BDRPC187**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO CUỐI KỲ**

**MÔN HỌC: ĐIỆN TOÁN ĐÁM MÂY**

*ĐỀ TÀI*

**XÂY DỰNG VPC VÀ CHẠY WEBSERVER TRÊN AWS**

**

**GVHD:** TS. Huỳnh Xuân Phụng

**(**Nhóm 4 – STT 27 – Lớp 04)

**Sinh viên thực hiện:**

Nguyễn Viết Quang - 20110549

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP.HCM KHOA CNTT**  \*\*\*\*\*\*\* | **XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh Phúc**  \*\*\*\*\*\*\* |

# **PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên : Nguyễn Viết Quang MSSV: 20110549

Ngành: Công nghệ thông tin

Tên đề tài: Xây dựng VPC và chạy webserver trên AWS

Họ và tên Giáo viên hướng dẫn: TS. Huỳnh Xuân Phụng

**NHẬN XÉT**

Đề tài :

..................................................................................................................................................

..................................................................................................................................................

..................................................................................................................................................

..................................................................................................................................................

1. Ưu điểm:

..................................................................................................................................................

.................................................................................................................................................

..................................................................................................................................................

..................................................................................................................................................

1. Khuyết điểm

..................................................................................................................................................

..................................................................................................................................................

..................................................................................................................................................

..................................................................................................................................................

1. Kiến nghị về đề tài
2. Xếp loại:
3. Điểm:

Tp*. Hồ Chí Minh, ngày tháng năm 2021*

Giáo viên hướng dẫn

*(Ký & ghi rõ họ tên)*

# 

# **MỤC LỤC**

[**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN** 1](#_Toc120998415)

[**MỤC LỤC** 2](#_Toc120998416)

[**LỜI CẢM ƠN** 4](#_Toc120998417)

[**PHÂN CÔNG** 5](#_Toc120998418)

[**PHẦN 1: MỞ ĐẦU** 6](#_Toc120998419)

[**PHẦN 2: NỘI DUNG** 7](#_Toc120998420)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 7](#_Toc120998421)

[**1.1.** **Amazon Virtual Private Cloud (VPC)** 7](#_Toc120998422)

[**1.2.** **Amazon DynamoDB** 8](#_Toc120998423)

[**1.3.** **Amazon Elastic Compute Cloud (EC2)** 10](#_Toc120998424)

[**1.4.** **Amazon Simple Storage Service (S3)** 12](#_Toc120998425)

[CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT 13](#_Toc120998426)

[**2.1.** **Phân tích yêu cầu** 13](#_Toc120998427)

[**2.1.1.** **Chức năng của website** 13](#_Toc120998428)

[**2.1.2.** **Thiết kế Cơ sở dữ liệu NoSQL** 13](#_Toc120998429)

[**2.2.** **Cài đặt chương trình** 14](#_Toc120998430)

[**2.2.1.** **Xây dựng Website Quản lý thông tin sinh viên** 14](#_Toc120998431)

[**2.2.2. Cấu trúc thư mục của website 14**](#_Toc120998432)

[**2.2.3. Giao diện Website 15**](#_Toc120998433)

[**2.2.4.** **Xây dựng dịch vụ tương tác với Amazon DynamoDB** 20](#_Toc120998434)

[**2.2.5.** **Xây dựng dịch vụ tương tác với Amazon S3** 28](#_Toc120998435)

[**2.2.** **Chạy project trên AWS** 29](#_Toc120998436)

[**2.3.1.** **Xây dựng VPC** 29](#_Toc120998437)

[**2.3.2.** **Tạo EC2** **instance** 32](#_Toc120998438)

[**2.3.3.** **Tạo bucket S3** 34](#_Toc120998439)

[**2.3.4.** **Tạo table DynamoDB và dữ liệu ban đầu** 34](#_Toc120998440)

[**2.3.5.** **Deploy WAR file** 35](#_Toc120998441)

[**2.3.6.** **Kiểm tra website** 35](#_Toc120998442)

[CHƯƠNG 3: KẾT LUẬN 36](#_Toc120998443)

[TÀI LIỆU THAM KHẢO 37](#_Toc120998444)

# **LỜI CẢM ƠN**

Trong thời gian 15 tuần học tập và nghiên cứu môn Điện toán đám mây tại Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh, tuy không dài nhưng em đã có cơ hội vận dụng những kiến thức đã được học trong nhà trường kết hợp với những kiến thức thu nhận được dưới sự chỉ bảo tận tình của giảng viên hướng dẫn để đứng trên bục giảng với vai trò là một người thầy, người truyền đạt lại kiến thức cho học sinh. Quá trình học tập đã giúp em có được một số kinh nghiệm để hoàn thành bài báo cáo lần này. Chân thành cảm đến Khoa Công Nghệ Thông Tin – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo điều kiện cho em học tập và nghiên cứu

Lời cảm ơn sâu sắc em gửi đến thầy Th.s Huỳnh Xuân Phụng - Giảng viên bộ môn Điện Toán Đám Mây đã sắp xếp thời gian quý báu của mình để hướng dẫn em để em có thể hoàn thành tốt bài báo cáo vầ đề tai **Xây dựng VPC và chạy webserver trên AWS**.

