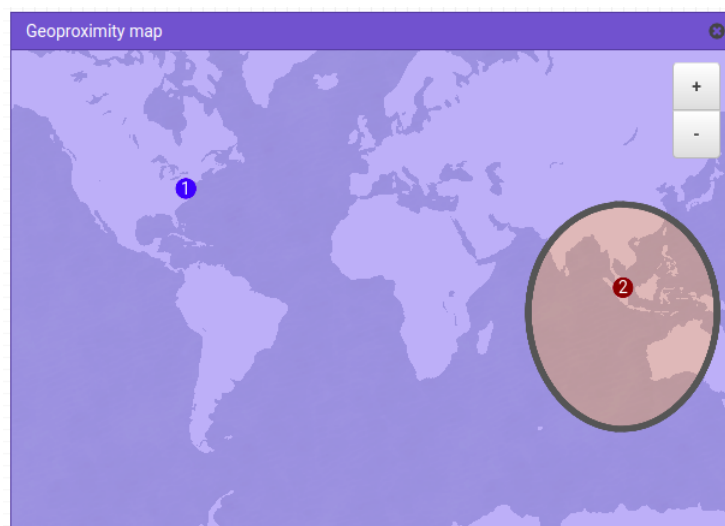
Với những yêu cầu như đề ra tại 4.1, ta nhận định:

- Ứng dụng cần một DNS có khả năng phân giải theo khu vực địa lý. Route53 hỗ trợ tính năng này
- Ứng dụng có cân bằng tải, Application Load Balancer phù hợp cho trường hợp này. Các loại load balancer khác như network load balancer cũng có thể đảm nhiệm cùng tình năng. Nhưng ở đây ta cần cân bằng tải ở tầng ứng dụng, việc sử dụng tới layer 3 là không cần thiết.
- Database trên AWS hầu hết đều cung cấp chức năng replicate. Vì vậy, ta chọn tùy theo tech stack. Ở đây chọn RDS MySQL.

Những lựa chọn trên không hẳn là tốt nhất, nhưng được cân nhắc sao cho vừa thể hiện tính phân tán của hệ thống, vừa sử dụng những tài nguyên ít chi phí, đồng thời có khả năng "trình diễn" sâu. Việc sử dụng các dịch vụ server less cũng được cân



