



**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

Khoa học và công nghệ đã và đang có những bước phát triển vĩ đại trong vài thập kỷ vừa qua. Các công trình nghiên cứu bước ra khỏi phòng thí nghiệm và phục vụ đại chúng, trong đó có các ứng dụng cung cấp khả năng tính toán vượt bậc, giá thành rẻ, dễ sử dụng được gọi là điện toán đám mây. 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 [**furht2010handbook**].

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.

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

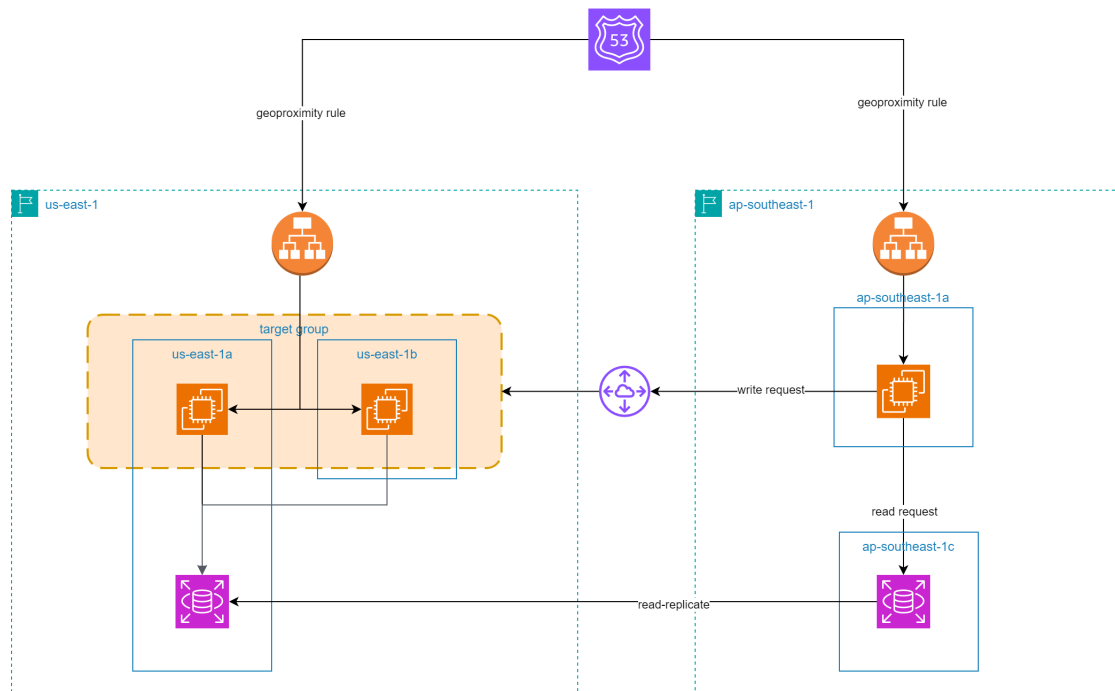
Hệ thống được triển khai là một hệ thống đơn giản, chỉ bao gồm backend và database. Người dùng sử dụng một domain duy nhất truy cập vào ứng dụng. Tùy vị trí của người dùng, ứng dụng được điều hướng đến các server khác nhau. Ứng dụng có sử dụng cân bằng tải. Database có khả năng chịu lỗi cao, thời gian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

4.2 Phân tích thiết kế hệ thống

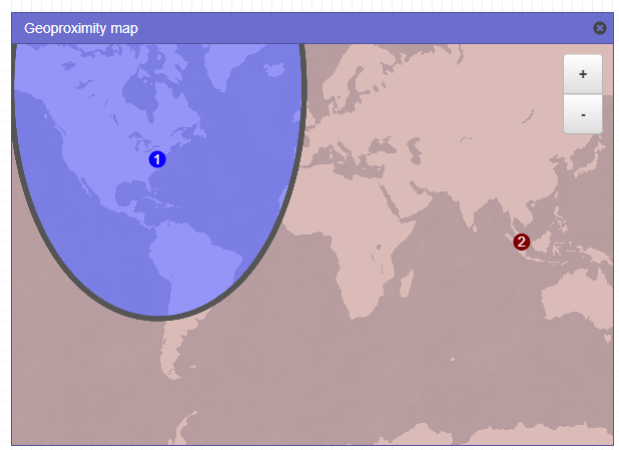
Với những yêu cầu như đề ra tại 4.1, ta nhận định:

- 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 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

4.4 Kết quả ứng dụng trong từng ngữ cảnh

5 Kết luận đề tài