|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH**  **KHOA CÔNG NGHỆ THÔNG TIN** |
|  |
| **MÔN HỌC: ĐIỆN TOÁN ĐÁM MÂY** |
|  |
| **ĐỀ TÀI**  **XÂY DỰNG VPC VÀ CHẠY WEBSERVER TRÊN AWS** |
| **GVHD: GV. HUỲNH XUÂN PHỤNG** |
| **Nhóm sinh viên thực hiện: Nhóm**   1. **NGUYỄN TUẤN VŨ - 18133064** 2. **NGUYỄN XUÂN HIỆU - 18133013** |
| **Tp. Hồ Chí Minh, tháng 11 năm 2020** |

**Lời nói đầu**

Chúng em xin chân thành cảm ơn sự hướng dẫn tận tình của thầy

***HUỲNH XUÂN PHỤNG*** đã giúp chúng em hoàn thành bài báo cáo cuối kì bộ môn Điện toán đám mây Tuy đã cố gắng học tập, tìm hiểu, phân tích thiết kế và xây dựng hệ thống nhưng ắt hẳn thiếu sót là điều không thể tránh khỏi. Chúng em rất mong nhận được sự thông cảm và góp ý chân thành của Thầy. Chúng em xin chân thành cảm ơn.

**Mục lục:**

Contents

[Phần mở đầu 4](#_Toc74523222)

[1. Nội dung đề tài 4](#_Toc74523223)

[2. Triển khai đề tài 4](#_Toc74523224)

[Phần lý thuyết 5](#_Toc74523225)

[1. Cloud Computing 5](#_Toc74523226)

[2. Amazon Virtual Private Cloud (Amazon VPC) 5](#_Toc74523227)

[2.1. Subnet 6](#_Toc74523228)

[2.2. Route Table 6](#_Toc74523229)

[2.3. Internet Gateway 7](#_Toc74523230)

[2.4. Network address translation (NAT) 8](#_Toc74523231)

[2.5. Security Group 8](#_Toc74523232)

[3. DynamoDB 10](#_Toc74523233)

[3.1. Các thành phần DynamoDB cơ bản: 10](#_Toc74523234)

[Phần demo 13](#_Toc74523235)

[1. VPC 13](#_Toc74523236)

[1.1. ElasticIP – Tạo Public IP để kết nối tới máy chủ EC2 13](#_Toc74523237)

[1.2. Tạo VPC có Public Subnet 1 và Private Subnet 1 13](#_Toc74523238)

[1.3. Tạo Public Subnet 2 và Private Subnet 2 15](#_Toc74523239)

[1.4. Route Table 17](#_Toc74523240)

[1.5. Security Group 18](#_Toc74523241)

[1.6. Tạo EC2 Web Server 19](#_Toc74523242)

[1.7. Tạo RDS Database Server 22](#_Toc74523243)

[1.8. Sử dụng MySQL Workbrench, dùng SSH kết nối EC2 và MySQL Server 28](#_Toc74523244)

[1.9. Tạo RDS MySQL Database schoolmanagement 29](#_Toc74523245)

[1.10. Kết nối SSH bằng CMD truy cập vào EC2 31](#_Toc74523246)

[1.11. Download Project School PHP – MySQL 33](#_Toc74523247)

[1.12. Kết quả Demo 35](#_Toc74523248)

[2. DynamoDB 37](#_Toc74523249)

[1.1. Cài đặt python và các thư viện cần thiết 37](#_Toc74523250)

[1.2. Tạo Endpoint DynamoDB 37](#_Toc74523251)

[1.3. Thiết lập Security Group cho EC2 đã tạo ở VPC 39](#_Toc74523252)

[1.4. Git Clone TicTacToe (đã tải ở VPC) 39](#_Toc74523253)

[1.5. Thiết lập key kết nối 39](#_Toc74523254)

[1.6. Chạy Flask Server 40](#_Toc74523255)

[Phần kết luận 42](#_Toc74523256)

[1. Kết quả đạt được 42](#_Toc74523257)

[2. Kết quả chưa đạt được 42](#_Toc74523258)

[3. Hướng phát triển 42](#_Toc74523259)

[4. Tài liệu tham khảo 42](#_Toc74523260)

1. Phần mở đầu
   1. Nội dung 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)

Github: <https://github.com/Vu-Nguyen2201/VPC-Webserver-AWS>

* 1. Triển khai đề tài

Xây dựng 1 VPC có 2 public subnet + 2 private subnet

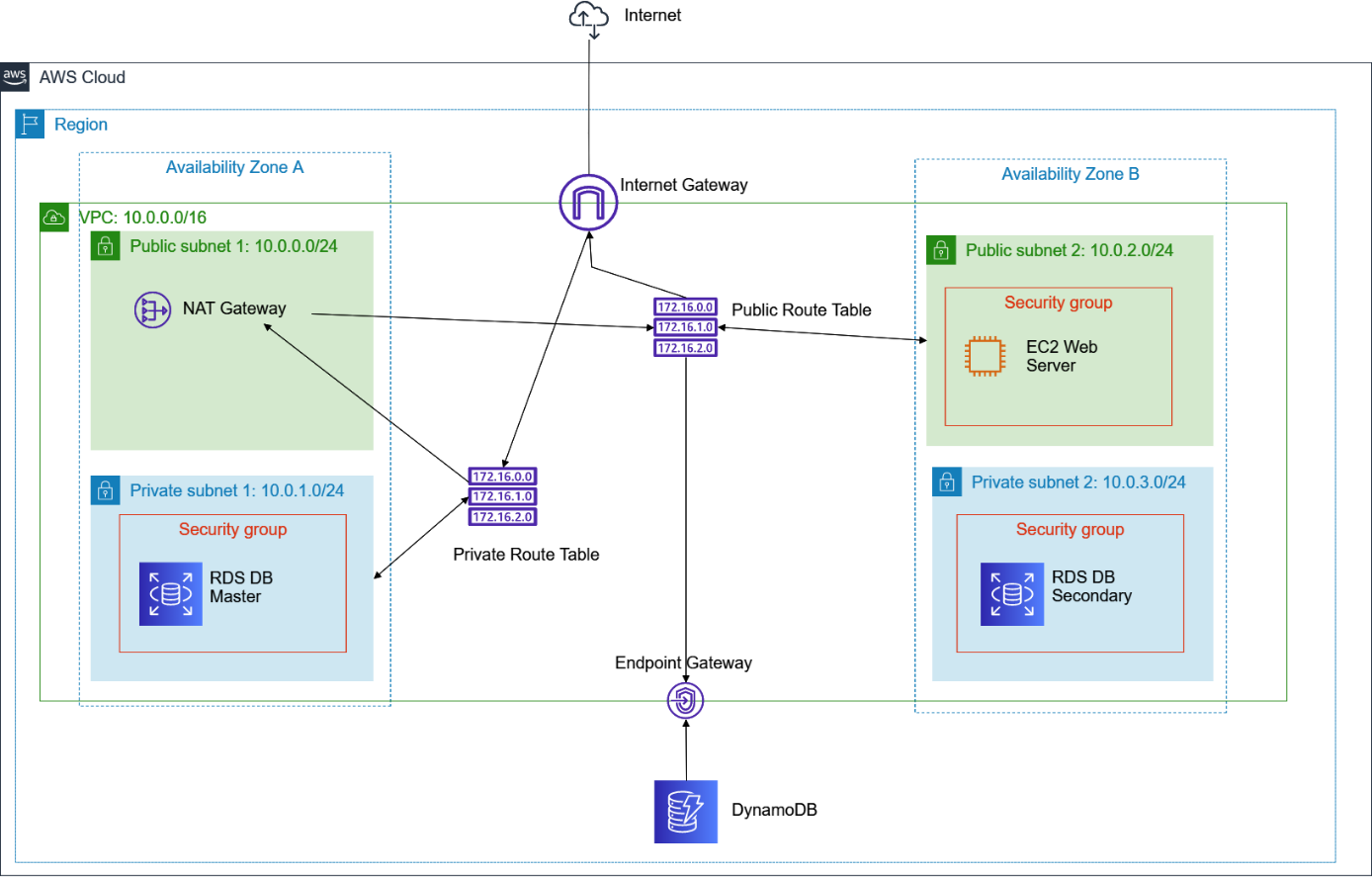
Project 1 – Trang web quản lý khoá học PHP - MySQL

- Thiết lập máy chủ EC2 trên public subnet và cài đặt PHP Web Service

- Thiết lập RDS MySQL, cài đặt schoolmanagement DB trên private subnet

Project 2 – Game đánh caro chạy trên Flask Framework (Python) và sử dụng DynamoDB

- Tạo 1 DynamoDB Endpoint cho phép EC2 kết nối với DynamoDB



1. Phần lý thuyết
   1. Cloud Computing

Điện toán đám mây là gì?

Điện toán đám mây là việc phân phối các tài nguyên CNTT theo nhu cầu qua Internet với chính sách thanh toán theo mức sử dụng. Thay vì mua, sở hữu và bảo trì các trung tâm dữ liệu và máy chủ vật lý, người dùng có thể tiếp cận các dịch vụ công nghệ, như xử lý tính toán, cơ sở dữ liệu, các dịch vụ IT khi cần thiết, từ nhà cung cấp dịch vụ đám mây như Amazon Web Services (AWS).

* 1. Amazon Virtual Private Cloud (Amazon VPC)

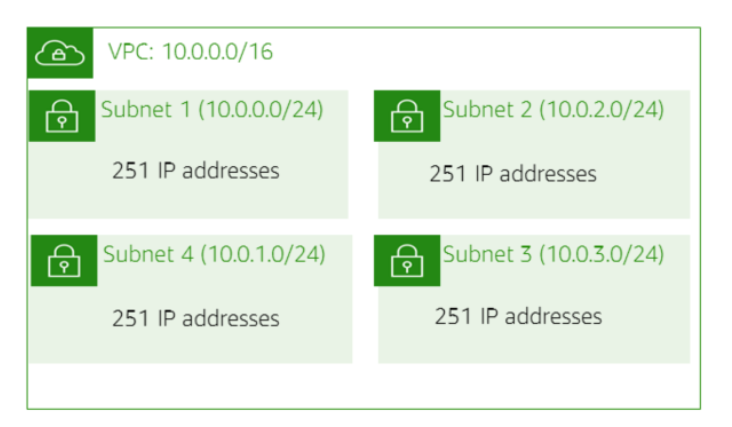


**Virtual Private Cloud** giúp người dùng tạo ra một môi trường mạng riêng ảo nơi mà bạn có thể sử dụng để kết nối nội bộ giữa các dịch vụ **AWS** theo cách bạn quản trị phục vụ cho các mục tiêu bảo mật hệ thống. Người dùng sẽ có toàn quyền quản lý môi trường hệ thống mạng riêng ảo bao gồm : tạo subnet, cấu hình bảng routing, gateway mạng, lựa chọn sử dụng IPv4 hay IPv6 trong VPC.

Bạn dễ dàng tuỳ biến cấu hình mạng đối với **Amazon VPC**. Ví dụ như bạn có thể tạo ra một subnet mạng public dành cho web server để bên ngoài truy cập vào, còn lại bạn cấu hình cho hệ thống backend như database hoặc ứng dụng server nằm trong lớp mạng nội bộ (private) không có đường kết nối Internet public. Bạn còn có thể gia cố thêm nhiều lớp bảo mật cho hệ thống mạng dịch vụ AWS của bạn với : security group, access control list network,… giúp bạn kiểm soát chặt chẽ hơn nữa quyền truy cập vào các Amazon EC2 Instance chẳng hạn.

Một số dịch vụ AWS có thể sử dụng**AWS VPC** như sau :

* Amazon EC2
* Amazon Route 53
* Amazon WorkSpaces
* Auto Scaling
* Elastic Load Balancing
* AWS Data Pipeline
* Elastic Beanstalk
* Amazon Elastic Cache
* Amazon EMR
* Amazon OpsWorks
* Amazon RDS
* Amazon Redshift
  + 1. Subnet



Sau khi tạo VPC, người dùng có thể chia nó thành một hoặc nhiều subnet.

Subnet: là một dải địa chỉ IP trong VPC của bạn. Bạn có thể khởi tạo tài nguyên AWS với một Subnet chỉ định. Ví dụ sử dụng public subnet để các dịch vụ của bạn truy cập được Internet, còn thì sử dụng Private Subnet để các dịch vụ của bạn trong lớp mạng nội bộ bảo mật không truy cập Internet.

* + 1. Route Table

Một bảng định tuyến bao gồm các rule được gọi là ‘route’, các route này sẽ giúp xác định đường đi của lưu lượng mạng ra vào. Mỗi subnet trong VPC của bạn sẽ được liên kết với một bảng định tuyến (route table), bảng định tuyến này sẽ quản lý route trong subnet. Một subnet chỉ có thể liên kế với 1 bảng định tuyến tại 1 thời điểm, nhưng chiều ngược lại bạn có thể liên kết nhiều subnet với 1 bảng định tuyến.

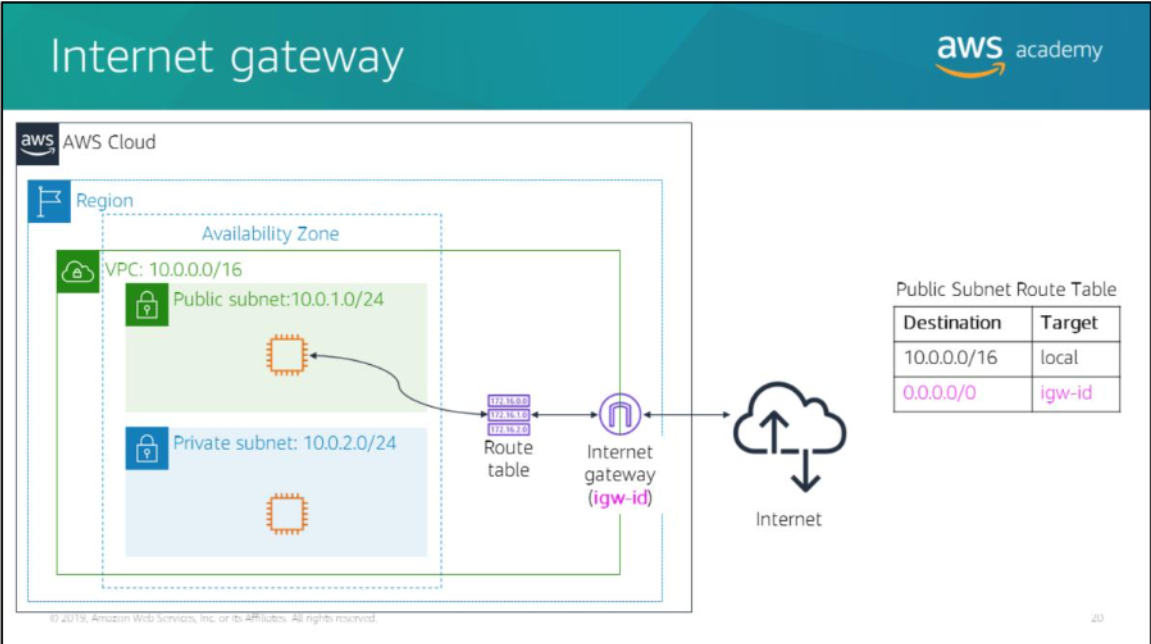


Destination là đích đến của lưu lượng mạng từ subnet, Dest có thể là 1 dải CIDR / IP address

Target là nơi gói tin cần đến trước khi đến Dest

* + 1. Internet Gateway

Internet Gateway cho phép bạn tạo một subnet public có route kết nối ra Internet public.