Hình 4.1: Sơ đồ tổng quan ứng dụng



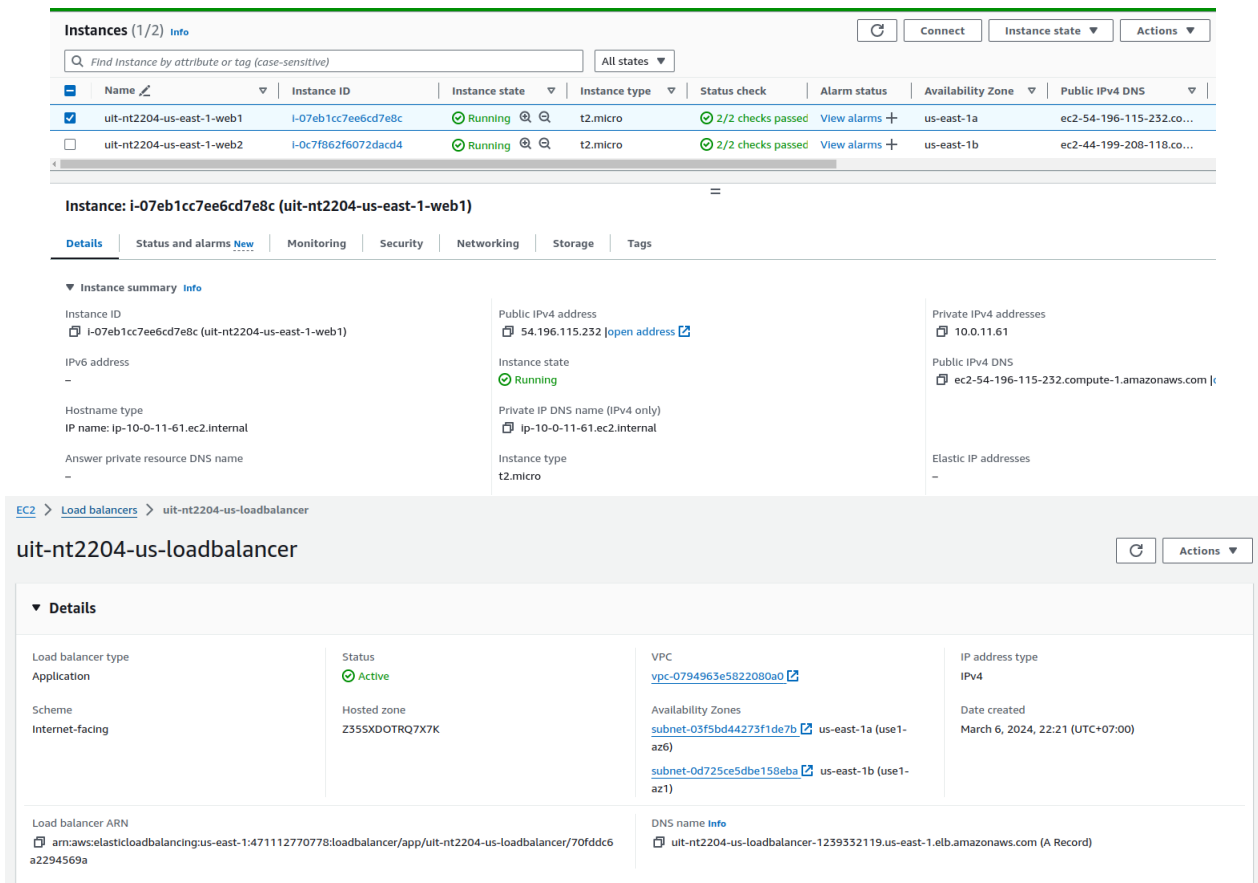
Hình 4.2: Mô tả geoproximity rule của Route 53

nhắc, tuy nhiên sẽ trùng đề tài. Khả năng dựng database trên các instance cũng có thể được thực hiện trong trường hợp production để giảm chi phí.

Tổng quan hệ thống trình bày trong hình 4.1.

Tại hình ảnh trên, người dùng sử dụng ứng dụng có domain name được quản lý bởi Route 53. Sử dụng geoproximity policy để điều hướng request tới từng region khác nhau. Nếu địa chỉ IP của người dùng nằm trong vùng màu xanh (số 1), các request sẽ được điều hướng tới region *us-east-1*, ngược lại các request sẽ được xử lý tại region *ap-southeast-1* (xem hình 4.2).

Tại backend, deploy web service nginx và một service Flask app đơn giản. Ứng dụng này trả về các truy xuất vào các database khi được yêu cầu hoặc trả về hostname



Hình 4.3: Các tài nguyên tạo tại region US

của server và timestamp ở mặc định. Trước khi request đi vào backend, sử dụng một Application Load Balancer nhằm cân bằng tải. Nó thực hiện heal check tới backend thông qua port 80 và serve port 80 ra ngoài. Từ bên ngoài không gọi thẳng được tới backend thông qua giao thức http.

Tại database, tiến hành deploy một RDS MySQL chính tại *us-east-1* và một RDS Read Replicate tại *ap-southeast-1* (do chi phí ở *us-east-1* rẻ hơn). AWS hỗ trợ chúng ta đồng bộ dữ liệu cho 2 database này với cam kết cao. Các request yêu cầu database tại *us-east-1* sẽ được xử lý trên RDS Master. Trong khi các request đọc database ở *ap-southeast-1* sẽ được đọc trên RDS Slave. Các yêu cầu ghi database tại *ap-southeast-1* được chuyển tiếp tới backend ở *us-east-1* thông qua một private connection (VPC Peering connection)

