**BỘ GIÁO DỤC VÀ ĐÀO TẠO TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT TP.HCM KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

#### ●●○○●●○○●●○○●●○○●●



**MÔN HỌC: ĐIỆN TOÁN ĐÁM MÂY**

**CHỦ ĐỀ:**

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

Mã môn học: CLC0332779\_22\_1\_05CLC

GVHD: Huỳnh Xuân Phụng

STT NHÓM : 43

Phan Nguyễn Hoài Nam – 20110080

Nguyễn Trúc An - 20110087

TP. Hồ Chí Minh, tháng 11 năm 2022

**Trường Đại học Sư phạm Kỹ thuật CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

#### TP. Hồ Chí Minh Độc lập – Tự do – Hạnh phúc

**Khoa Đào tạo Chất lượng cao**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Sinh viên thực hiện báo cáo đồ án:**

Phan Nguyễn Hoài Nam – 20110080

Nguyễn Trúc An - 20110087

**Đề tài:** XÂY DỰNG VPC VÀ CHẠY WEBSERVER TRÊN AWS

**Nhận xét của Giảng viên:**

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

TP. Hồ Chí Minh, ngày … tháng … năm 2022

**Giảng viên hướng dẫn**

**MỤC LỤC**

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

**PHÂN CÔNG**

**PHẦN 1: MỞ ĐẦU**

1. Tính cấp thiết của đề tài

2. Đối tượng nghiên cứu

3. Phạm vi nghiên cứu

4. Kết quả dự kiến

**PHẦN 2: NỘI DUNG**

**Chương 1: Cơ sở lý thuyết**

1.1 Dịch vụ Amazon Virtual Private Cloud (VPC)

1.2 Dịch vụ Amazon DynamoDB

1.3 Dịch vụ Amazon Elastic Compute Cloud (EC2)

1.4 Dịch vụ Amazon Simple Storage Service (S3)

**Chương 2: Hướng dẫn cài đặt**

**Chương 3: Kết luận**

**Tài liệu tham khảo**

**BẢNG PHÂN CÔNG**

Link github nhóm: <https://github.com/nguyentrucan/CloudComputing.git>

Phân công thực hiện xây dựng đề tài:

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **MSSV** | **Công việc** | **Tiến độ** |
| Nguyễn Trúc An | 20110087 | * Đăng nhập * Quản lý khoa * Quản lý giảng viên * Quản lý điểm * Quản lý lớp học phần | 100% |
| Phan Nguyễn Hoài Nam | 20110080 | * Quản lý sinh viên * Quản lý lớp sinh viên * Quản lý môn học * Quản lý vai trò hệ thống | 100% |

**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.

**2. 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.

**3. 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

**4. Kết quả dự kiến**

- 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: CỞ SỞ LÝ THUYẾT**

* 1. **Amazon Virtual Private Cloud (VPC)**

**Tổng quan:**

* Amazon Virtual Private Cloud (Amazon VPC) là dịch vụ cho phép bạn khởi chạy các tài nguyên AWS trong mạng ảo cô lập theo logic mà bạn xác định. Bạn có toàn quyền kiểm soát môi trường mạng ảo của mình, bao gồm lựa chọn dải địa chỉ IP, tạo các mạng con, cấu hình các bảng định tuyến và cổng kết nối mạng. Bạn có thể dùng cả IPv4 và IPv6 cho hầu hết các tài nguyên trong đám mây riêng ảo, giúp bảo mật nghiêm ngặt và truy cập dễ dàng các tài nguyên cũng như ứng dụng.
* 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