Chúng em xin cảm ơn thầy vì thời gian qua đã hướng dẫn và chỉ dạy chúng em nhiệt tình giúp chúng em nắm được những kiến thức bổ ích của môn học này.

Mặc dù đã có nhiều cố gắng nhưng đây vẫn là một kiến thức khá mới đối với em vậy nên cũng khó tránh khỏi những sai sót nên em mong thaanhf thông cảm và hướng dẫn em khác phục vào nhứng lần sau ạ

Chúng em xin chân thành cảm ơn!

**PHÂN CÔNG**

**Phân công thực hiện xây dựng website quản lý thông tin sinh viên**

**Link github:** https://github.com/TangSan003/Project-Cuoi-Ki-of-Cloud.git

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Công việc** |
| Nguyễn Viết Quang | 20110549 | * Xây dựng VPC: 1 public subnet + 1 private subnet * Khởi tạo EC2 (cài đặt máy chủ web) và DynamoDB * Trang web quản lý thông tin sinh viên (Giao diện + Xử lý + Cơ sở dữ liệu đơn giản) |

**PHẦN 1: MỞ ĐẦU**

1. **Tính cấp thiết của đề tài**

Cơ sở dữ liệu (database) nói chung và cơ sở dữ liệu không quan hệ (noSQL database) nói riêng là một thành phần không thể thiếu trong việc tạo nên các phần mềm ứng dụng hoặc website có như cầu lưu trữ thông tin có thể hoặc không do người dùng nhập vào. Nhờ chúng mà dữ liệu với những insight quan trọng có thể được lưu trữ một cách dễ dàng. Vì vậy mà việc tạo nên một hệ thống có khả năng làm cho việc tương tác đến cơ sở dữ liệu của các lập trình viên trở nên dễ dàng hơn là một việc cấp thiết cần phải được thực hiện.

1. **Đối tượng nghiên cứu**

Đối với đề tài này, đối tượng nghiên cứu là VPC, DynamoDB, EC2. Đồng thời kết hợp với các kiến thức về Front-end và Back-end để tạo một trang web Quản lý thông tin sinh viên.

1. **Phạm vi nghiên cứu**

Đề tài này chủ yếu tập trung vào việc kết hợp sử dụng VPC và EC2 nhằm thực hiện với chạy webserver, đồng thời sử dụng DynamoDB để làm Cơ sở dữ liệu

1. **Kết quả dự kiến đạt được**

- Cài đặt giao diện website Quản lý thông tin sinh viên

- Cài đặt phía backend

- Xây dựng VPC với 2 subnet: private và public

- Xây dựng máy ảo EC2 nằm trong VPC đã tạo để host website

- Thêm, sửa, xóa các item và các table trong DynamoDB

**PHẦN 2: NỘI DUNG**

**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

**Link slide:** https://www.canva.com/design/DAFUWVAUAZs/r-qAYDmQVxq5RVDJbLV0fw/edit

* 1. **Amazon Virtual Private Cloud (VPC)**
     1. **Tổng quan:**

Amazon Virtual Private Cloud (VPC) là một dịch vụ điện toán đám mây thương mại cung cấp cho người dùng một đám mây riêng ảo, bằng cách “cung cấp một phần bị cô lập về mặt logic của Amazon Web Services (AWS) Cloud”. Amazon Virtual Private Cloud (Amazon VPC) cho phép bạn khởi chạy các tài nguyên AWS vào một mạng ảo mà bạn đã xác định. Mạng ảo này gần giống với mạng truyền thống mà bạn vận hành trong trung tâm dữ liệu của riêng mình, với những lợi ích của việc sử dụng cơ sở hạ tầng có thể mở rộng của AWS.