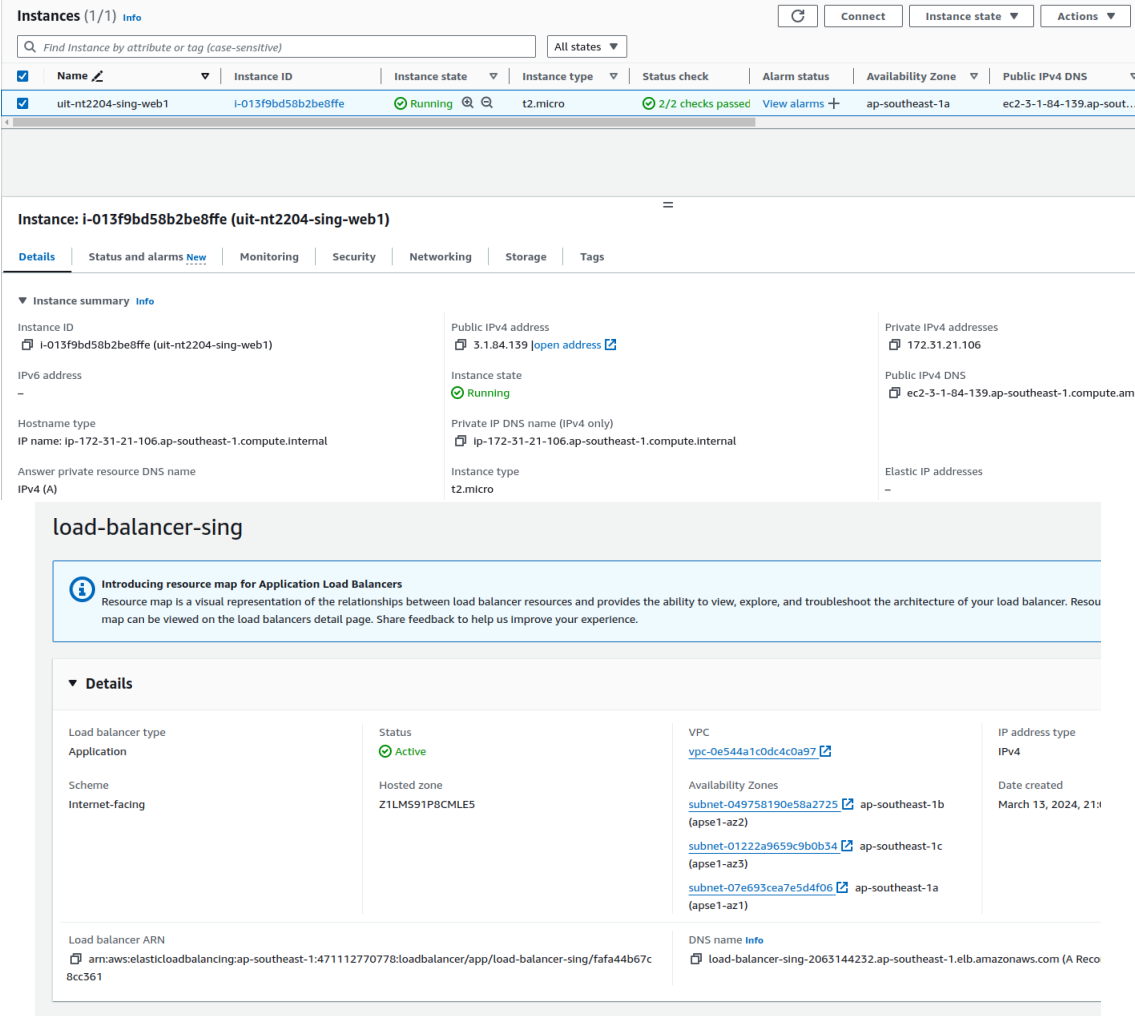
4.3 Hiện thực ứng dụng

Chúng ta tiến hành tạo các tài nguyên như đã trình bày trong 4.2.