Cổng internet phục vụ hai mục đích: cung cấp target trong bảng route table VPC để có thể định tuyến internet và thực hiện dịch địa chỉ mạng (Nat) cho các instances được chỉ định địa chỉ IPv4 công cộng (public subnet)

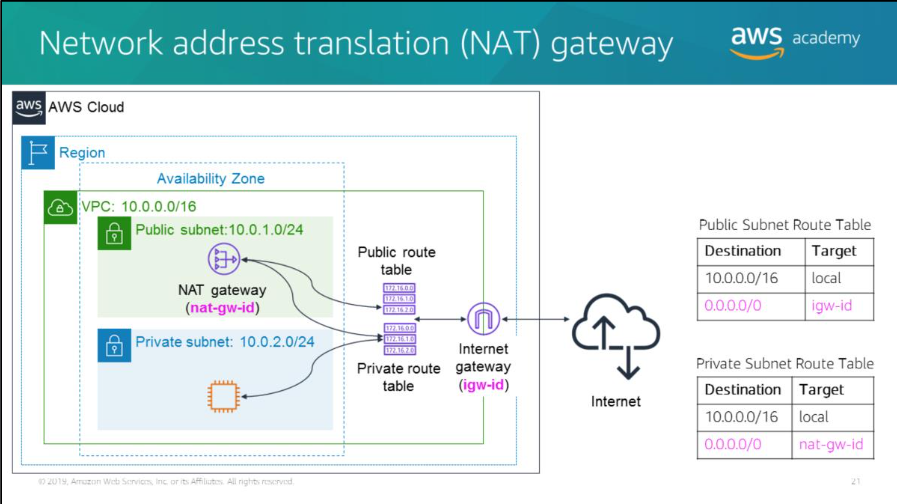
Internet Gateway cung cấp một cách hợp lý NAT thay mặt cho instances, để khi lưu lượng truy cập rời khỏi mạng con VPC của bạn và đi đến internet, trường địa chỉ trả lời được đặt thành public IPv4 address hoặc Elastic IP của instances và không phải private IP address của instances

* + 1. Network address translation (NAT)

NAT gateway cho phép instances ở private subnet kết nối với internet hoặc các dịch vụ AWS, nhưng ngăn không cho internet kết nối tới instances này

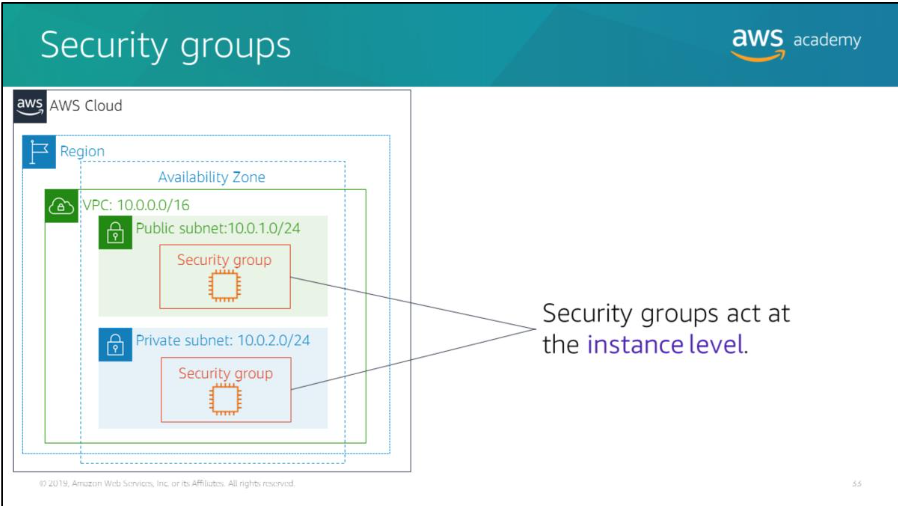
NAT gateway được đặt trong public subnet để có thể kết nối internet

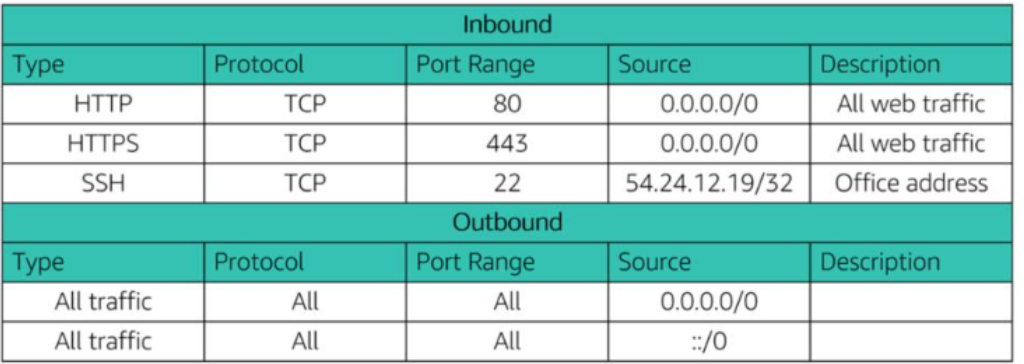
Thiết lập lại route table(0.0.0.0/0, nat) để private subnet có thể kết nối tới internet



* + 1. Security Group

Là lớp bảo mật đầu tiên dành cho các Instance, hoạt động như một firewall ảo và bạn sẽ phải định nghĩa rule firewall trước khi traffic ra vào Instance.





Cho phép HTTP, HTTPS, SSH ngoài (inbound traffic) vào Instance.

Cho phép tất cả traffic trong (outbound traffic) Instnace đi ra.

Mặc định khi khởi tạo, Inbound không có rule nào cả và Outbound thì cho phép tất cả lưu lượng ra khỏi instance => không cho phép bất cứ lưu lượng nào được truy cập vào trong.

* Cho phép tất cả traffic ngoài (inbound traffic) vào Instance.
* Cho phép tất cả traffic trong (outbound traffic) Instnace đi ra.
  1. DynamoDB
* **DynamoDB** là môt dịch vụ quản lý **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** 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 bơi khách hàng và lượng dữ liệu lưu trữ, và đồng thời đảm bảo hiệu suất nhanh và đồng nhất. Tất cả dự liệu được lưu trữ trên SSD và tự động được sao chép ra các vùng sẵn sằng (***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***)
* Nếu bạn là một nhà quản lý dữ liệu, bạn có thể tạo một bảng dữ liệu mới, mở rộng hay thu hẹp request capacity mà không bị giảm hiệu suất, và có thể thây được các thông số qua **AWS Management Console**. Với **DynamoDB**, bạn có thể phó thác gánh nặng quản lý và mở rộng dữ liệu cho AWS và không phải lo lắng về việc cung cấp hardware, thiết lập và cài đặt, sao chép dữ liệu.
  + 1. Các thành phần DynamoDB cơ bản:

1. **Tables** - Tương tự như các hệ thống cơ sở dữ liệu khác, DynamoDB lưu trữ dữ liệu trong các bảng. Một bảng là một tập hợp các dữ liệu. Ví dụ: hãy xem bảng ví dụ có tên Mọi người mà bạn có thể sử dụng để lưu trữ thông tin liên hệ cá nhân về bạn bè, gia đình hoặc bất kỳ ai khác mà bạn quan tâm. Bạn cũng có thể có bảng Ô tô để lưu trữ thông tin về các phương tiện mà mọi người lái.
2. **Items**  - Mỗi table chứa không hoặc nhiều Items. Một item là một nhóm các thuộc tính đó là nhận dạng duy nhất trong số tất cả các mặt hàng khác. Các item trong DynamoDB tương tự như rows, records hoặc tuples trong các hệ thống cơ sở dữ liệu khác. Trong DynamoDB, không có giới hạn về số lượng mục bạn có thể lưu trữ trong một bảng.
3. **Attributes** - Mỗi Item bao gồm một hoặc nhiều **Attributes**. Một **Attributes** là một yếu tố dữ liệu cơ bản, một cái gì đó mà không cần phải được chia nhỏ thêm nữa. Các **Attributes** trong DynamoDB tương tự với các trường hoặc cột trong các hệ thống cơ sở dữ liệu khác.
4. **Primary Key**: Khi tạo một bảng, thì ngoài việc khai báo tên bảng bạn cần chỉ rõ primary key . Primary key để đảm bảo rằng không có hai items nào trong bảng có cùng primary key. DynamoDB hỗ trợ hai loại primary key sau:
5. **Partition key:**

* Một khóa chính đơn giản, bao gồm một thuộc tính được gọi là Partition key.
* DynamoDB sử dụng giá trị của Partition key làm đầu vào cho Hash function bên trong. Đầu ra từ Hash function xác định phân vùng (bộ nhớ vật lý bên trong DynamoDB) mà mục sẽ được lưu trữ.
* Trong bảng chỉ có Partition key, không có hai mục nào có thể có cùng giá trị Partition key

1. **Partition key and sort key:**

* Được gọi là Primary Key tổng hợp , loại khóa này bao gồm hai thuộc tính. Thuộc tính đầu tiên là Partition key và thuộc tính thứ hai là sort key
* DynamoDB sử dụng giá trị Partition key làm đầu vào cho hàm Hash bên trong. Đầu ra từ hàm Hash xác định phân vùng (bộ nhớ vật lý bên trong DynamoDB) mà mục sẽ được lưu trữ. Tất cả các mục có cùng giá trị Partition key được lưu trữ cùng nhau, theo thứ tự được sắp xếp theo giá trị sort key.
* Trong bảng có Partition key và sort key, hai mục có thể có cùng giá trị Partition key. Tuy nhiên, hai mục đó phải có giá trị sort key khác nhau.

1. **Secondary index:**

Để giúp cho việc truy cập dữ liệu, Amazon DynamoDB tạo và quản lý index cho primary key. Việc này giúp cho hệ thống truy cập dữ liệu khá nhanh khi chỉ định primary key. Tuy nhiên, nhiều hệ thống có lẽ sẽ hưởng lợi từ việc có một hoặc hai key thay thế (alternate). Để tăng tốc độ truy cập bạn có thể tạo ra secondary index.

Một secondary index là một cấu trúc dữ liệu chứa tập con của các attributes trong một bảng, cùng với một key thay thế để hỗ trợ cho Query operation. Với một secondary index, các query sẽ không bị giới hạn khi chỉ sử dụng được primary key; bạn có thể truy vấn dữ liệu bằng việc sử dụng key thay thế từ secondary index. Một bảng có thể có nhiều secondary index để hỗ trợ cho nhiều query pattern.

DynamoDB hỗ trợ hai loại index:

* Global secondary index: một loại index có hash và range key khác với hash và range key từ bảng gốc.
* Local secondary index: một loại index mà có hash key trùng với hash key từ bảng gốc và range key là một attribute khác.

1. **DynamoDB Streams:**

DynamoDB Streams là một tính năng tùy chọn ghi lại các sự kiện sửa đổi dữ liệu trong bảng DynamoDB. Dữ liệu về các sự kiện này xuất hiện trong luồng theo thời gian gần thực và theo thứ tự các sự kiện đã xảy ra.

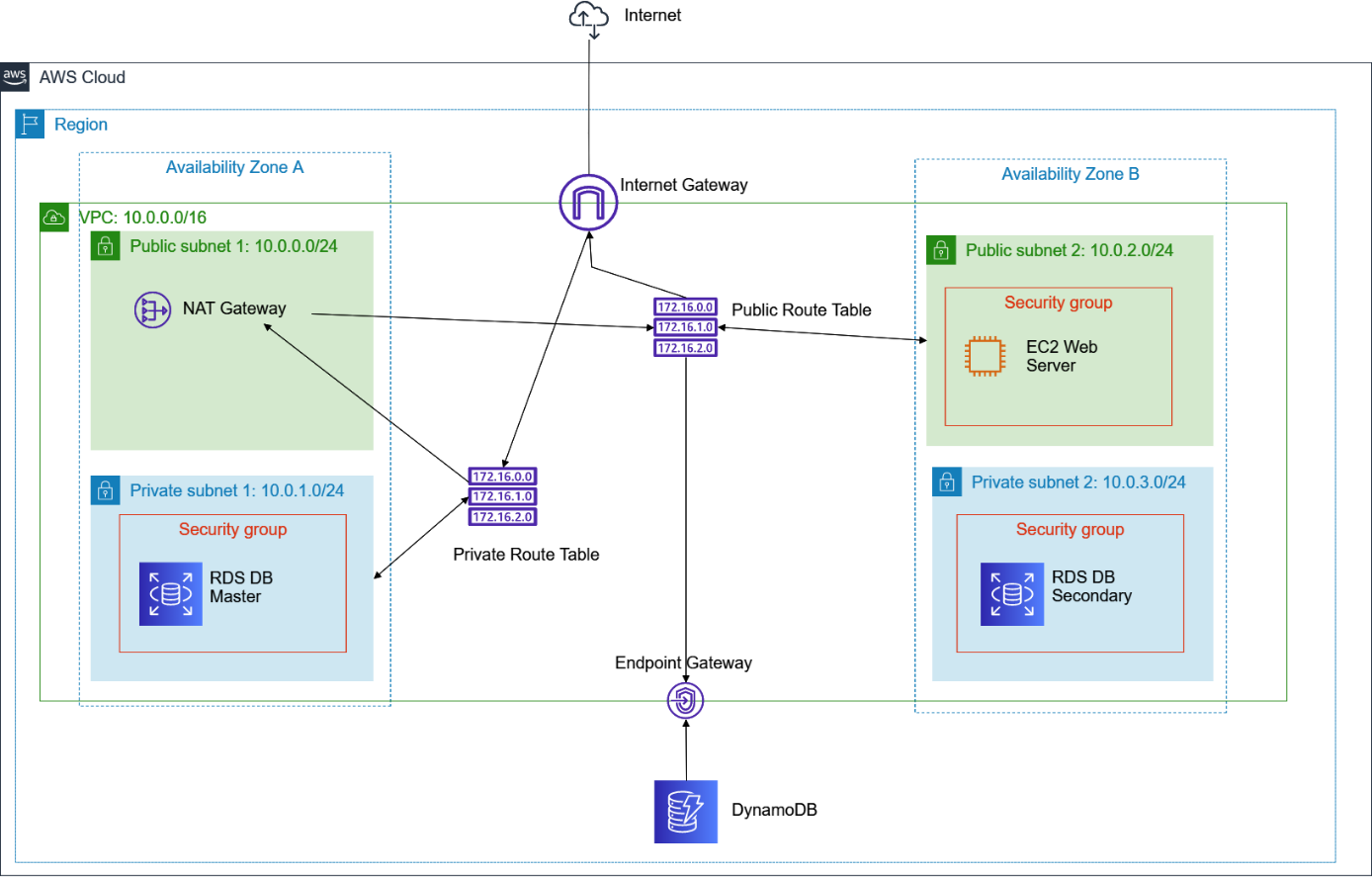
Mỗi sự kiện được đại diện bởi một *bản ghi luồng* . Nếu bạn bật luồng trên bảng, DynamoDB Streams sẽ ghi một bản ghi luồng bất cứ khi nào một trong các sự kiện sau xảy ra:

* Một mục mới được thêm vào bảng: Luồng chụp ảnh của toàn bộ mục, bao gồm tất cả các thuộc tính của nó.
* Một mặt hàng được cập nhật: Luồng ghi lại hình ảnh "trước" và "sau" của bất kỳ thuộc tính nào đã được sửa đổi trong mặt hàng.
* Mục bị xóa khỏi bảng: Luồng chụp ảnh của toàn bộ mục trước khi bị xóa.