Là một trong các dịch vụ nền tảng của AWS, Amazon VPC sẽ giúp bạn dễ dàng tùy chỉnh cấu hình mạng của VPC. Bạn có thể tạo một mạng con công khai cho các máy chủ web có quyền truy cập internet. Dịch vụ này cũng cho phép bạn đặt các hệ thống backend, như máy chủ ứng dụng hoặc cơ sở dữ liệu, trong mạng con riêng tư không có quyền truy cập internet. Với Amazon VPC, bạn có thể sử dụng nhiều lớp bảo mật, bao gồm các nhóm bảo mật và danh sách kiểm soát truy cập mạng, để giúp kiểm soát quyền truy cập vào các phiên bản Amazon EC2 trong mỗi mạng con

* + 1. **Các tính năng**

Các tính năng dùng để cấu hình VPC

* **VPC**: VPC là một mạng ảo gần giống với mạng truyền thống mà bạn vận hành trong trung tâm dữ liệu của riêng mình. Sau khi tạo VPC, bạn có thể thêm mạng con.
* **Subnets**: Mạng con là một dải địa chỉ IP trong VPC của bạn. Một mạng con phải nằm trong một Vùng sẵn sàng duy nhất. Sau khi thêm mạng con, bạn có thể triển khai tài nguyên AWS trong VPC của mình.
* **Địa chỉ IP**: Bạn có thể gán địa chỉ IPv4 và địa chỉ IPv6 cho VPC và mạng con của mình. Bạn cũng có thể đưa địa chỉ IPv4 và IPv6 GUA công khai của mình tới AWS và phân bổ chúng cho các tài nguyên trong VPC của bạn, chẳng hạn như phiên bản EC2, cổng NAT và Bộ cân bằng tải mạng.
* **Routing**: Sử dụng bảng định tuyến để xác định nơi lưu lượng truy cập mạng từ mạng con hoặc cổng của bạn được hướng đến.
* **Gatewates và endpoints**: Một cổng kết nối VPC của bạn với một mạng khác. Ví dụ: sử dụng cổng internet để kết nối VPC của bạn với internet. Sử dụng điểm cuối VPC để kết nối với các dịch vụ AWS một cách riêng tư mà không cần sử dụng cổng internet hoặc thiết bị NAT
* **Peering connections:** Sử dụng kết nối ngang hàng VPC để định tuyến lưu lượng giữa các tài nguyên trong hai VPC.
* **Traffic Mirroring:** Sao chép lưu lượng mạng từ các giao diện mạng và gửi nó đến các thiết bị giám sát và bảo mật để kiểm tra gói sâu.
* **Transit gateways:** Sử dụng một cổng chuyển tuyến, hoạt động như một trung tâm trung tâm, để định tuyến lưu lượng giữa các VPC, kết nối VPN và kết nối AWS Direct Connect của bạn
* **VPC Flow Logs:** Nhật ký luồng ghi lại thông tin về lưu lượng IP đi và đến từ các giao diện mạng trong VPC của bạn.
* **Kết nối VPN:** Kết nối VPC của bạn với mạng tại chỗ bằng Mạng riêng ảo AWS (AWS VPN).
  + 1. **Làm việc với Amazon VPC**

Bạn có thể tạo và quản lý VPC của mình bằng bất kỳ giao diện nào sau đây:

* **AWS Management Console**: Cung cấp giao diện web mà bạn có thể sử dụng để truy cập các VPC của mình.
* **AWS Command Line Interface (AWS CLI)**: Cung cấp các lệnh cho một loạt các dịch vụ AWS, bao gồm cả Amazon VPC và được hỗ trợ trên Windows, Mac và Linux. Để biết thêm thông tin, hãy xem Giao diện dòng lệnh AWS.
* **AWS SDKs**: Cung cấp API dành riêng cho ngôn ngữ và xử lý nhiều chi tiết kết nối, chẳng hạn như tính toán chữ ký, xử lý yêu cầu thử lại và xử lý lỗi. Để biết thêm thông tin, hãy xem AWS SDK.
* **Truy vấn API:** Cung cấp các hành động API cấp thấp mà bạn gọi bằng các yêu cầu HTTPS. Sử dụng API truy vấn là cách trực tiếp nhất để truy cập Amazon VPC, nhưng cách này yêu cầu ứng dụng của bạn xử lý các chi tiết cấp thấp như tạo hàm băm để ký yêu cầu và xử lý lỗi. Để biết thêm thông tin, hãy xem các hành động của Amazon VPC trong Tham chiếu API Amazon EC2.
  1. **Amazon DynamoDB**
     1. **Tổng quan**