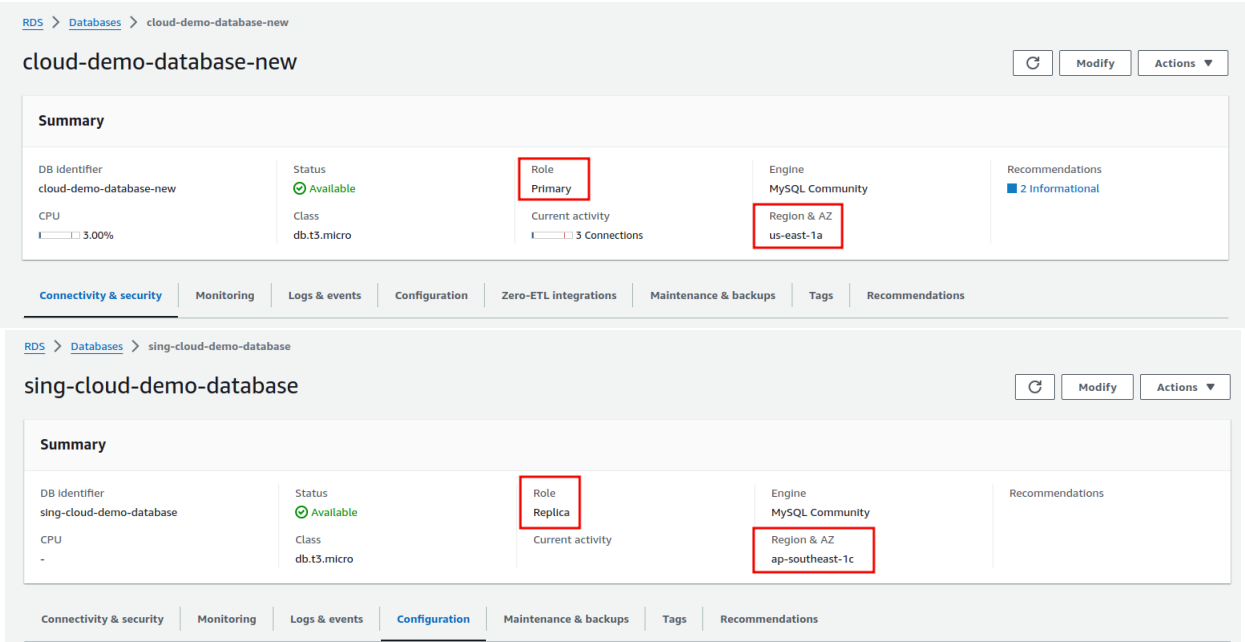
Tại region *us-east-1*, tiến hành tạo các tài nguyên EC2 và Application Load Balancer.

Tương tự trong hình 4.4 là các tài nguyên ở region Sing.

Đối với database, database có sự khác biệt giữa Primary và Replica (hình 4.5)



Hình 4.4: Các tài nguyên tạo tại region Sing



Hình 4.5: Database tại hai region US và Sing

Records (1/9) [Info](#)

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

Type ▼

Routing pol... ▼

Alias ▼

< 1 >

<input type="checkbox"/>	nt2204.sonnh.net	CNAME	Geoproxi...	ap-southeast-1	No	load-balancer-sing-2063144...	300	-	-	sing-loadbalancer
<input type="checkbox"/>	nt2204.sonnh.net	CNAME	Geoproxi...	us-east-1	No	ult-nt2204-us-loadbalancer...	300	-	-	us-loadbalancer
<input type="checkbox"/>	sing-lab.sonnh.net	CNAME	Simple	-	No	load-balancer-sing-2063144...	60	-	-	-
<input type="checkbox"/>	us-lab.sonnh.net	CNAME	Simple	-	No	ult-nt2204-us-loadbalancer...	300	-	-	-

Hình 4.6: Cấu hình các record trong Route53

Đối với Route53, ta cấu hình domain *nt2004.sonnh.net* theo rule Geoproximity như trình bày tại 4.2. Ngoài ra, để thuận tiện testing, cấu hình thêm các domain *sing-lab.sonnh.net* và *us-lab.sonnh.net* như hình 4.6.

4.4 Kết quả ứng dụng trong từng ngữ cảnh

Kết quả, hình 4.7 thể hiện, khi cùng truy cập một domain, với vị trí tại Việt Nam thì ứng dụng của chúng ta được xử lý bởi backend đặt tại Sing, còn khi chuyển vùng bằng VPN sang US thì lại được xử lý bởi backend đặt tại US.