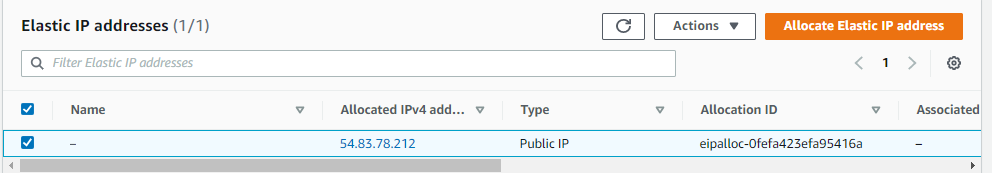
Mỗi bản ghi luồng cũng chứa tên của bảng, dấu thời gian sự kiện và siêu dữ liệu khác. Bản ghi luồng có thời gian tồn tại là 24 giờ; sau đó, chúng tự động bị xóa khỏi luồng.

Bạn có thể sử dụng DynamoDB Streams cùng với AWS Lambda để tạo trình *kích hoạt* —mã tự động chạy bất cứ khi nào một sự kiện quan tâm xuất hiện trong một luồng

1. Phần demo
2. **Project 1 – Trang web quản lý khoá học PHP - MySQL**

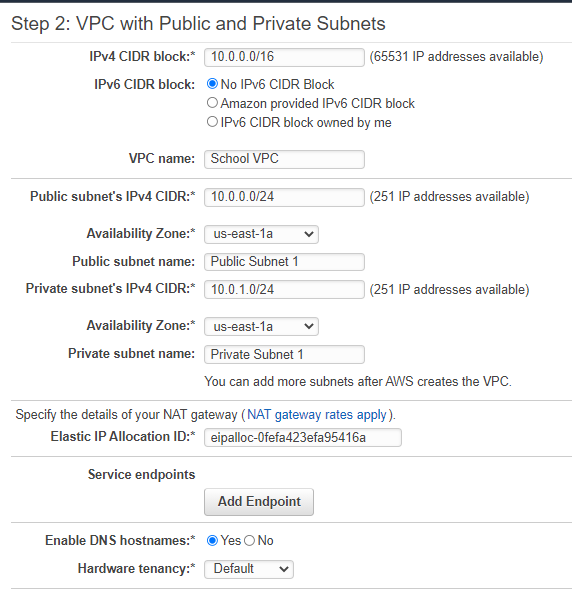


* 1. VPC
     1. ElasticIP – Tạo Public IP để kết nối tới máy chủ EC2

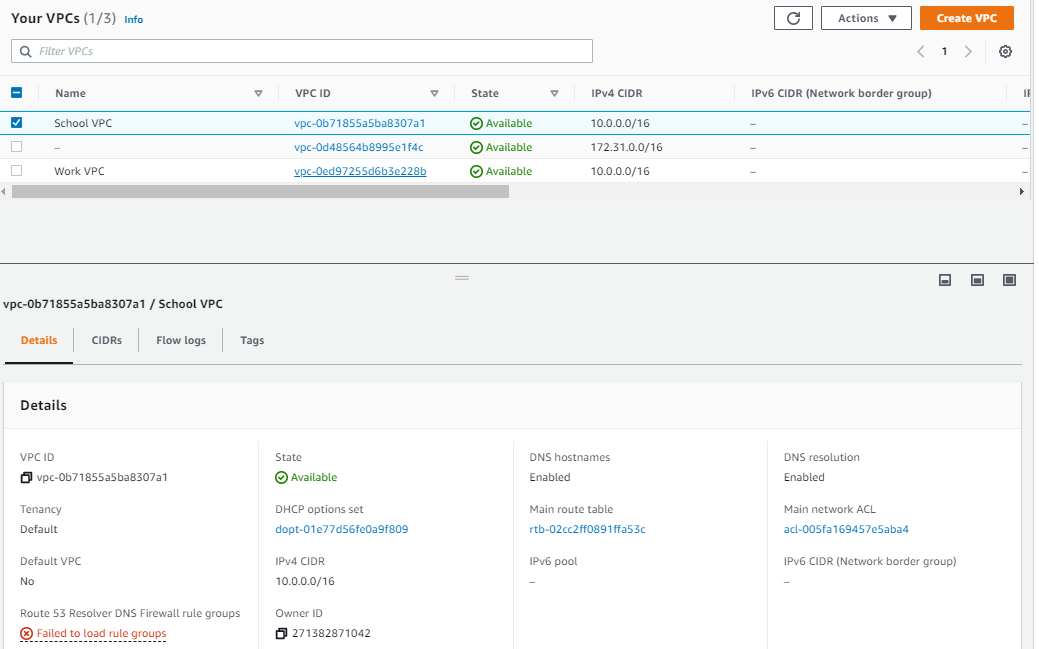


* + 1. Tạo VPC có Public Subnet 1 và Private Subnet 1

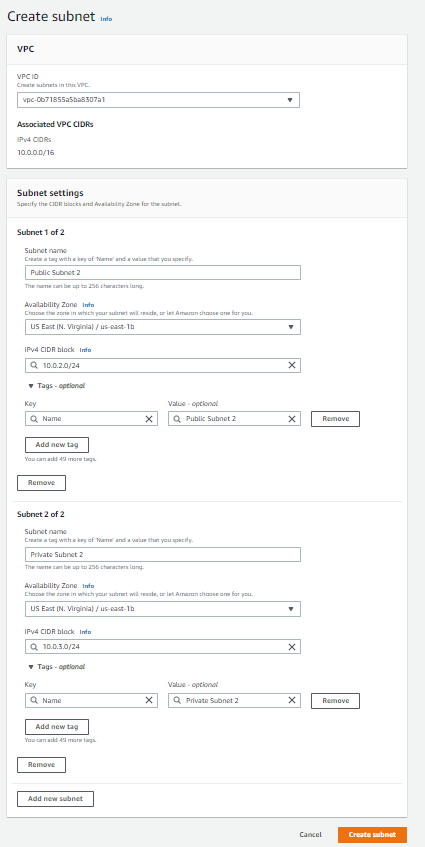
School VPC



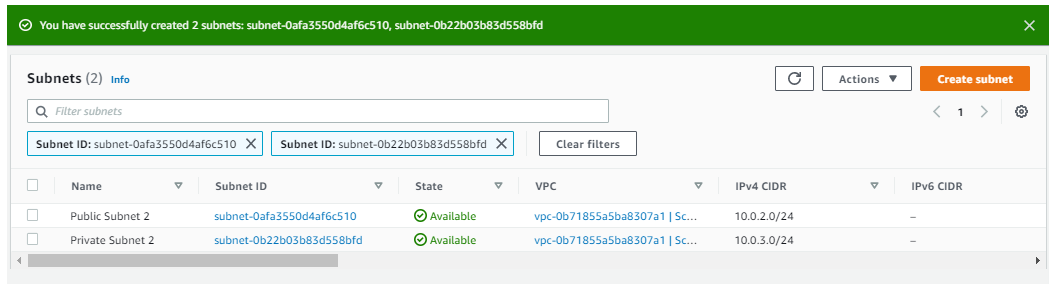
Kết quả



* + 1. Tạo Public Subnet 2 và Private Subnet 2



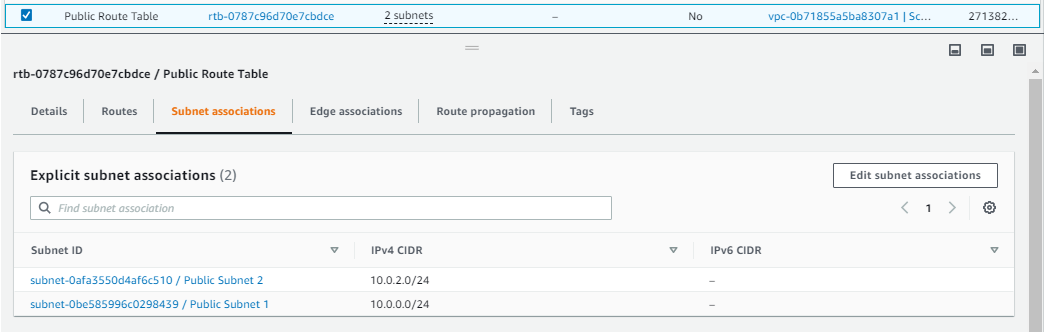
Kết quả



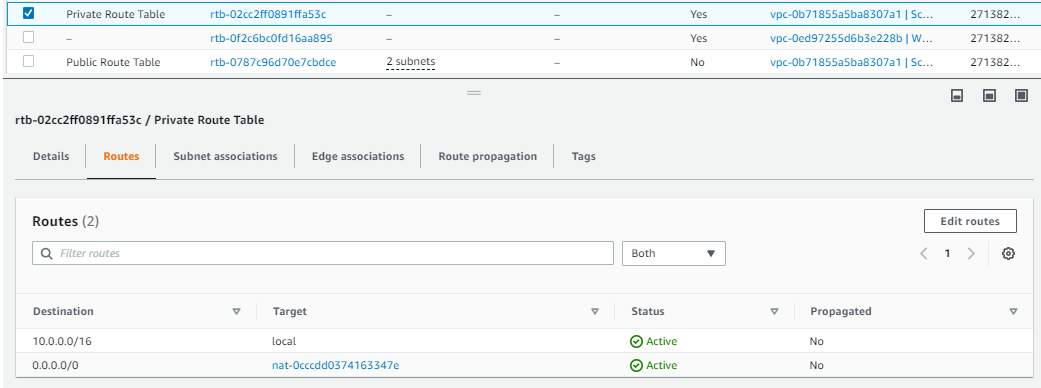
* + 1. Route Table

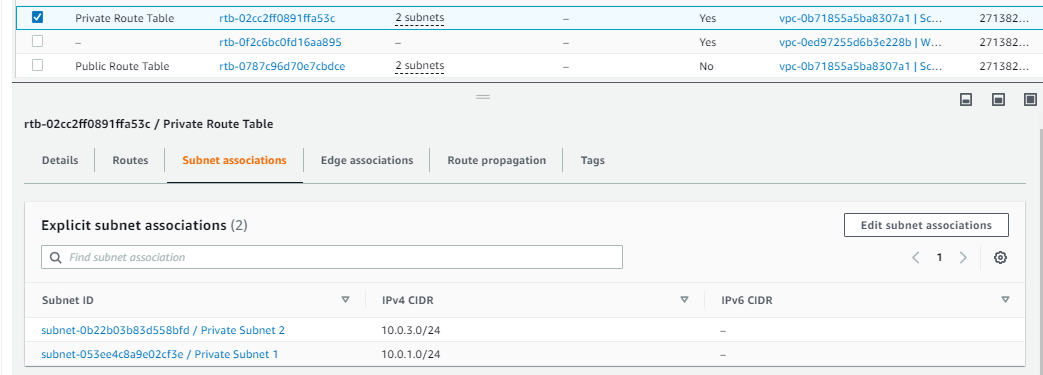
Thiết lập Route Table cho Public Subnet





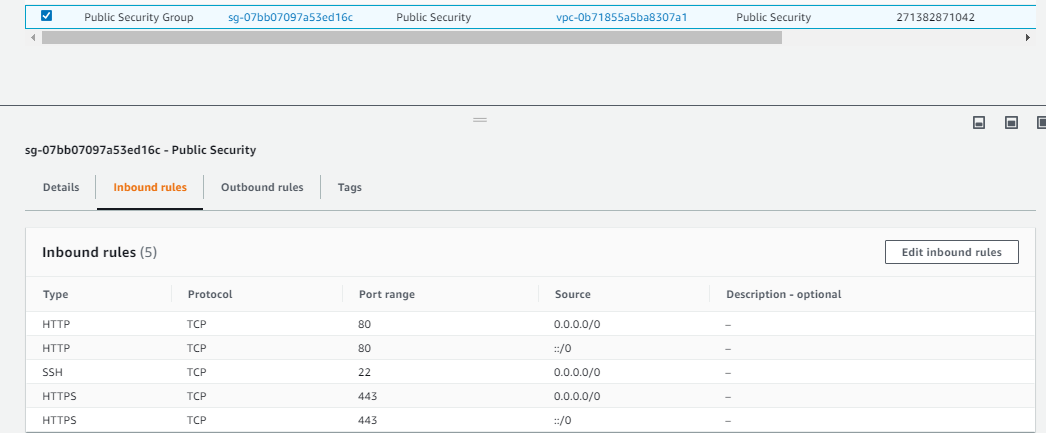
Thiết lập Route Table cho Private Subnet