Amazon DynamoDB là một dịch vụ cơ sở dữ liệu NoSQL được quản lý toàn phần, cung cấp hiệu suất nhanh và có thể dự đoán được với khả năng mở rộng liền mạch. DynamoDB cho phép bạn giảm bớt gánh nặng quản trị trong việc vận hành và thay đổi quy mô cơ sở dữ liệu phân tán để bạn không phải lo lắng về việc cung cấp, thiết lập và cấu hình phần cứng, sao chép, vá lỗi phần mềm hoặc thay đổi quy mô cụm. DynamoDB cũng cung cấp tính năng mã hóa khi lưu trữ, giúp loại bỏ gánh nặng vận hành và sự phức tạp liên quan đến việc bảo vệ dữ liệu nhạy cảm. Để biết thêm thông tin, hãy xem Mã hóa DynamoDB ở trạng thái nghỉ.

Với DynamoDB, bạn có thể tạo các bảng cơ sở dữ liệu có thể lưu trữ và truy xuất bất kỳ lượng dữ liệu nào cũng như phục vụ mọi cấp độ lưu lượng truy cập yêu cầu. Bạn có thể tăng hoặc giảm quy mô công suất thông lượng của bảng mà không có thời gian ngừng hoạt động hoặc suy giảm hiệu suất. Bạn có thể sử dụng Bảng điều khiển quản lý AWS để giám sát việc sử dụng tài nguyên và chỉ số hiệu suất.

DynamoDB cung cấp khả năng sao lưu theo yêu cầu. Nó cho phép bạn tạo các bản sao lưu đầy đủ cho các bảng của mình để lưu giữ và lưu trữ lâu dài cho các nhu cầu tuân thủ quy định. Để biết thêm thông tin, hãy xem Sử dụng sao lưu và khôi phục theo yêu cầu cho DynamoDB.

Bạn có thể tạo các bản sao lưu theo yêu cầu và bật tính năng khôi phục tại thời điểm cho các bảng Amazon DynamoDB của mình. Khôi phục tại thời điểm giúp bảo vệ các bảng của bạn khỏi các thao tác ghi hoặc xóa vô tình. Với khôi phục tại thời điểm, bạn có thể khôi phục bảng về bất kỳ thời điểm nào trong 35 ngày

DynamoDB cho phép bạn tự động xóa các mục đã hết hạn khỏi bảng để giúp bạn giảm mức sử dụng bộ nhớ và chi phí lưu trữ dữ liệu không còn phù hợp.

* + 1. **Tính khả dụng**

DynamoDB tự động phân bổ dữ liệu và lưu lượng truy cập cho các bảng của bạn qua một số lượng máy chủ đủ để xử lý các yêu cầu lưu trữ và thông lượng của bạn, đồng thời duy trì hiệu suất ổn định và nhanh chóng. Tất cả dữ liệu của bạn được lưu trữ trên ổ cứng thể rắn (SSD) và được tự động sao chép trên nhiều Vùng sẵn sàng trong Khu vực AWS, mang lại độ sẵn sàng cao và độ bền dữ liệu tích hợp sẵn. Bạn có thể sử dụng các bảng toàn cầu để giữ cho các bảng DynamoDB được đồng bộ hóa trên các Khu vực AWS.

* 1. **Amazon Elastic Compute Cloud (EC2)**
     1. **Tổng quan**

Amazon Elastic Compute Cloud (Amazon EC2) cung cấp khả năng điện toán có thể thay đổi quy mô trong Đám mây Amazon Web Services (AWS). Sử dụng Amazon EC2 giúp bạn không phải đầu tư trước vào phần cứng, nhờ đó bạn có thể phát triển và triển khai ứng dụng nhanh hơn. Bạn có thể sử dụng Amazon EC2 để khởi chạy số lượng hoặc số lượng máy chủ ảo tùy theo nhu cầu của mình, định cấu hình bảo mật và kết nối mạng cũng như quản lý dung lượng lưu trữ. Amazon EC2 cho phép bạn tăng hoặc giảm quy mô để xử lý các thay đổi về yêu cầu hoặc mức độ phổ biến tăng đột biến, giúp bạn giảm nhu cầu dự báo lưu lượng.