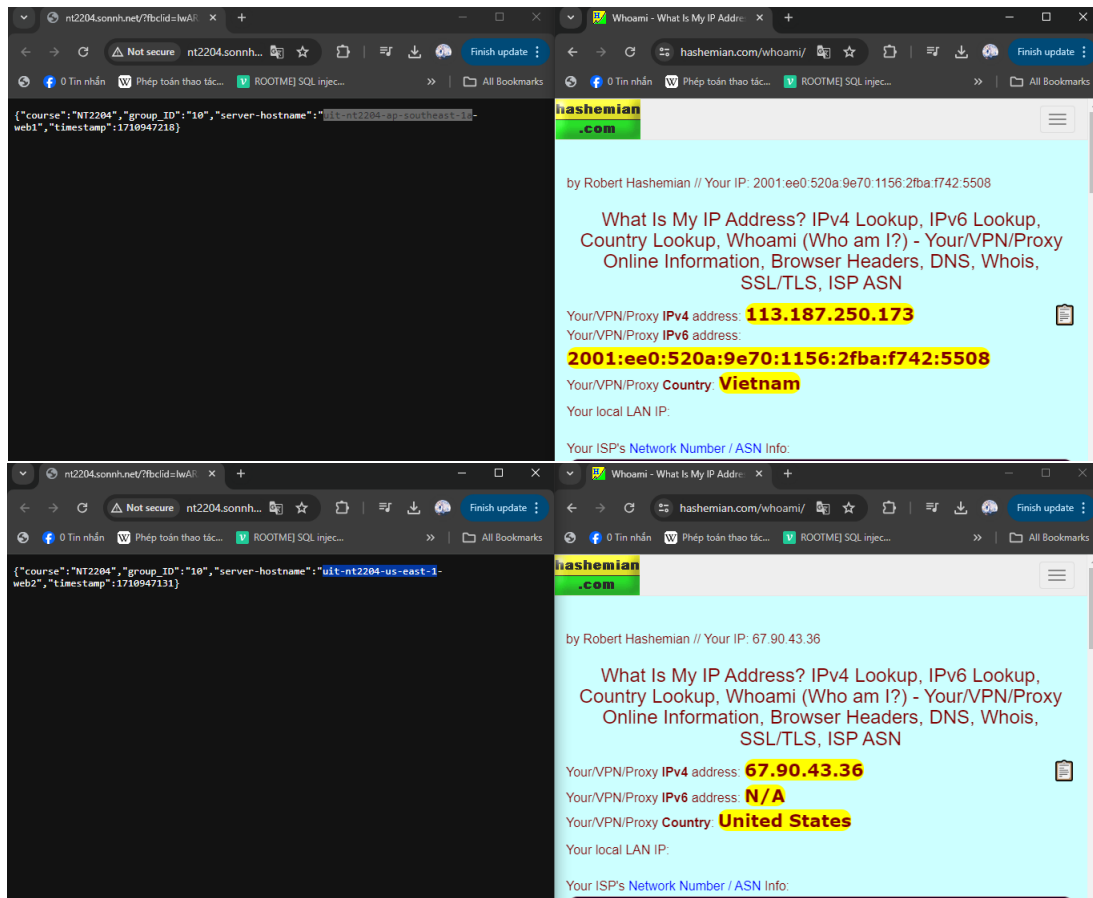
Tại app domain *sing-lab.sonnh.net*, ta request hai loại thao tác với database khác nhau (action=create_table và action=select_records). Kết quả khi request tạo bảng, là yêu cầu cần ghi, nó được xử lý bởi backend đặt tại US (do forward request sang), ngược lại thì được xử lý ở Sing (xem hình 4.8)

Một ví dụ nữa, ta sẽ thêm các records vào database với domain sing, và sau đó query từ domain us. Hình 4.9 cho thấy, record insert vào database được xử lý tại US, nhưng bởi server web2, trong khi request lấy dữ liệu lại được xử lý bởi server web1. Có sự xuất hiện này là do ta đã sử dụng Application Load Balancer.