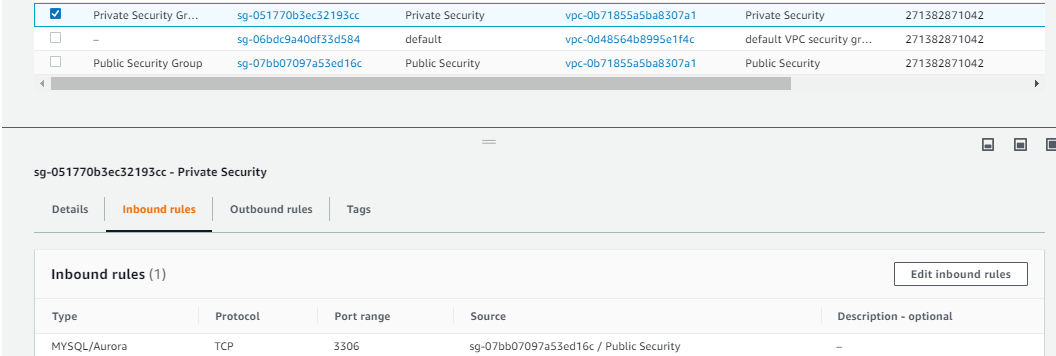


* + 1. Security Group

Thiết lập Security Group cho Web Server EC2

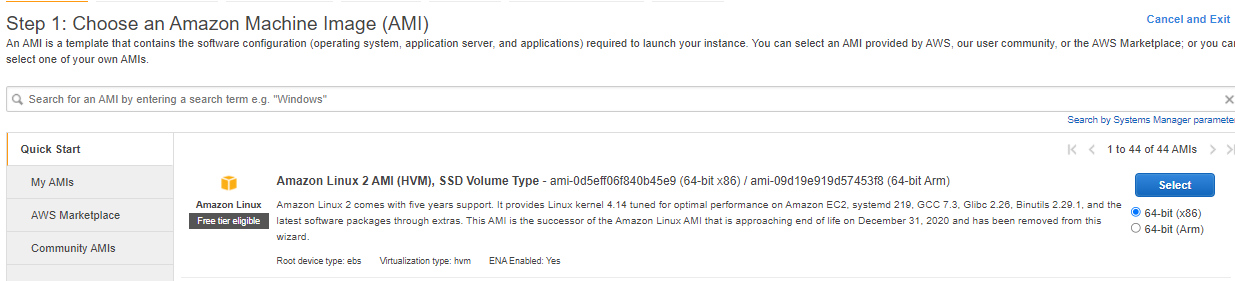


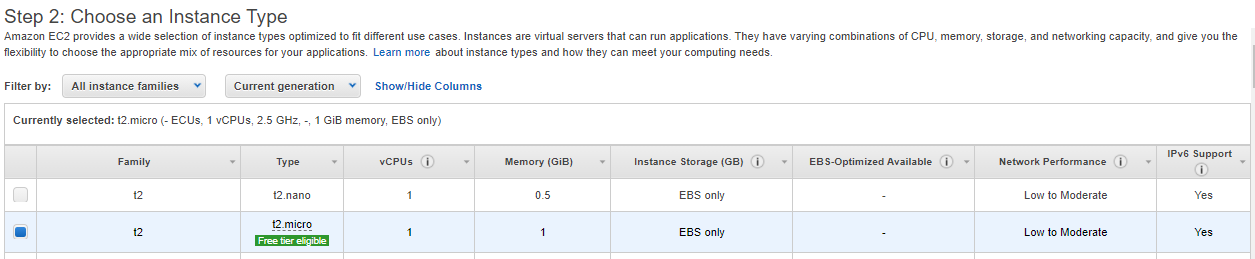
Thiết lập Security Group cho Database Server



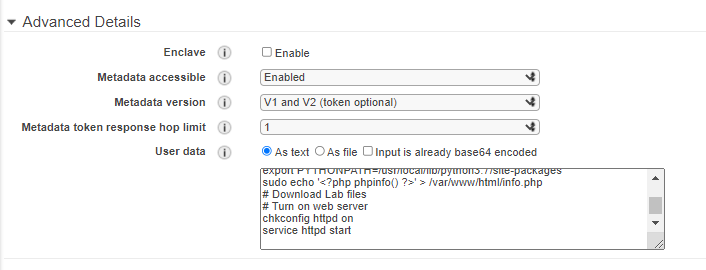
* + 1. Tạo EC2 Web Server

Chọn Amazon Linux 2 AMI









#!/bin/bash

# Install Apache Web Server and PHP

yum install -y httpd git

amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2

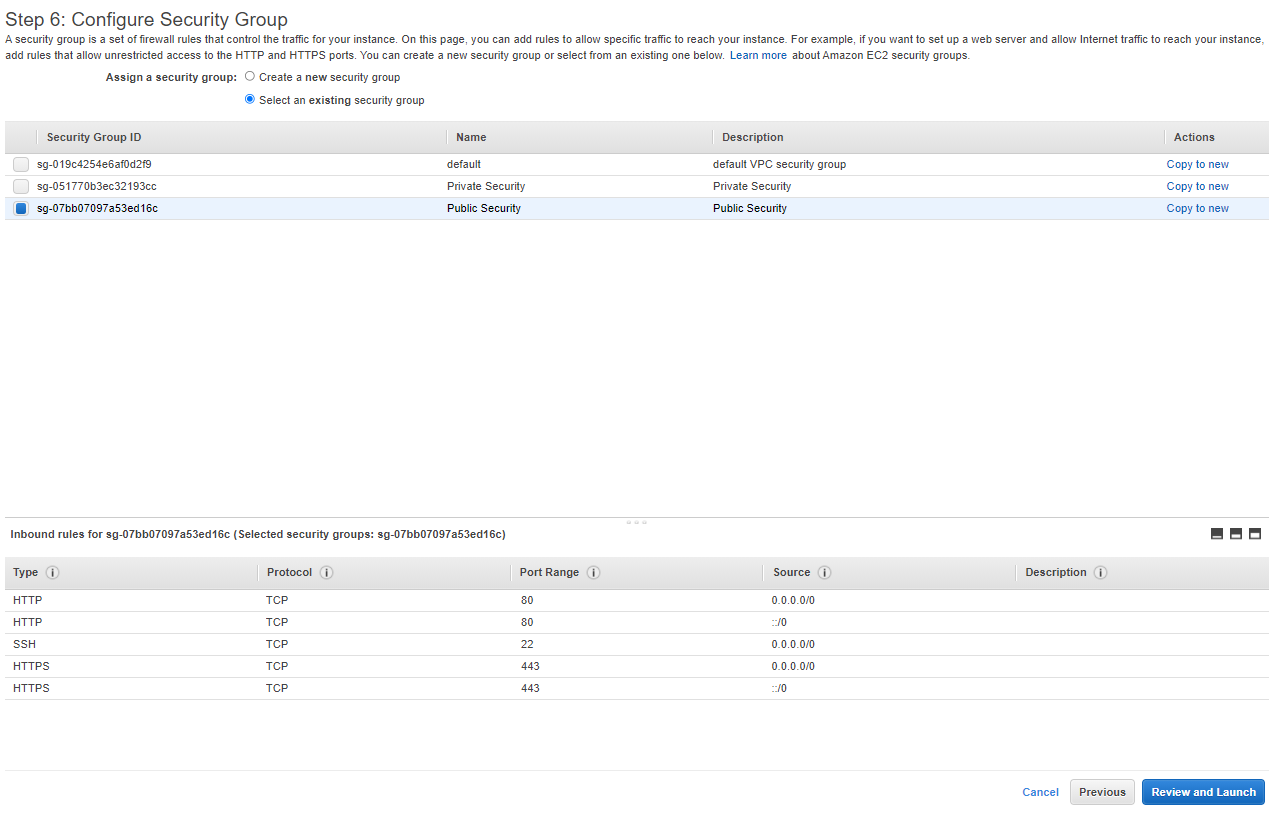
sudo echo '<?php phpinfo() ?>' > /var/www/html/info.php