* + 1. **Các tính năng của Amazon EC2**

Amazon EC2 cung cấp các tính năng sau:

* Môi trường máy tính ảo, được gọi là phiên bản
* Các mẫu được cấu hình sẵn cho các phiên bản của bạn, được gọi là Amazon Machine Images (AMI), đóng gói các bit bạn cần cho máy chủ của mình (bao gồm cả hệ điều hành và phần mềm bổ sung)
* Các cấu hình khác nhau của CPU, bộ nhớ, dung lượng lưu trữ và kết nối mạng cho các phiên bản của bạn, được gọi là các loại phiên bản
* Bảo mật thông tin đăng nhập cho các phiên bản của bạn bằng các cặp khóa (AWS lưu trữ khóa chung và bạn lưu trữ khóa riêng tư ở nơi an toàn)
* Dung lượng lưu trữ dành cho dữ liệu tạm thời bị xóa khi bạn dừng, ngủ đông hoặc chấm dứt phiên bản của mình, được gọi là dung lượng lưu trữ phiên bản
* Ổ đĩa lưu trữ liên tục cho dữ liệu của bạn bằng cách sử dụng Amazon Elastic Block Store (Amazon EBS), được gọi là ổ đĩa Amazon EBS
* Nhiều vị trí thực cho tài nguyên của bạn, chẳng hạn như phiên bản và ổ đĩa Amazon EBS, được gọi là Khu vực và Vùng sẵn sàng
* Tường lửa cho phép bạn chỉ định các giao thức, cổng và dải IP nguồn có thể tiếp cận các phiên bản của bạn bằng cách sử dụng các nhóm bảo mật
* Địa chỉ IPv4 tĩnh cho điện toán đám mây động, được gọi là địa chỉ IP đàn hồi
* Siêu dữ liệu, được gọi là thẻ, mà bạn có thể tạo và gán cho tài nguyên Amazon EC2 của mình
* Các mạng ảo mà bạn có thể tạo được tách biệt về mặt logic với phần còn lại của Đám mây AWS và bạn có thể tùy ý kết nối với mạng của riêng mình, được gọi là các đám mây riêng ảo (VPC)
  + 1. **Dịch vụ liên quan**

**Amazon EC2 Auto Scaling** Amazon EC2 Auto Scaling đảm bảo luôn có sẵn số lượng phiên bản Amazon EC2 chính xác để xử lý tải cho ứng dụng của mình. Khi tạo bộ sưu tập các phiên bản EC2, được gọi là nhóm Auto Scaling. Chúng ta có thể chỉ định số lượng phiên bản tối thiểu trong mỗi nhóm Auto Scaling và Amazon EC2 Auto Scaling đảm bảo rằng nhóm của mình không bao giờ thấp hơn kích thước này. Bạn có thể chỉ định số lượng phiên bản tối đa trong mỗi nhóm Auto Scaling và Amazon EC2 Auto Scaling đảm bảo rằng nhóm của bạn không bao giờ vượt quá kích thước này. Nếu chúng ta chỉ định dung lượng mong muốn, khi mình tạo nhóm hoặc bất kỳ lúc nào sau đó, Amazon EC2 Auto Scaling đảm bảo rằng nhóm của chúng ta có nhiều phiên bản như vậy. Nếu như chỉ định các chính sách thay đổi quy mô, thì Amazon EC2 Auto Scaling có thể khởi chạy hoặc chấm dứt các phiên bản khi nhu cầu đối với ứng dụng của bạn tăng hoặc giảm.

**AWS CloudFormation:** AWS CloudFormation là dịch vụ giúp bạn lập mô hình và thiết lập các tài nguyên AWS để bạn có thể dành ít thời gian hơn cho việc quản lý các tài nguyên đó và có nhiều thời gian hơn để tập trung vào các ứng dụng chạy trong AWS. Bạn tạo một mẫu mô tả tất cả các tài nguyên AWS mà bạn muốn (chẳng hạn như phiên bản Amazon EC2 hoặc phiên bản Amazon RDS DB) và CloudFormation sẽ đảm nhận việc cung cấp và định cấu hình các tài nguyên đó cho bạn. Bạn không cần phải tạo và định cấu hình tài nguyên AWS riêng lẻ cũng như tìm ra cái gì phụ thuộc vào cái gì; CloudFormation xử lý việc đó. Các tình huống sau đây minh họa cách CloudFormation có thể trợ giúp.

**AWS Elastic Beanstalk:** Với Elastic Beanstalk, bạn có thể nhanh chóng triển khai và quản lý các ứng dụng trong Đám mây AWS mà không cần phải tìm hiểu về cơ sở hạ tầng chạy các ứng dụng đó. Beanstalk đàn hồi giảm độ phức tạp trong quản lý mà không hạn chế lựa chọn hoặc kiểm soát. Bạn chỉ cần tải ứng dụng của mình lên và Elastic Beanstalk sẽ tự động xử lý các chi tiết về cung cấp dung lượng, cân bằng tải, mở rộng quy mô và theo dõi tình trạng của ứng dụng.