**Chức năng:**

* **VPC**: là một mạng ảo gần giống với mạng truyền thống mà ta sẽ sử dụng nó trong trung tâm dữ liệu của mình. Sau khi tạo mới một VPC, ta có thể tạo các subnets
* **Subnets**: Một subnet là một dãi địa chỉ IP trong VPC ta tạo. Một subnet bắt buộc phải ở trong một AZ. Sau khi thêm subnets, bạn có thể triển khai tai nguyên lên VPC
* **Địa chỉ IP**: ta có thể gán địa chỉ IPv4 và IPv6 cho VPC hoặc subnet mình tạo.Nếu ta có các địa chỉ IPv4, IPv6 công khai của riêng mình, ta có thể cấp phát chúng cho các tài nguyên trong VPC như EC2 instance, NAT gateways, Network Load Balancers
* **Routing**: Cung cấp bảng Route để xác định nơi các lưu lượng mạng từ các subnet hoặc gateway hướng tới
* **Gatewates và endpoints**: Một gateway giúp kết nối VPC tới một mạng khác ngoài VPC. Ví dụ, ta dùng một internet gateway để kết nối VPC của mình tới internet. Dùng **VPC endpoint** để kết nối các dịch vụ AWS riêng tư mà không cần đến **internet gateway** hoặc thiết bị **NAT**
* **Peering connections:** sử dụng kết nối VPC peering để định tuyến lưu lượng giữa các tài nguyên trong 2 VPCs
* **Traffic Mirroring:** Copy lưu lượng mạng từ các network interfaces và gửi tới các thiết bị giám sát và bảo mật để phân tích các gói tin
* **Transit gateways:** Sử dụng một transit gateway (hoạt động giống một hub trung tâm) để định tuyến lưu lượng mạng giữa các VPC, kết nối VPN và các kết nối AWS Direct Connect
* **VPC Flow Logs:** Flow log dùng để ghi lại thông tin về các địa chi IP truy cập đến hoặc đi từ các network interface trong VPC ta tạo
  1. **Amazon DynamoDB**

**Tổng quan:**

* DynamoDB là môt dịch vụ cơ sở dữ liệu NoSQL có khả năng đáp ứng hiệu suất cao và nhanh kèm theo khả năng mở rộng. Nếu bạn là một nhà phát triển, bạn có thể sử dụng DynamoDB để tạo ra một bảng có khả năng lưu trữ và truy xuất bất kỳ số lượng dữ liệu, mà vẫn có thể phục vụ cho bất kỳ mức độ request traffic. DynamoDB được quản lý đầy đủ, hỗ trợ cả mô hình dữ liệu dưới dạng tài liệu và cặp value-key.
* DynamoDB tự động phân tán dữ liệu và traffic của một bảng ra một số lượng server vừa đủ để có thể xử lý request capacity đặt ra bởi khách hàng và lượng dữ liệu lưu trữ, đồng thời đảm bảo hiệu suất nhanh và đồng nhất. Tất cả dữ liệu được lưu trữ trên SSD và tự động được sao chép ra các vùng sẵn sàng (Availability Zones) trong một khu vực (Region) để cung cấp độ sẵn sàng cao và độ bền của dữ liệu (high availablity and data durability)

**Chức năng:**

* **AWS Glue Views** cho phép các lập trình viên tạo nên các giao diện trực quan cho phép dễ dàng tìm kiếm, kết hợp, nhân bản giữa nhiều kho dữ liệu khác nhau theo thời gian thực mà không cần phải tự viết code tùy biến, dữ liệu tổng hợp có thể được chuyển đến các dịch vụ khác như: Redshift, S3, Aura … phục vụ nhiều mục đích khác nhau.
* **AWS Database Migration Service (AWS DMS)** cho phép chuyển dữ liệu trong cơ sở dữ liệu quan hệ hoặc MongoDB sang DynamoDB
* **PartiSQL** là một ngôn ngữ truy vấn tương thích với SQL, có thể liên kết với nhiều cơ sở dữ liệu với nhiều định dạng dữ liệu khác nhau như dữ liệu có cấu trúc, dữ liệu bán cấu trúc, dữ liệu dạng lưới. Ngôn ngữ truy vấn này được áp dụng trong nhiều dịch vụ của AWS, trong đó có Dynamodb.
* **Amazon Kinesis** được sử dụng để tổng hợp và xử lý các luồng dữ liệu lớn theo thời gian thực. Các dữ liệu này được lưu trong các data record và có thể được sử dụng cho nhiều mục đích khác nhau: tạo thông báo, tạo các bảng báo cáo, các bảng giá có giá trị thay đổi theo thời gian,…
  1. **Amazon Elastic Compute Cloud (EC2)**