# Download Lab files

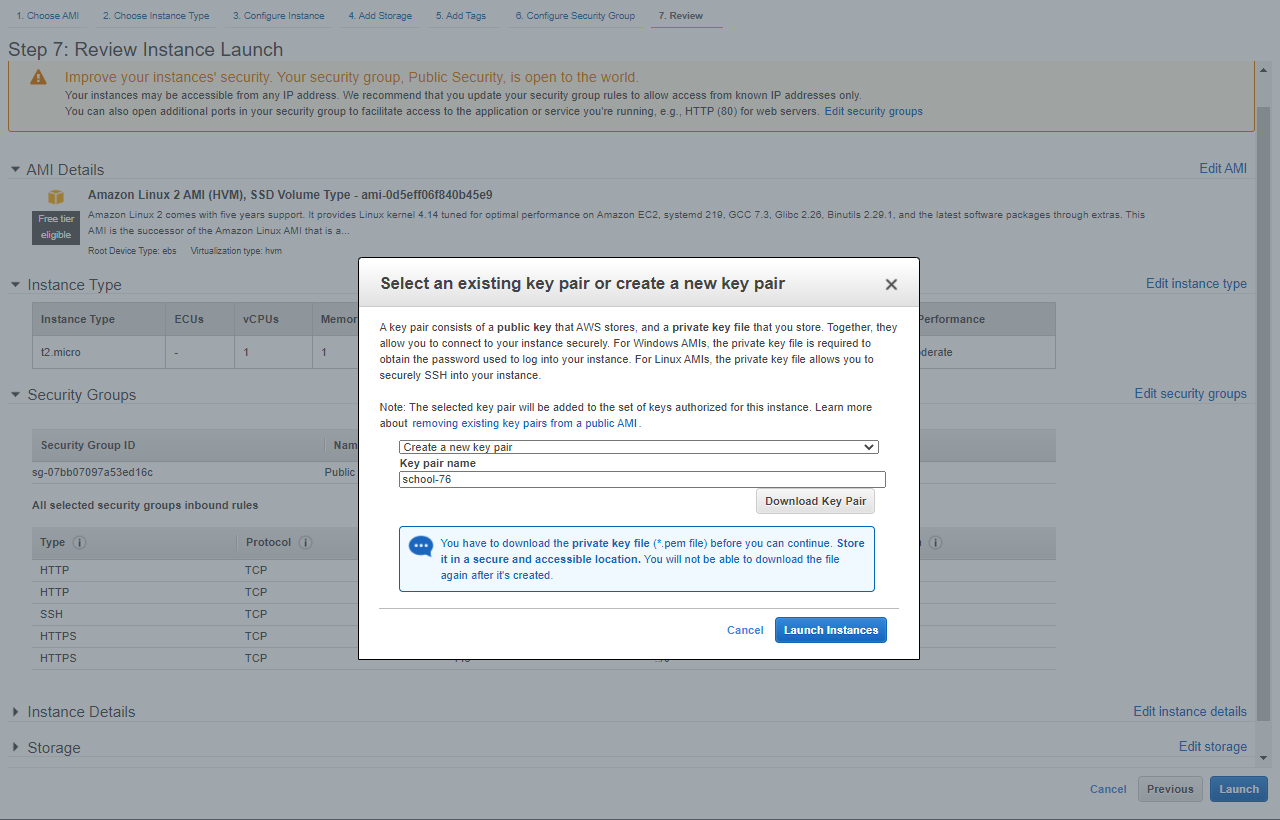
# Turn on web server

chkconfig httpd on

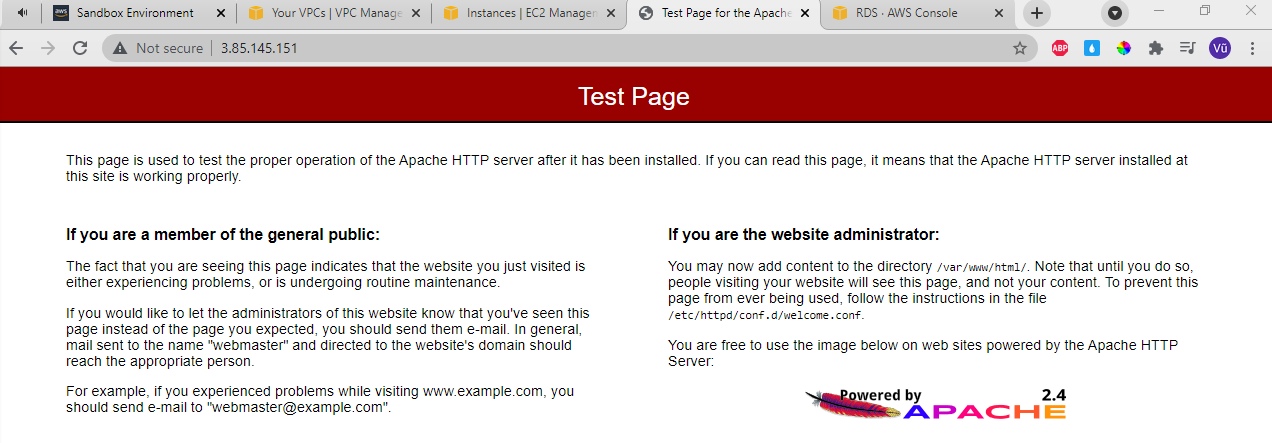
service httpd start

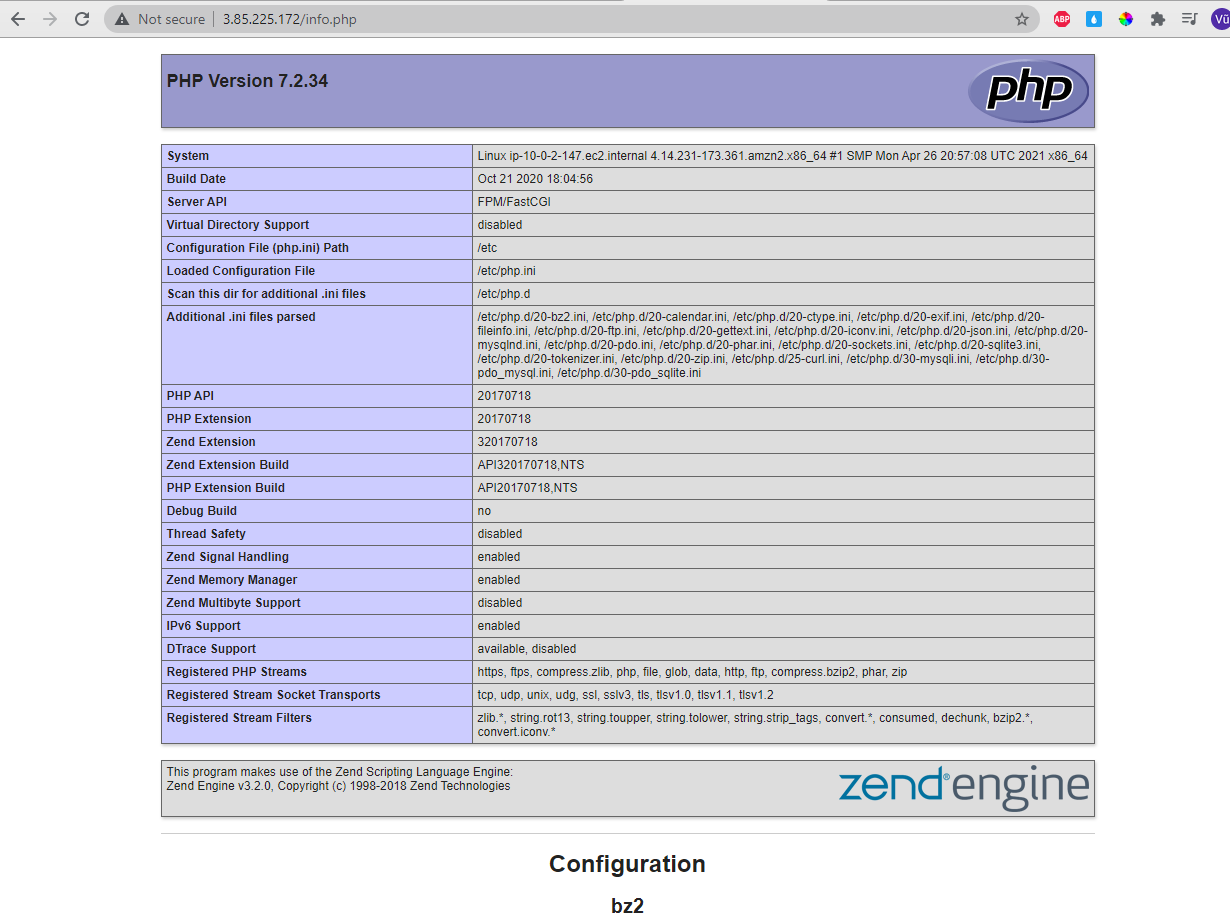


Download Key Pair để kết nối SSH



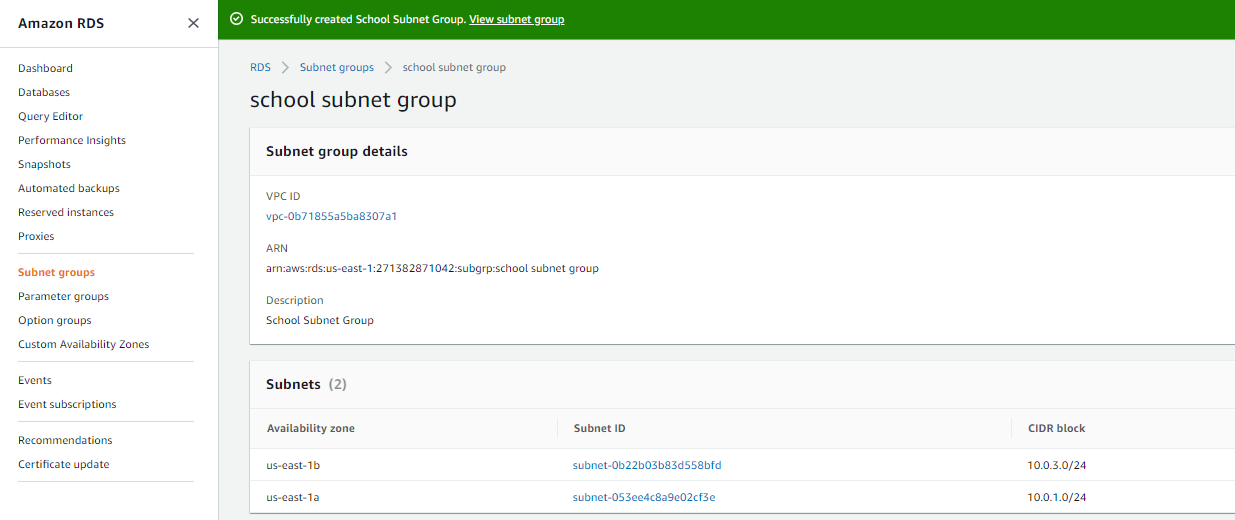
Test máy chủ EC2



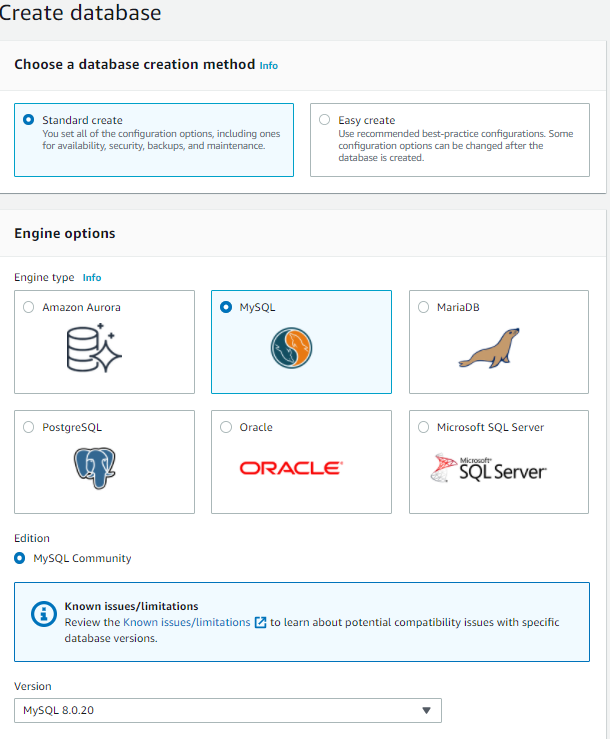


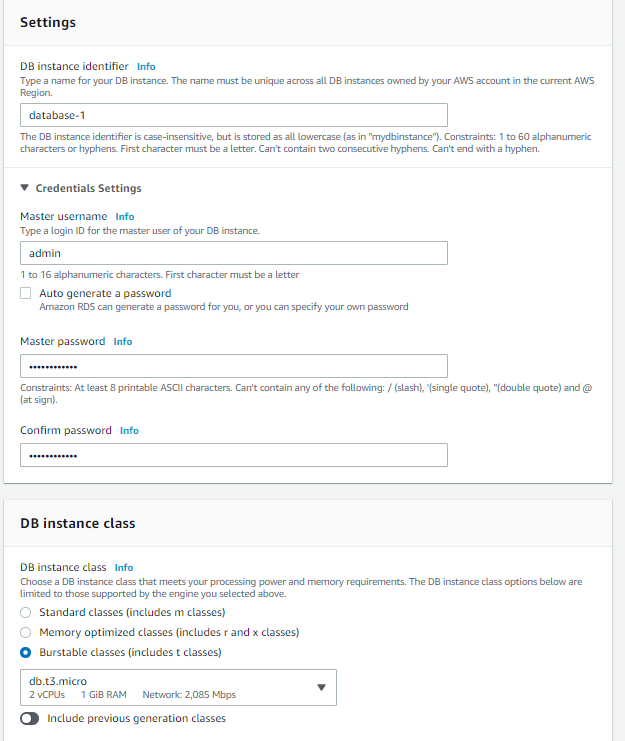
* + 1. Tạo RDS Database Server

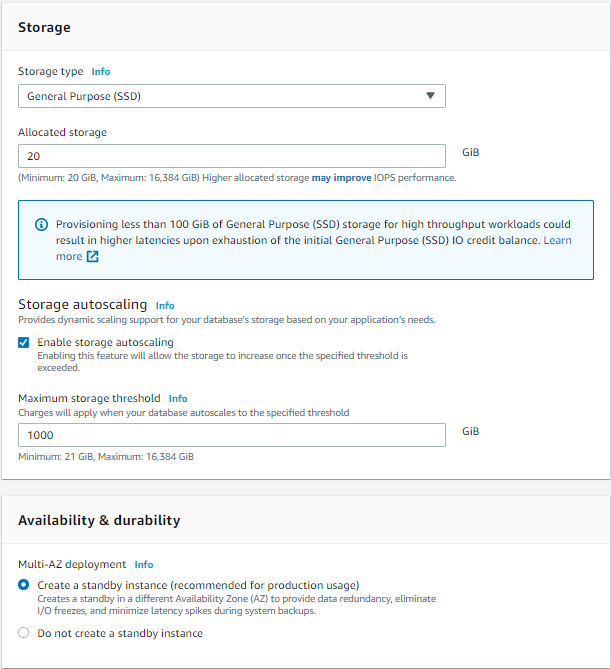
Tạo Subnet Group

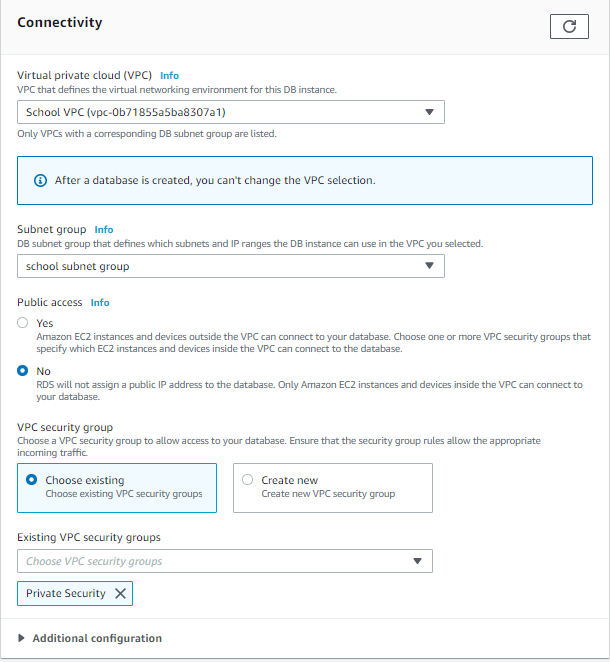


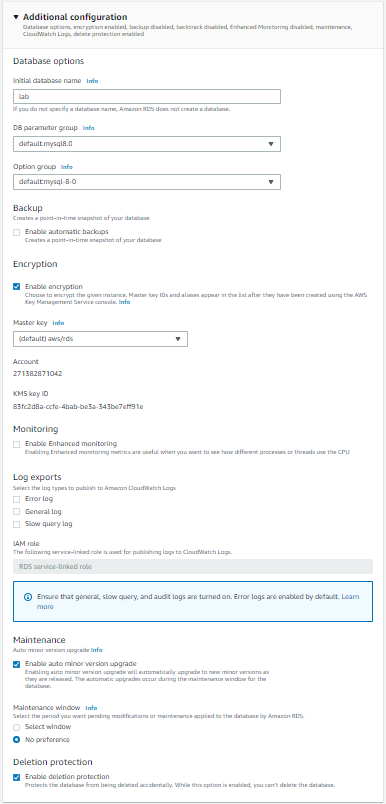
Tạo Database MySQL

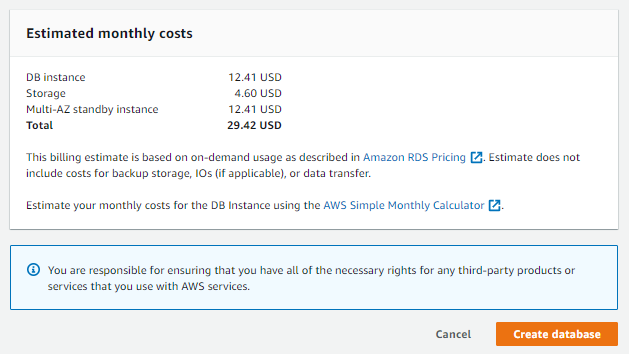




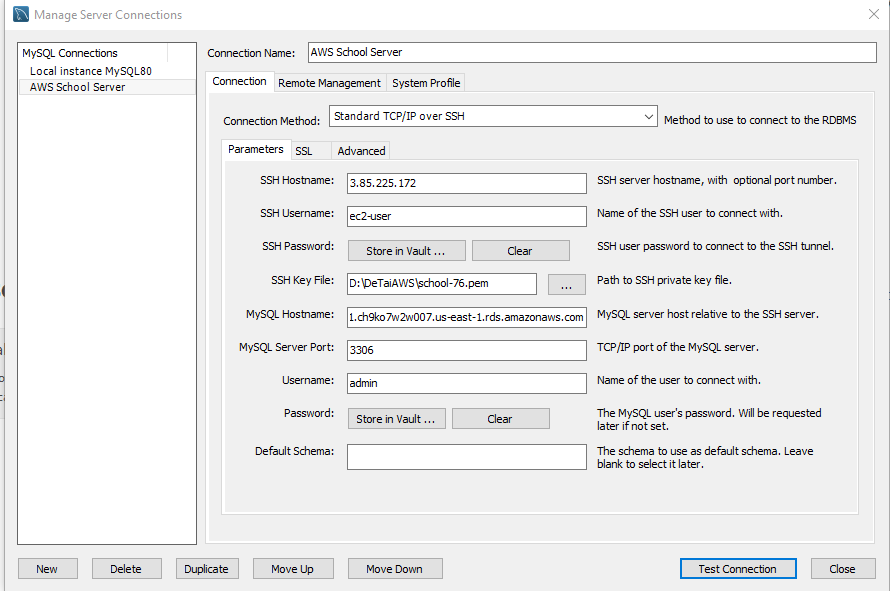




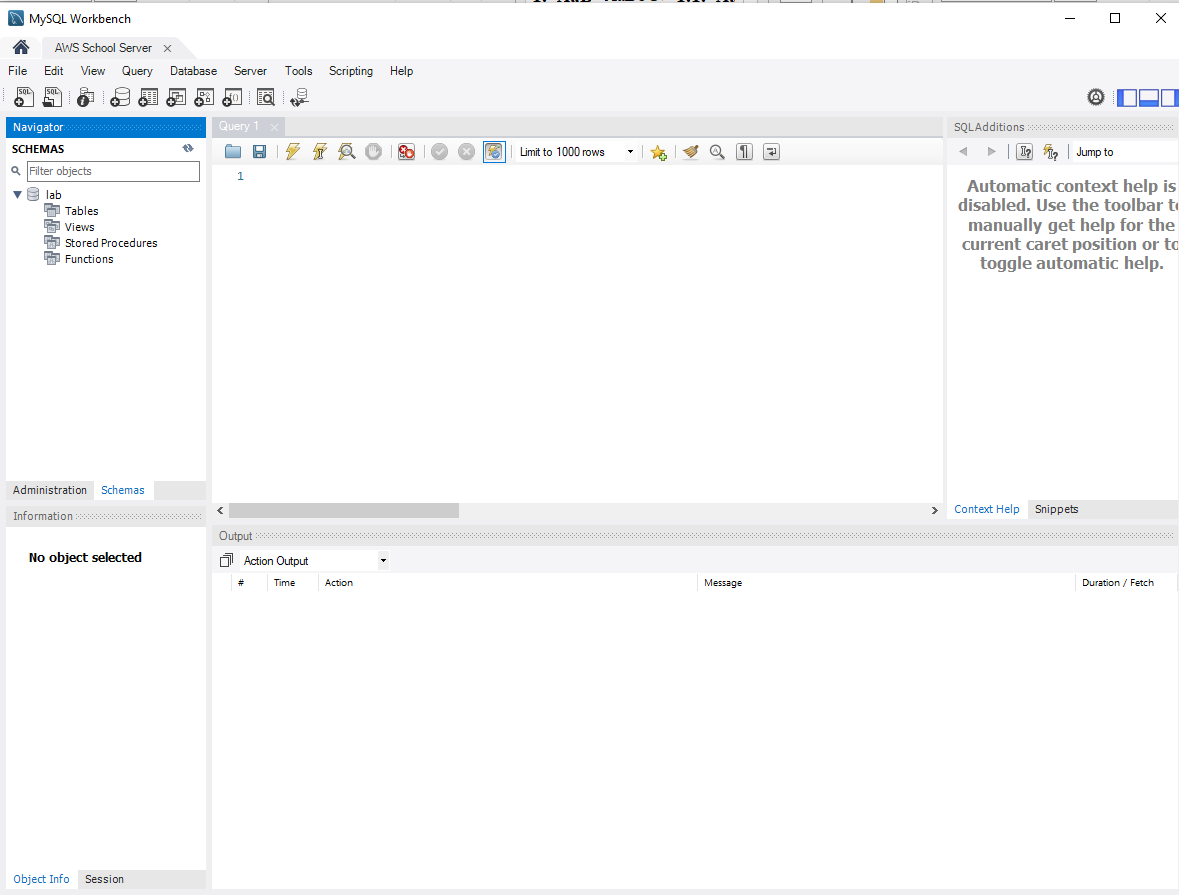




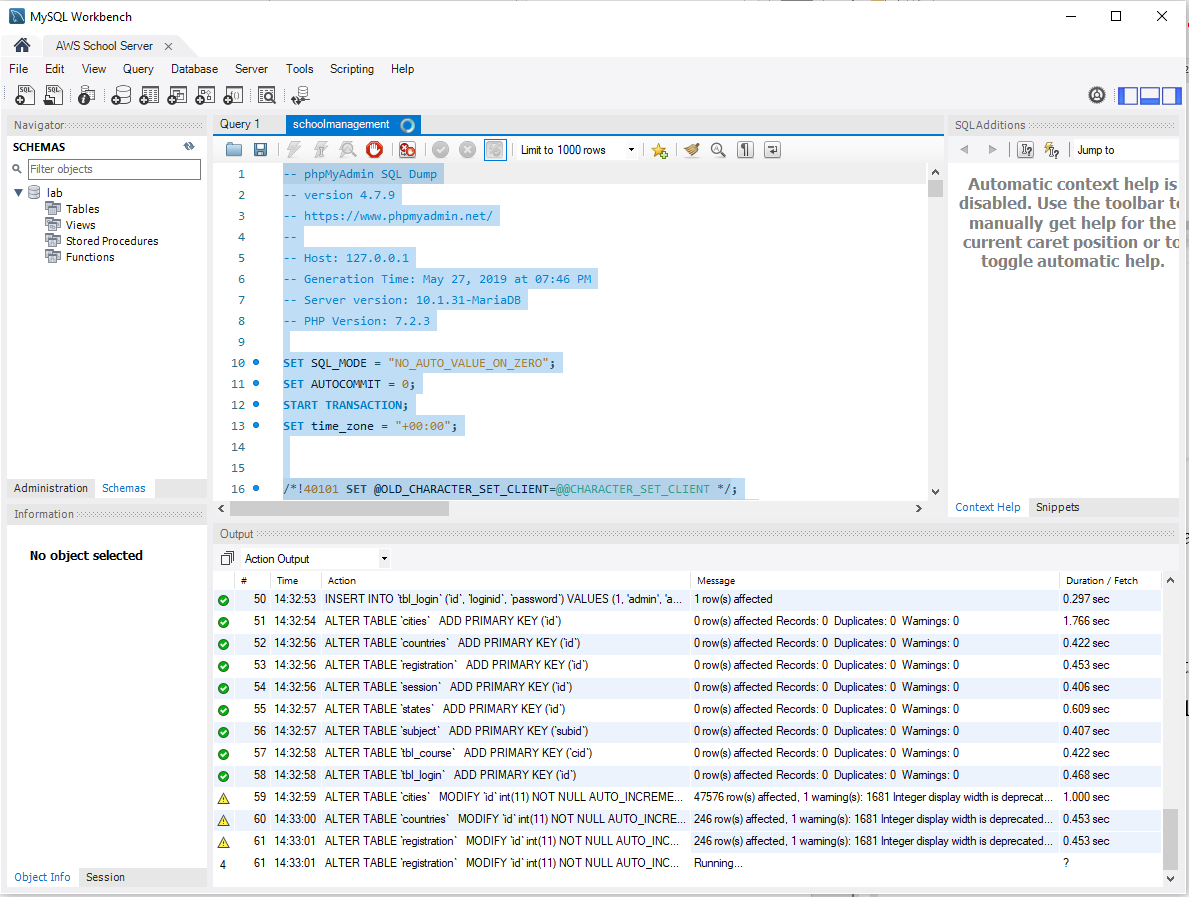
* + 1. Sử dụng MySQL Workbrench, dùng SSH kết nối EC2 và MySQL Server



Kết nối thành công



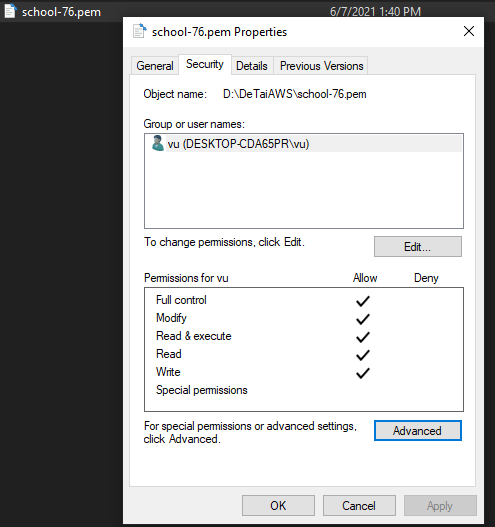
* + 1. Tạo RDS MySQL Database schoolmanagement



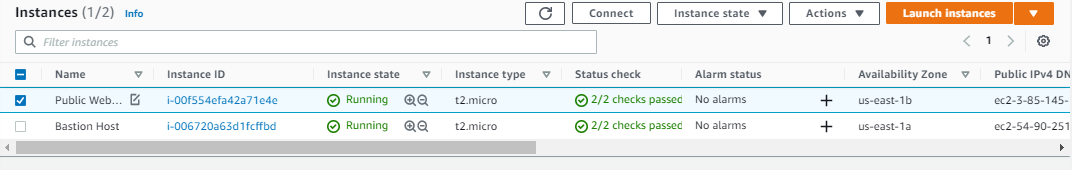


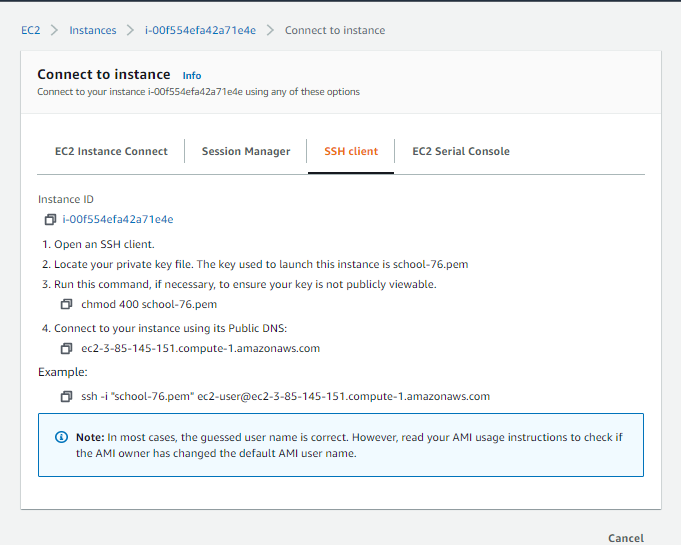
* + 1. Kết nối SSH bằng CMD truy cập vào EC2

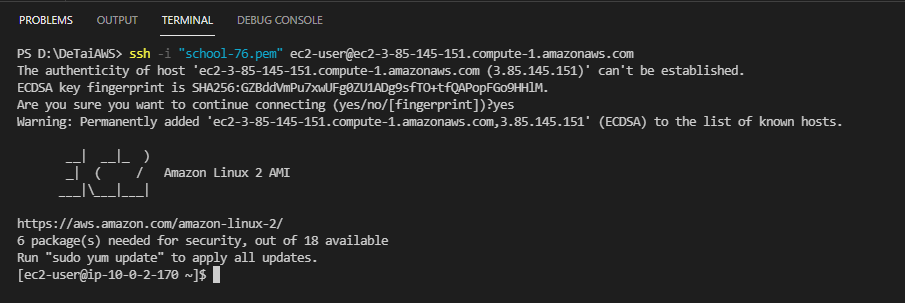
Chuyển quyền sử dụng file key pair thành user windows đang sử dụng



Click vào connect trong trang EC2







Thiết lập quyền sử dụng folder /var/www

1. Add your user (in this case, ec2-user) to the apache group.

[ec2-user ~]$ **sudo usermod -a -G apache** *ec2-user*