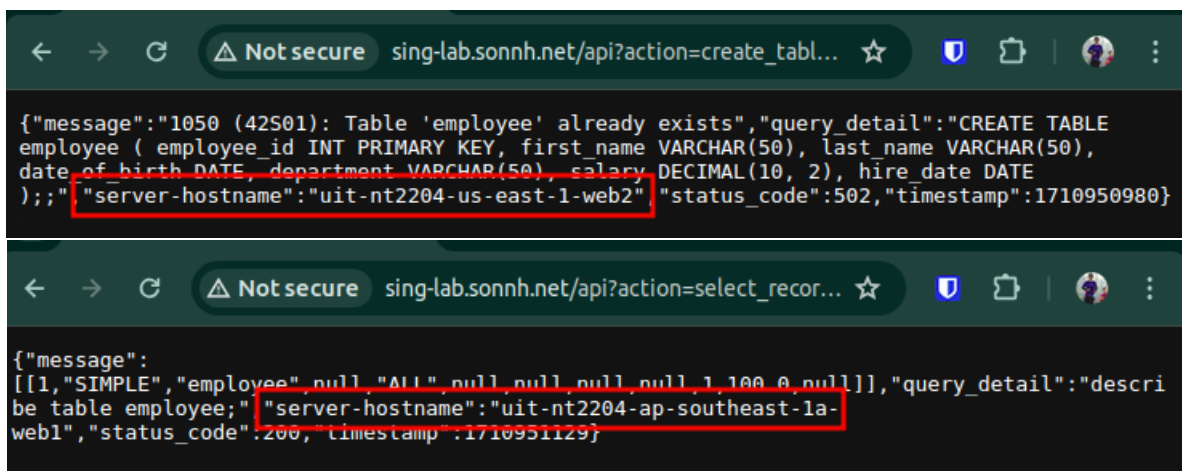
5 Kết luận đề tài

Như vậy, thông qua đề tài, nhóm đã ứng dụng một số dịch vụ được cung cấp bởi AWS để thể hiện sự phân tán của ứng dụng trên môi trường cloud. Bài báo cáo đã:

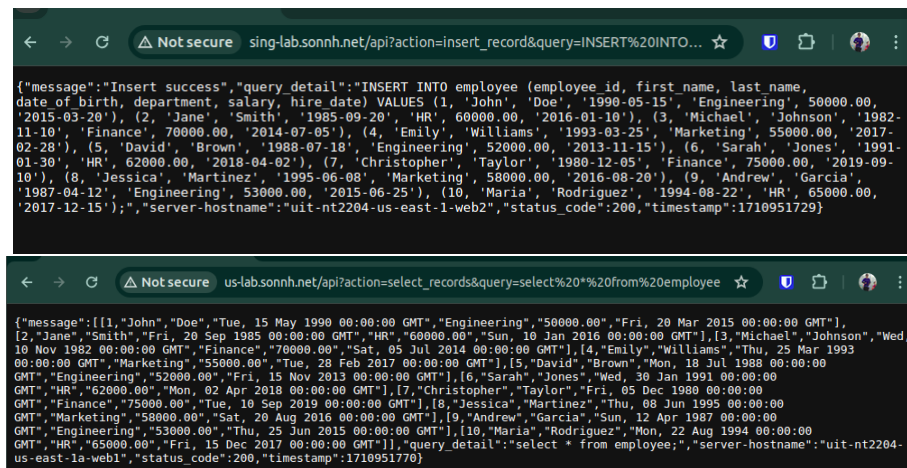
- Giới thiệu một số dịch vụ của AWS
- Áp dụng các dịch vụ vào việc triển khai một hệ thống phân tán



Hình 4.7: Truy cập cùng một domain ở các vị trí khác nhau



Hình 4.8: Các yêu cầu khác nhau cho app domain tại Sing



Hình 4.9: Thực hiện tạo records và sau đó query với hai domain khác nhau

Mặc dù vậy, do giới hạn về tài nguyên cũng như thời gian, nhóm chưa trình bày ứng dụng dưới dạng UI để nhìn mà chỉ hiển thị ở API response. Đây cũng là hướng phát triển ứng dụng trong giai đoạn tiếp theo.

Tài liệu tham khảo

- [1] Borivoje Furht, Armando Escalante, et al. *Handbook of cloud computing*. Vol. 3. Springer, 2010.
- [2] Peter Mell, Tim Grance, et al. “The NIST definition of cloud computing”. In: (2011).