**Tổng quan:**

* Amazon Elastic Compute Cloud (Amazon EC2) là một cơ sở hạ tầng điện toán đám mây được cung cấp bởi Amazon Web Services (AWS) giúp cung cấp tài nguyên máy tính ảo hoá theo yêu cầu.
* Amazon EC2 cung cấp các ứng dụng máy tính ảo hoá có thể mở rộng về khả năng xử lý cùng các thành phần phần cứng ảo như bộ nhớ máy tính (ram), vi xử lý, linh hoạt trong việc lựa chọn các phân vùng lưu trữ dữ liệu ở các nền tảng khác nhau và sự an toàn trong quản lý dịch vụ bởi kiến trúc ảo hoá đám mây mạnh mẽ của AWS.
* Amazon EC2 sẽ cung cấp một hoặc máy chủ ảo có thể kết hợp với nhau để dễ dàng triển khai ứng dụng nhanh nhất và đảm bảo tính sẵn sàng cao nhất. Thậm chí về mặt thanh toán bạn dễ dàng biết được các mức chi phí cần thanh toán dựa trên thông tin tài nguyên bạn sử dụng

**Chức năng:**

* Cung cấp môi trường máy tính ảo, còn gọi là instance
* Cung cấp sẵn các mẫu hệ điều hành cho instance (còn gọi là Amazon Machine Images - AMIs), gồm các gói cài đặt cần thiết cho server
* Cho phép tuỳ chỉnh, thiết lập cấu hình instance của bạn: CPU, bộ nhớ, lưu trữ…
* Bảo mật thông tin cho instance thông qua việc dùng key – pairs (AWS lưu trữ public key, người dùng lưu trữ private key)
* Lưu trữ tạm thời dữ liệu đã bị xóa sau khi ta thực hiện stop, hibernate hoặc terminate instance
* Cung cấp tường lửa cho phép bạn chỉ định giao thức, cổng dịch vụ và các nguồn IP được phép truy cập instance của bạn
* Dùng địa chỉ IPv4 tĩnh, hay gọi là Elastic IP address

* 1. **Amazon Simple Storage Service (S3)**

**Tổng quan**

* Là dịch vụ đám mây lưu trữ do đó bạn có thể tải lên các tệp, các tài liệu, các dữ liệu tải về của người dùng hoặc các bản sao lưu
* Amazon S3 lưu trữ dữ liệu như các object trong các bucket. Một object gồm 1 file và metadata mô tả cho file (tùy chọn).
* Để lưu 1 object trong Amazon S3, bạn tải file lên 1 bucket. Khi đã tải file, bạn có thể gán quyền cho đối tượng cũng như bổ sung metadata.
* Bucket là các thùng chứa cho các object. Bạn có thể tạo 1 hay nhiều bucket nhưng mỗi bucket sẽ có tên riêng biệt, không trùng nhau. Với mỗi bucket, bạn có thể điều khiển việc truy xuất đến nó (ai có thể tạo, xóa và xem các object trong bucket), xem nhật ký truy xuất đến bucket và đến các object bên trong, cũng như chọn region mà Amazon S3 sẽ lưu bucket và nội dung trong nó.

**Chức năng**

* Lưu trữ các tệp, tài liệu hoặc bất cứ loại dữ liệu nào của người dùng
* Truy xuất dữ liệu
* Có khả năng mở rộng và độ linh hoạt cao
* Bảo mật dữ liệu được lưu và phân quyền truy cập dữ liệu

**CHƯƠNG 2: HƯỚNG DẪN CÀI ĐẶT**

**1. Cài đặt một máy ảo EC2**

- Tạo máy ảo EC2 Cloud9Ubuntu cấu hình thì tùy theo nhu cầu sử dụng

- Tạo lần lượt các giao thức SSH, HTTP, HTTPS và một giao thức có Port là 8889 để truy cập web cho máy ảo.

**2. Tạo 1 địa chỉ ElasticIP và gán nó vào máy ảo EC2 vừa tạo**

Ở đây nhóm mình dùng ElasticIP là:

**3. Thực hiện cấu hình máy ảo**

- Chạy lệnh "sudo apt-get update" để cập nhật phiên bản mới nhất, "sudo apt-get install npm" để cài đặt Nodejs.

- Chạy lệnh aws configure và cung cấp lần lượt các thông tin AWS Access Key ID, AWS Secret Access Key, Default region name, Default output format.