1. Log out and then log back in again to pick up the new group, and then verify your membership.
   1. Log out (use the **exit** command or close the terminal window):

[ec2-user ~]$ **exit**

* 1. To verify your membership in the apache group, reconnect to your instance, and then run the following command:
  2. [ec2-user ~]$ **groups**

ec2-user adm wheel apache systemd-journal

1. Change the group ownership of /var/www and its contents to the apache group.

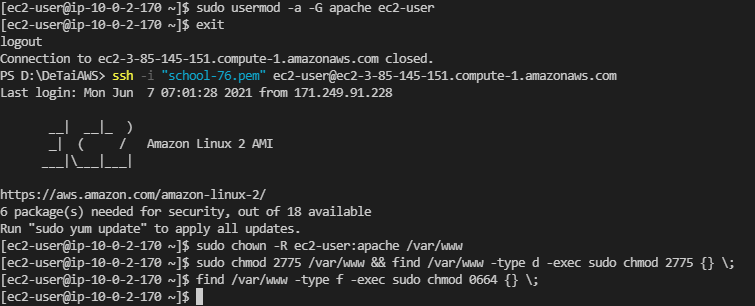
[ec2-user ~]$ **sudo chown -R ec2-user:apache /var/www**

1. To add group write permissions and to set the group ID on future subdirectories, change the directory permissions of /var/www and its subdirectories.

[ec2-user ~]$ **sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;**

1. To add group write permissions, recursively change the file permissions of /var/www and its subdirectories:

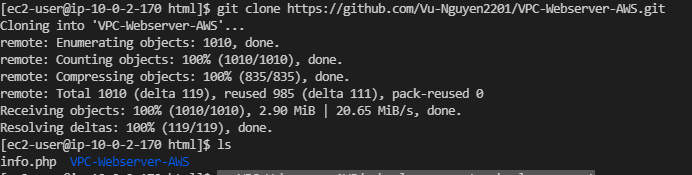
[ec2-user ~]$ **find /var/www -type f -exec sudo chmod 0664 {} \;**



* + 1. Download Project School PHP – MySQL

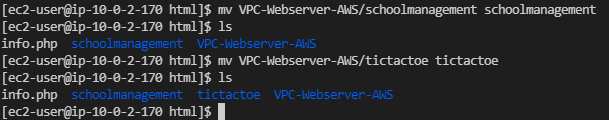
Dùng git để Download Project School PHP – MySQL (Project không dùng cho mục đích thương mại, chỉ dùng cho mục đích giáo dục, nghiên cứu)

git clone https://github.com/Vu-Nguyen2201/VPC-Webserver-AWS.git

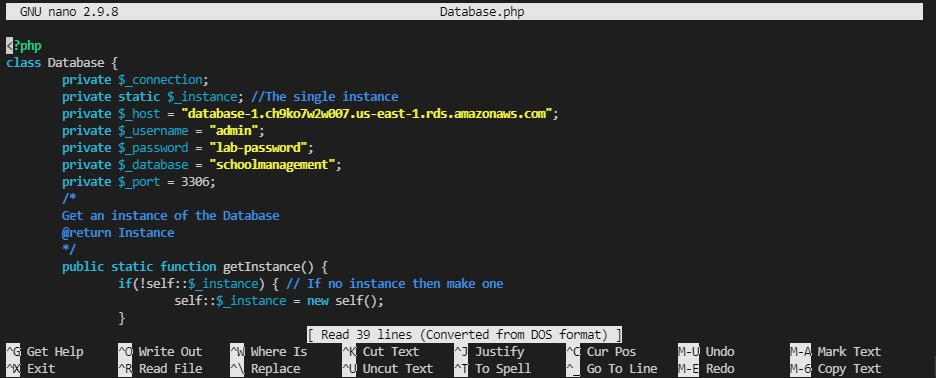


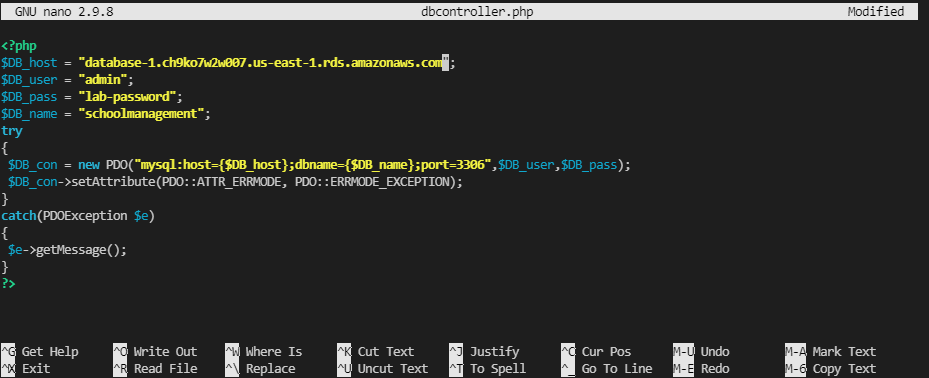
mv VPC-Webserver-AWS/schoolmanagement schoolmanagement