**AWS OpsWorks**: AWS OpsWorks là dịch vụ quản lý cấu hình giúp bạn định cấu hình và vận hành các ứng dụng trong doanh nghiệp đám mây bằng cách sử dụng Puppet hoặc Chef. AWS OpsWorks Stacks và AWS OpsWorks for Chef Automate cho phép bạn sử dụng các giải pháp và sách dạy nấu ăn của Chef để quản lý cấu hình, trong khi OpsWorks dành cho Puppet Enterprise cho phép bạn định cấu hình máy chủ chính Puppet Enterprise trong AWS. Puppet cung cấp một bộ công cụ để thực thi trạng thái mong muốn của cơ sở hạ tầng của bạn và tự động hóa các tác vụ theo yêu cầu.

**Elastic Load Balacing**: dùng để tự động phân phối lưu lượng truy cập ứng dụng đến trên nhiều phiên bản

**Amazon Relational Database Service (Amazon RDS)** Để có cơ sở dữ liệu quan hệ được quản lý trên đám mây, hãy sử dụng Amazon Relational Database Service (Amazon RDS) để khởi chạy một phiên bản cơ sở dữ liệu. Mặc dù bạn có thể thiết lập cơ sở dữ liệu trên phiên bản EC2, nhưng Amazon RDS mang lại lợi thế trong việc xử lý các tác vụ quản lý cơ sở dữ liệu của bạn, chẳng hạn như vá lỗi phần mềm, sao lưu và lưu trữ các bản sao lưu.

**Amazon Elastic Container Service (Amazon ECS)** dùng để quản lý bộ chứa Docker trên một cụm phiên bản EC2 dễ dàng hơn

**Amazon CloudWatch** dùng để giám sát số liệu thống kê cơ bản cho các phiên bản và ổ đĩa Amazon EBS của bạn

**Amazon GuardDuty** dùng để phát hiện việc sử dụng phiên bản EC2 có khả năng trái phép hoặc gây hại

## CHƯƠNG 2: THIẾT KẾ VÀ CÀI ĐẶT

1. **Phân tích yêu cầu**

Yêu cầu của đề tài:

* Xây dựng VPC: 1 public subnet + 1 private subnet
* Khởi tạo 1 EC2 (cài đặt web server) + 1 DynamoDB
* Website quản lý thông tin sinh viên (Giao diện + Xử lý + Database đơn giản)

1. **Website quản lý sinh viên**

* Các chức năng
* Quản lý (Sinh viên, điểm giảng viên, khoa, lớp sinh viên, học phần, lớp học phần
* Quản lý tài khoản(Sing Viên, Giảng Viên,….)
* Quản lý vai trò của các tài khoản

- Các role:

* Người quản trị
* Giảng viên

- Phân chia chức năng theo role

* Người quản trị: Tất cả chức năng của website quản lý sinh viên
* Giảng viên: Quản lí điểm các nhóm lớp giảng viên đó dạy và quản lý thông tin của chính giảng viên đó

1. **Thiết kế Cơ sở dữ liệu NoSQL**

**faculty**(facultyId, facultyName, image, deleted)

**student\_class**(studentClassId, studentClassName, facultyId, deleted)

**student**(studentId, studentClassId, studentName, dob, address, gender, phone, image, deleted)

**subject**(subjectId, subjectName, creditsNo, periodsNo, deleted)

**grade**(studentId, subjectGroupId, middleGrade, finalGrade, totalGrade, deleted)

**lecture**(lectureId, lectureName, dob, address, gender, phone, image, deleted)

**subject\_group**(subjectGroupId, subjectGroupName, subjectId, lectureId, deleted)

**user**(username, password, lectureId, deleted)

**role**(roleId, roleName, deleted)

**user\_role**(userName, roleId)

* 1. **Chạy project trên AWS**

Clone project theo link github

* + 1. **Xây dựng VPC**

**Xây Dựng VPC với**:

Địa chỉ mạng IPv4 cho VPC: **10.0.0.0/16**

Đặt cấu hình 2 Subnet:

* 1 Public Subnet có địa chỉ mạng: **10.0.0.0/24**
* 1 Private Subnet có địa chỉ mạng: **10.0.1.0/24**