* Chạy lệnh aws configure set

aws\_session\_token "FwoGZXIvYXdzEEcaDGslf2+P2VbqJpNPOSLPASTQEHPAq6GJx8Fphl91bcH6rRWZ8tNtG10LgGfk22nC2fn79IaLbmiuBmuZdddHbz57QGFvPRHcn8Io8uzLy7jgCdOPA8sr9Sdkb6jwiW93ApatsenznX/GbcebmhwwdBqnTwWCbG+x4oyB+nnznAUlzQ1AbcDuvcxRHHI29demW6I99i6l8qcXJ84BDF6orwEhhEDVRXpKIOveg46nacSTXbJXVRUvxEViHULRVpIzu4Vao3eThyRyZ6sdshnvEoRT9gmMLRKzAALk92j4pSjXz5CVBjItI4riGHdsKzHdchBi1M8U+77quVOfW42tednV7vYrGwDhERpjgsPYeYjXzzXj" để cung cấp token.

* Kiểm tra cách cấu hình kết nối với AWS CLI có thành công hay không: aws sts get-caller-identity
* Tạo thêm các Simple queue service, với 25 SQS

(ChangePass,DeleteDiemm,DeleteGV,DeleteKhoa,DeleteLop,DeleteMonHoc,DeletePhanCong,DeleteSV,Delete\_Account,InsertDiem,InsertGV,InsertLop,InsertMonHoc,InsertPhanCong,InsertSV,Insert\_Account,UpdateDiem,UpdateGV,UpdateKhoa,UpdateLop,UpdateMonHoc,UpdatePhanCong,UpdateSV,Update\_Account,test, )

* Tạo 8 tables Dynamodb

(Account,Diems,GiangViens,Khoas,Lops,MonHocs,PhanCongs,SinhViens)

* Tên các table trong DynamoDB có thể thấy ở các file \*.js ở trong folder /environment/Model

**4. Chạy trương trình**

- Đi đến thư mục environment bằng lệnh "cd AWS/environment"

- Chạy lệnh "node index" để chạy trang web

**5. Áp dụng VPC vào ứng dụng**

- Tạo vpc:

aws ec2 create-vpc-endpoint --vpc-id vpc-086106c39d2c8ae5c --service-namecom.amazonaws.us-east-1.dynamodb--route-table-ids rtb-0a374d0d08f4479fb

**6. Deploy Docker**

Cấu hình file Dockerfile

* Đăng nhặp Docker trên máy: docker Login

Tiến hành build file docker bằng lệnh: docker build -t 19133064/aws:19133064 -f DockerFile .

* Lệnh chạy Docker:

docker run 19133064/aws:19133064

* Deploy Docker lên DockerHub:

docker push 19133064/aws:191133064s

**7. Chú ý**

Nếu xảy ra trùng port thì phải kill port:

sudo kill -9 $(sudo lsof -t -i:8889)

**CHƯƠNG 3: KẾT LUẬN**

**3.1 Kết quả đạt được**

Sau một 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:

Kiến thức tìm hiểu được

* 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
* Biết sử dụng AWS SDK bằng Java

Chương trình đã làm được: Xây dựng hoàn chỉnh một website quản lý sinh viên đơn giản bằng Java, HTML, CSS, JS và các dịch vụ của AWS: VPC, EC2, S3, DynamoDB với các chức năng cơ bản như:

* CRUD trên website
* Tạo các table trong DynamoDB.
* Thêm, sửa, xóa, đọc các Item trong table của DynamoDB.

**3.2 Ưu, nhược điểm**

Ưu điểm:

* Website với giao diện đơn giản, dễ tương tác
* Tương tác nhanh vì DynamoDB là cơ sở dữ liệu NoSQL

Nhược điểm:

* Các bước cài đặt và deploy lên EC2 khó
* Database đơn giản
* Giao diện không được đẹp mắt
* Tốn phí vì sử dụng dịch vụ của AWS và phụ thuộc phần lớn vào AWS trong vấn đề chạy code

**TÀI LIỆU THAM KHẢO**

1. Amazon EC2: <https://docs.aws.amazon.com/ec2/index.html>
2. Amazon VPC: <https://docs.aws.amazon.com/vpc/>
3. Amazon DynamoDB: <https://docs.aws.amazon.com/dynamodb/>
4. Amazon S3: <https://docs.aws.amazon.com/s3/index.html>
5. Link github tham khảo: <https://github.com/tranquoctuan2001/dynamodb-vpc-nodejs>