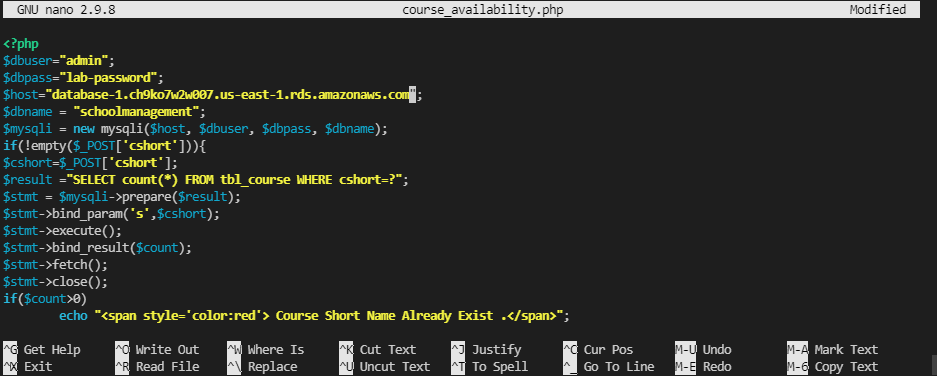
mv VPC-Webserver-AWS/tictactoe tictactoe



Chỉnh lại hostname, username, password, database

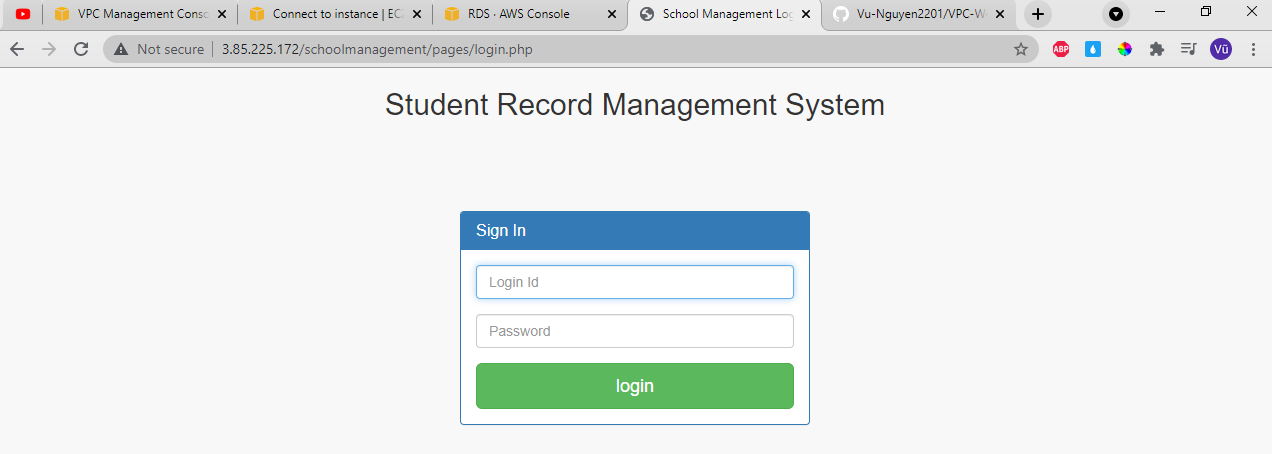




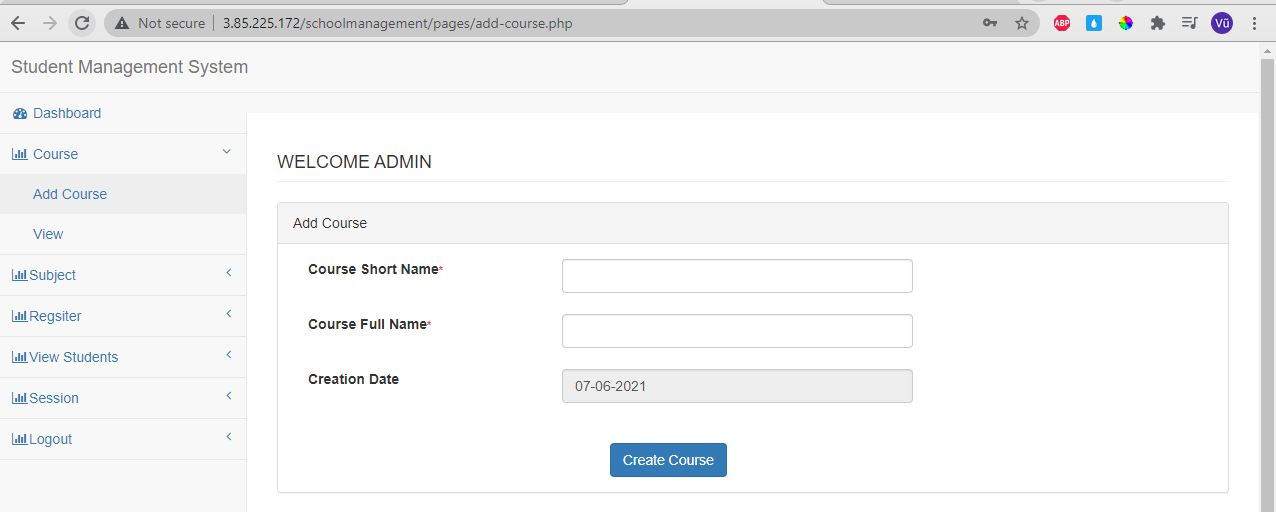


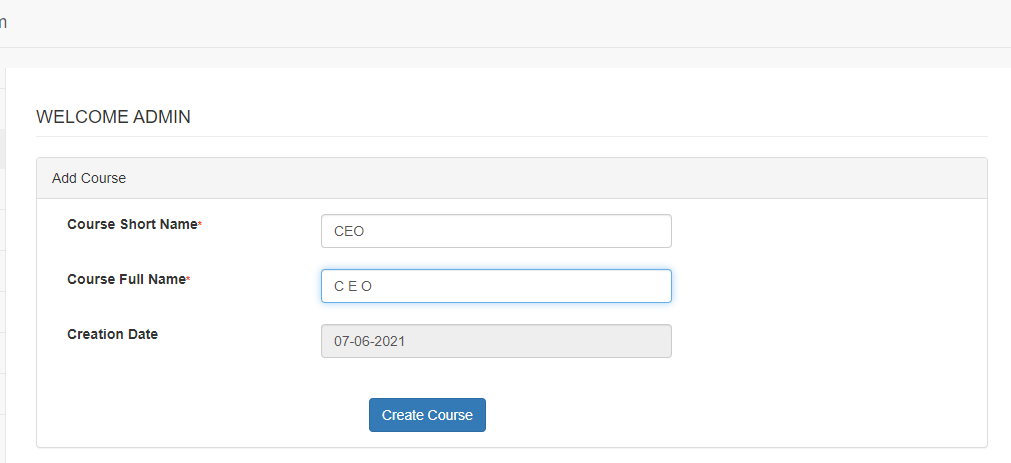
* + 1. Kết quả Demo

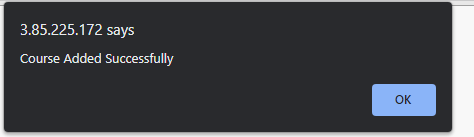
Login admin/admin

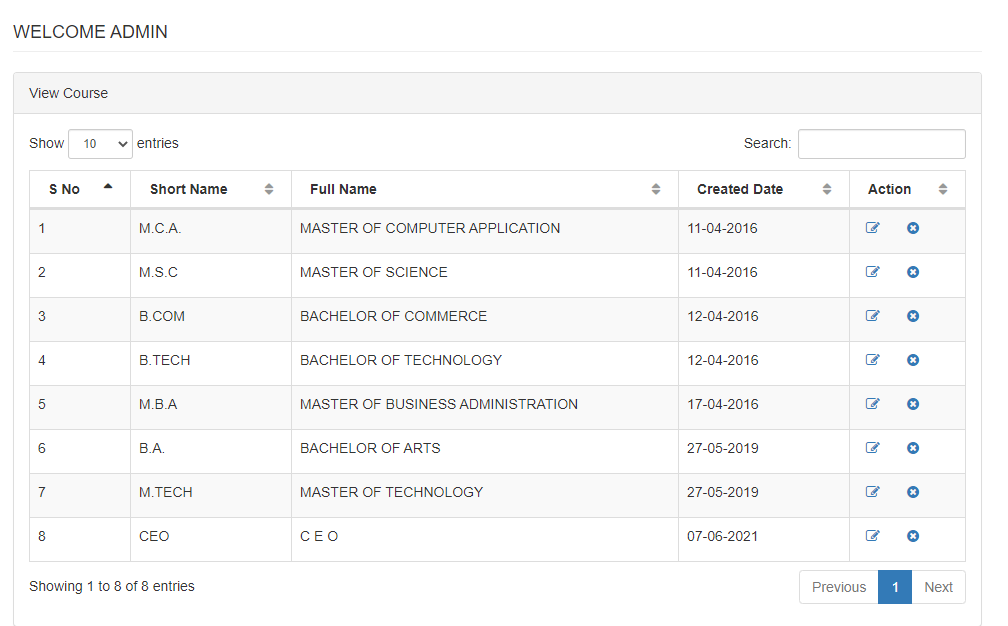


Login succes









* 1. DynamoDB

1. **Project 2 – Game đánh caro chạy trên Flask Framework (Python) và sử dụng DynamoDB**
2. Trò chơi Tic-Tac-Toe là một ứng dụng web ví dụ được xây dựng trên Amazon DynamoDB. Ứng dụng sử dụng AWS SDK cho Python (Boto) để thực hiện các lệnh gọi DynamoDB cần thiết để lưu Game Data trong bảng DynamoDB và Flask framework web Python để minh họa sự phát triển ứng dụng end-to-end trong DynamoDB, bao gồm cách lập mô hình dữ liệu.
   * 1. Cài đặt python và các thư viện cần thiết

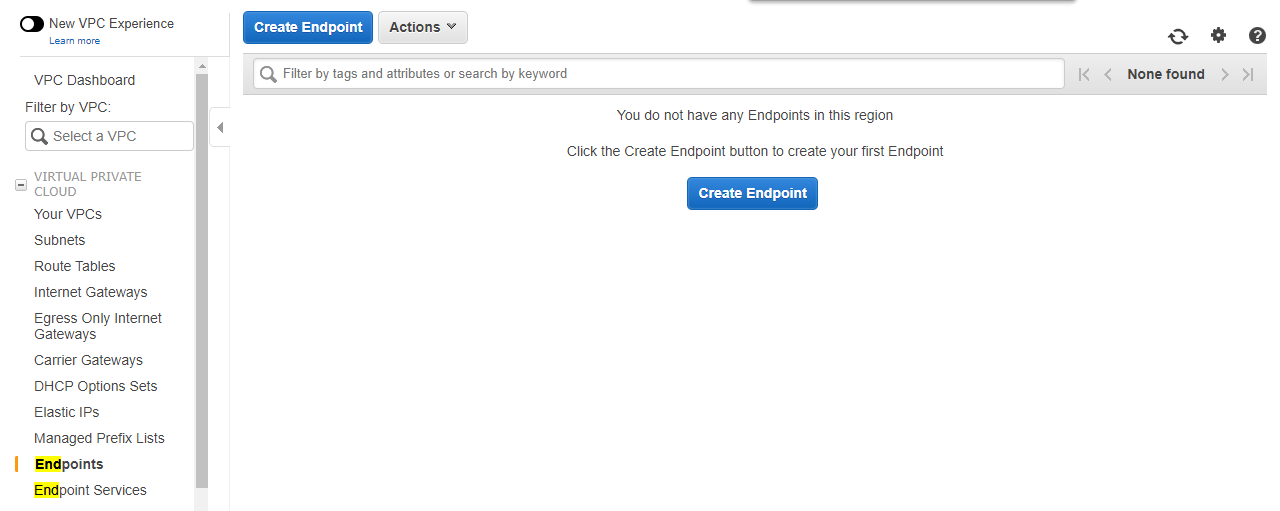
sudo amazon-linux-extras install python3

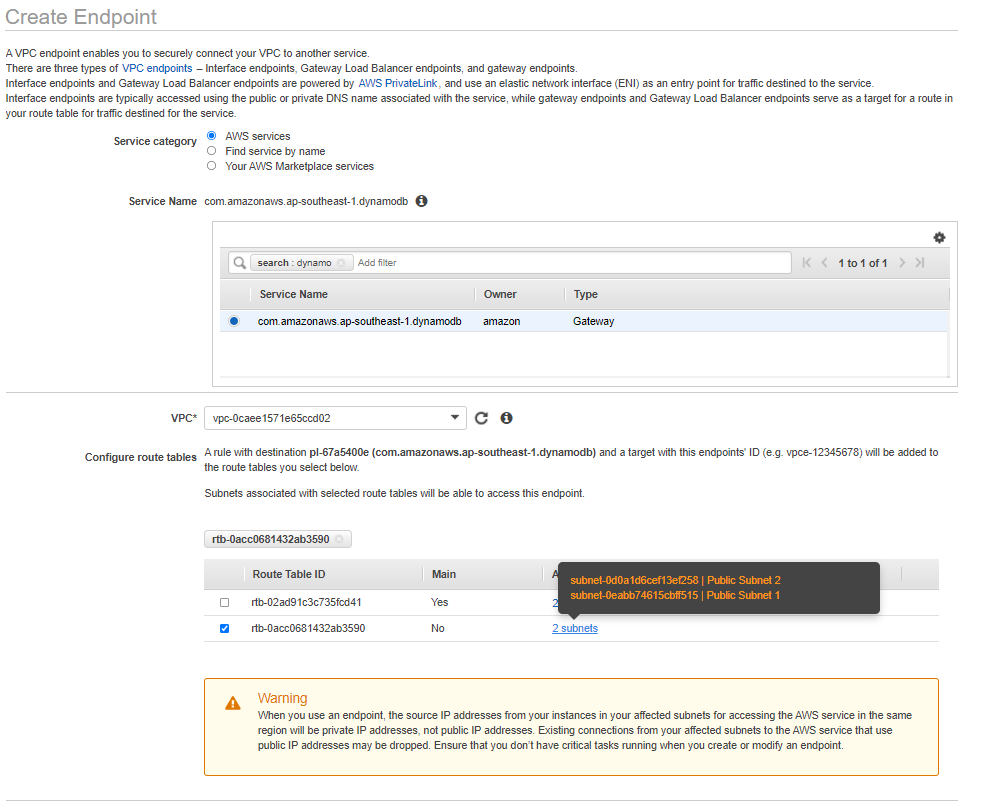


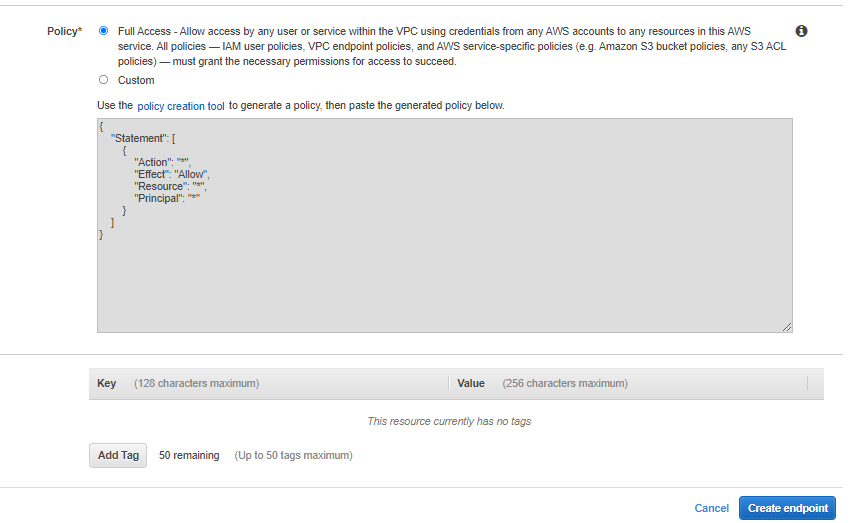
sudo python3 -m pip install Flask boto configparser