Ta có cấu hình như sau

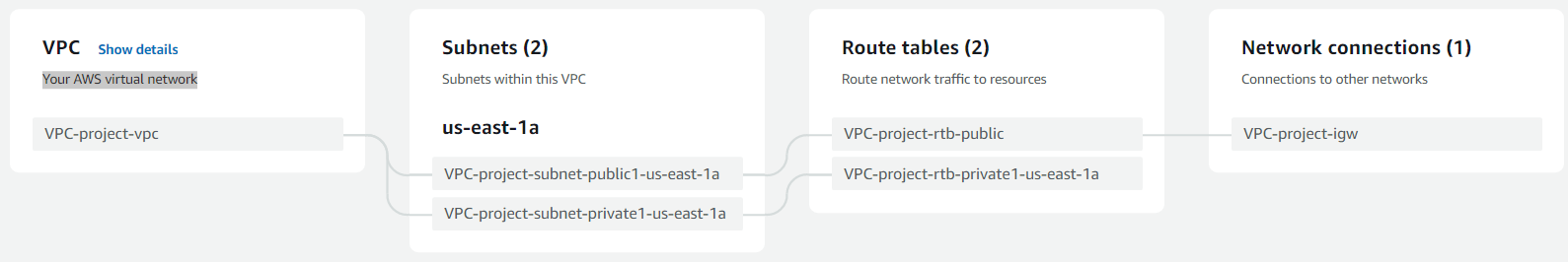
Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ta có thể xem Mạng ảo AWS của mình

****

Tạo Một **Security Group** (coi như một tường lửa ảo) giúp lọc các traffic đến VPC với cấu hình:

* Security group name: Web Security Group
* Inbound rules: HTTP, HTTPs, All Traffic
* Source: Anywhere – IPv4

Ta cài đặt cấu hình như sau:

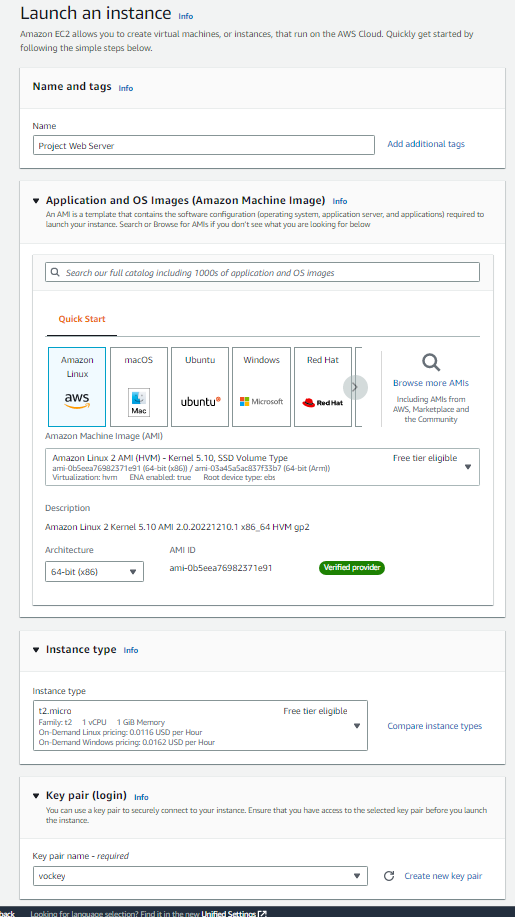
Ảnh có chứa văn bản

Mô tả được tạo tự động

* + 1. **Tạo EC2** **instance**

Tạo một EC2 instance có tên là **Project Web Server** với:

* Amazon Machine Image (AMI): **Amazon Linux 2**
* Key pair name: **vockey**

****

* Tại Network Settings, chọn Edit và chỉnh sửa cấu hình:
* VPC: **VPC-project-vpc**
* Chọn Subnet: **public**
* Auto-assign public IP: **Enable**
* Firewall (security groups): click **Select existing security group**
* Common security groups: chọn **Select existing security group** => **Web Security Group**

**Cấu hình** **Network:**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Click: **Launch istance**

Đợi một lát để khởi chạy EC2

Sau khi EC2 instance vừa tạo được khởi chạy, ta tiến hành kết nối đến máy ảo EC2

* Chạy lệnh: **sudo su** – để truy cập **user root**
* Chạy lệnh: **sudo yum update** – để cập nhật các package cho Hệ điều hành

Cài đặt **Apache Tomcat** trên máy ảo EC2 để có thể chạy ứng dụng java web:

* Vào đường link để tải bản Tomcat mới nhất: [Apache Tomcat® - Apache Tomcat 10 Software Downloads](https://tomcat.apache.org/download-10.cgi)
* Chạy lệnh: **cd /opt** – để chuyển đến thư mục opt
* Chạy lệnh: **wget <Đường dẫn đến file TomCat>** – để tải file tom cat
* Chạy lệnh: **tar -xvzf /opt/apache-tomcat-10.0.27.tar.gz** – để giải nén file vừa tải

Cài đặt Java:

* Chuyển đến thư mục tom cat: **cd apache-tomcat-10.0.27/bin**
* Cài đặt Java: **yum install java-1.8.0**

Truy cập đến Manage App:

* Tìm các file content.xml : **find / -name context.xml**
* Comment các thẻ **valve** trong từng file

1. **vim /opt/apache-tomcat-10.0.27/webapps/host-manager/META-INF/context.xml**
2. **vim /opt/apache-tomcat-10.0.27 /webapps/manager/META-INF/context.xml**

Thêm các username và role vào file bằng lệnh

* Chạy lệnh: **vim ../conf/tomcat-users.xml**
* Thêm vào File

**<role rolename="manager-gui"/>**

**<role rolename="manager-script"/>**

**<role rolename="manager-jmx"/>**

**<role rolename="manager-status"/>**

**<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>**

**<user username="deployer" password="deployer" roles="manager-script"/>**

**<user username="tomcat" password="s3cret" roles="manager-gui"/>**

Chạy lệnh: **./startup.sh**

* + 1. **Tạo bucket S3**

1. Truy cập dịch vụ S3 trên AWS
2. Chọn Create Bucket
3. Đặt tên bucket: **studentbk**
4. Chọn **ACLs enabled**
5. Bỏ chọn **Block all public access**
6. Cick vào ô **I acknowledge that the current settings might result...** để cho phép truy cập link các file được lưu trong S3 từ internet
7. Sau khi tạo thành công, chọn bucket mới tạo,
8. Chọn tab Permission
   * Bucket Policy
   * Edit
   * Copy Policy sau và dán vào ô trống

{

"Version":"2012-10-17",

"Statement":[

{

"Sid":"PublicReadGetObject",

"Effect":"Allow",

"Principal":"\*",

"Action":[ "s3:GetObject" ],

"Resource":[ "arn:aws:s3:::studentbk/\*" ]

}

]

}

* + Save
    1. **Tạo table DynamoDB và dữ liệu ban đầu**

Ta sẽ tạo table DynamoDB từ máy

1. Đầu tiên, ta cần cài đặt AWS CLI
2. Sau khi cài xong, ta mở CMD và chạy lệnh **aws configure**
3. Tiếp theo, nhập lần lượt **Access key**, **Secret access key**, **region: 'us-east-1'**, **output format: 'json'** được lấy từ tài khoản Learner Lab sau khi đã Start Lab
4. Chạy lệnh **aws configure set aws\_session\_token ‘{value}’** với **{value}** là token của phiên đăng nhập hiện tại mà tài khoản AWS đang chạy
5. Chạy file **DynamoDBUtils.java** để tiến hành tạo các table trên **Amazon DynamoDB** và thêm dữ liệu ban đầu
   * 1. **Deploy WAR file**

Vào địa chỉ IPv4 của instance: Project Web Server

Click vào Manage app

Nhập địa

* Username: admin
* Password: admin

Deploy website lên host.

Vào và khởi chạy web site.

## CHƯƠNG 3: KẾT LUẬN

1. **Kết quả đạt được**

Sau thời gian nghiên cứu và thực hiện đề tài “**Xây dựng VPC và chạy webserver trên AWS**”, nhóm chúng em đã đạt được những kết quả như sau:

* Nắm bắt được các kiến thức cũng như những cách hoạt động của các dịch vụ VPC, EC2, S3, DynamoDB và áp dụng kiến thức để thiết kế và xây dựng một mô hình tương tác với database.
* Nắm bắt được mô hình hoạt động, cách tạo một VPC, cách phân chia subnet
* Biết cách deploy website lên EC2
* Biết dùng S3 để lưu trữ ảnh
* Xây dựng hoàn chỉnh một website quản lý sinh viên đơn giản
* **Ưu điểm**
* Website với giao diện đơn giản
* Tương tác nhanh vì DynamoDB là CSDL dạng NoSQL

1. **Nhược điểm**

* Các bước thực hành chưa đc thuần thục
* Giao diện khá đơn giản
* Còn phù thuộc nhiều vào aws

**TÀI LIỆU THAM KHẢO**

1. Amazon EC2: [What is Amazon EC2? - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html)
2. Amazon VPC: [What is Amazon VPC? - Amazon Virtual Private Cloud](https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html)
3. Amazon DynamoDB: [What is Amazon DynamoDB? - Amazon DynamoDB](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html)
4. Link github tham khảo: <https://github.com/tranquoctuan2001/dynamodb-vpc-nodejs>