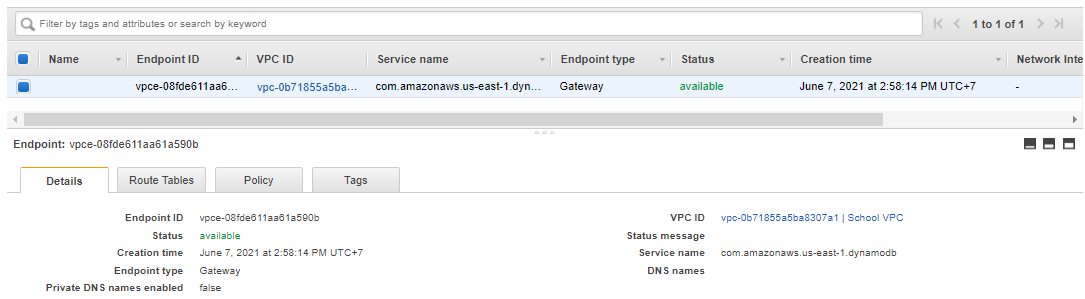
export PYTHONPATH=/usr/local/lib/python3.7/site-packages

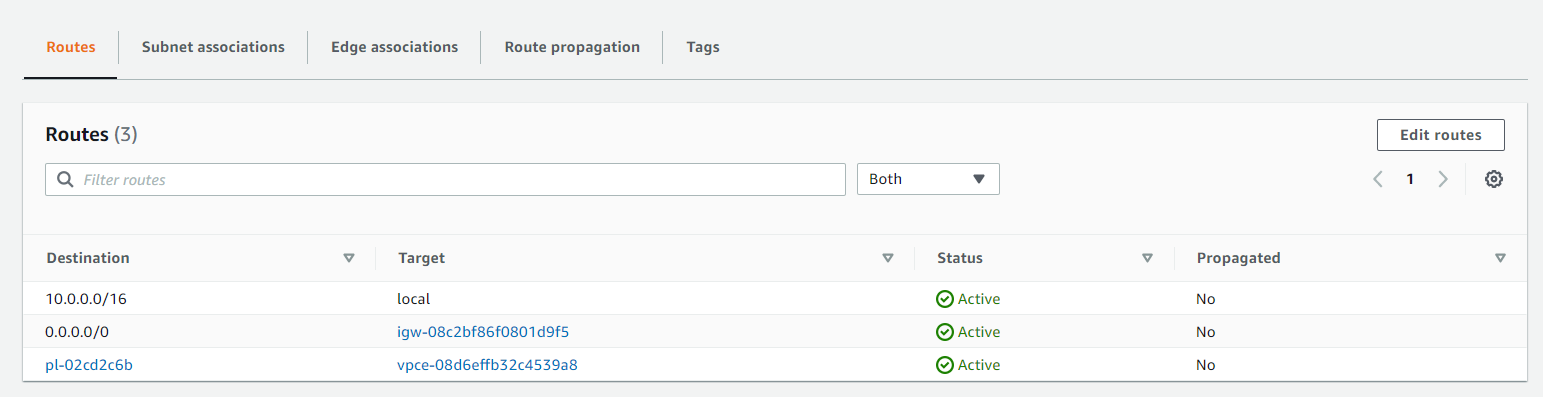
* + 1. Tạo Endpoint DynamoDB



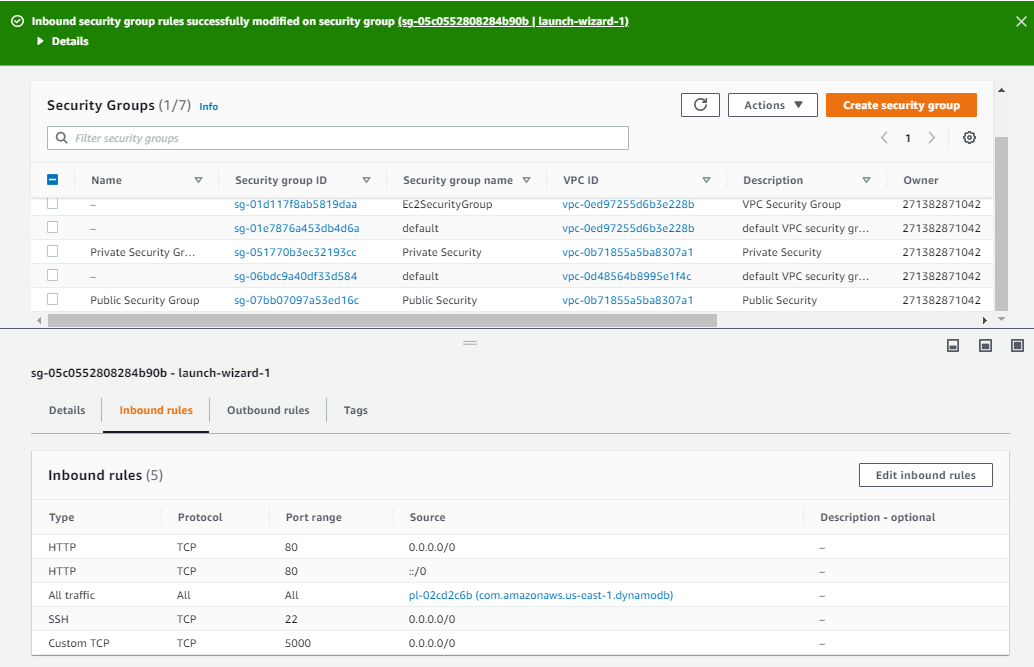








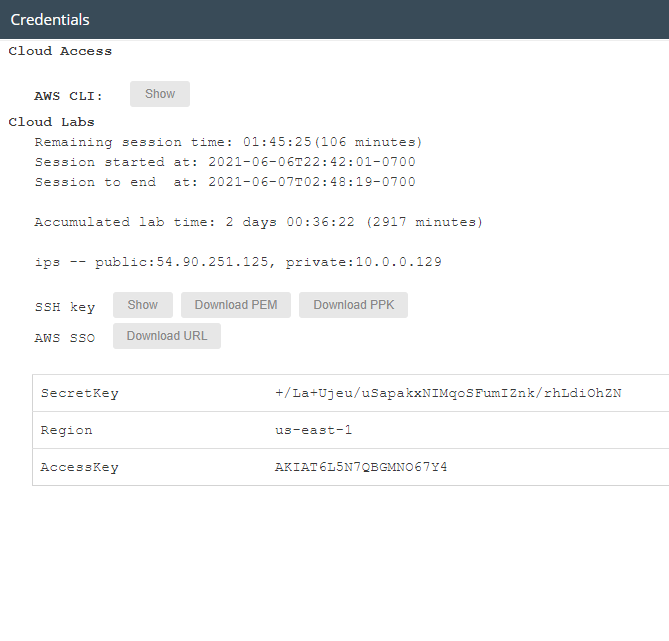
* + 1. Thiết lập Security Group cho EC2 đã tạo ở VPC



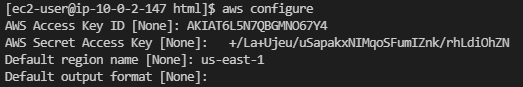
* + 1. Git Clone TicTacToe (đã tải ở VPC)



* + 1. Thiết lập key kết nối



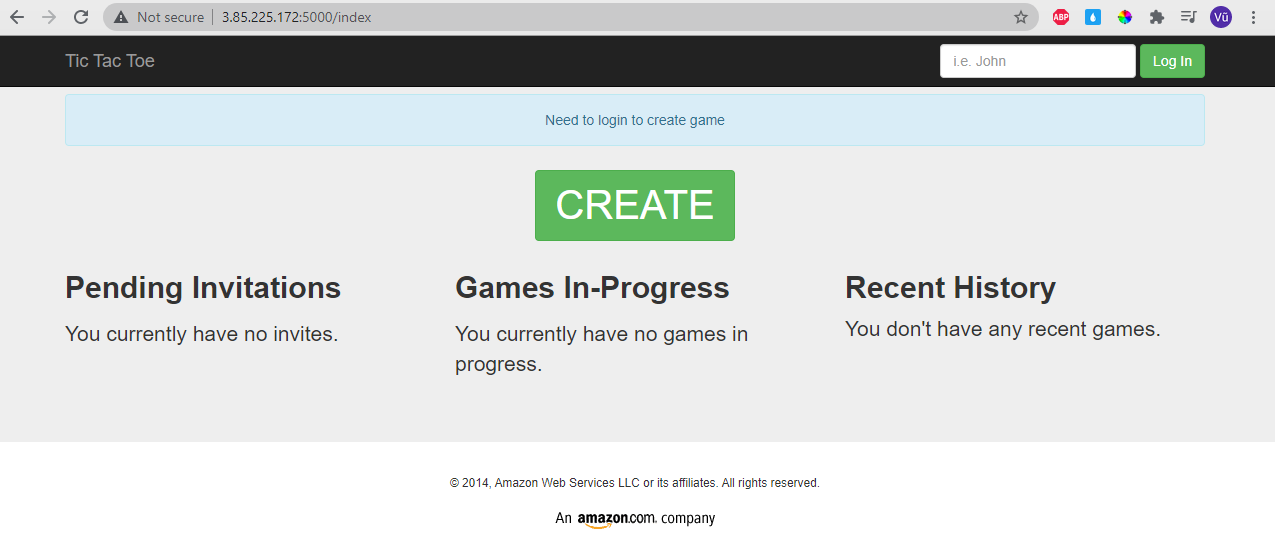
aws configure

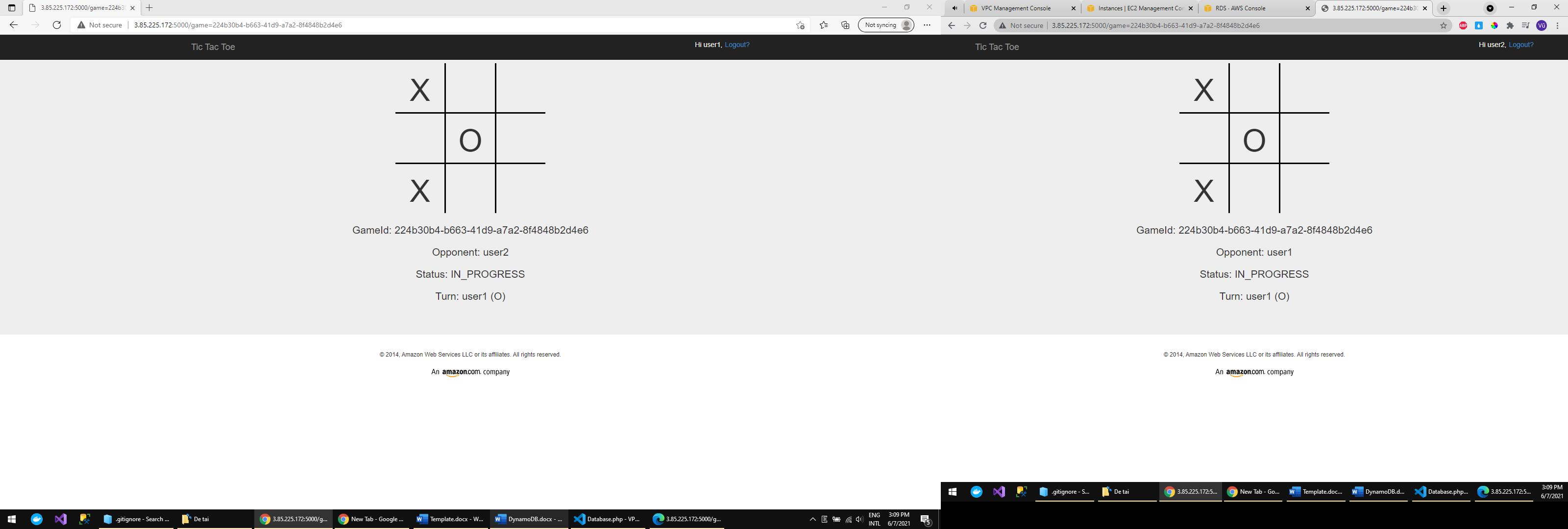


Thiết lập web.config

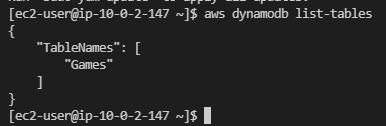
* + 1. Chạy Flask Server

python3 application.py --serverPort 5000 --port 8000 --config /var/www/htmltictactoe/web.config





Tạo Table DynamoDB thành công



1. Phần kết luận
   1. Kết quả đạt được

- Về mặt lý thuyết, hiểu được ý nghĩa của Cloud Computing, cấu trúc VPC, các thành phần network (route tables, network access controls, security groups)

- Nắm được cách thiết kế 1 kiến trúc AWS, cách triển khai, những khó khăn khi triển khai

* 1. Kết quả chưa đạt được

- Chưa tìm hiểu được cách sử dụng các dịch vụ load balancing, hệ phân tán

* 1. Hướng phát triển

- Xây dựng VPC chứa nhiều máy chủ, sử dụng load balancing

* 1. Tài liệu tham khảo

VPC - PHP – MySQL Web Services:

[Student Record System In php - PHPGurukul](https://phpgurukul.com/student-record-system-php/)

[Connect to your Linux instance using SSH - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html)

[Tutorial: Install a LAMP web server on Amazon Linux 2 - Amazon Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-lamp-amazon-linux-2.html)

[Cách tạo Cơ sở dữ liệu MySQL – Amazon Web Services](https://aws.amazon.com/vi/getting-started/hands-on/create-mysql-db/)

[Connect to Amazon RDS DB instance that's running MySQL using MySQL Workbench](https://aws.amazon.com/vi/premiumsupport/knowledge-center/connect-rds-mysql-workbench/)

[Connect to a Private Amazon RDS MySQL DB Instance that Uses a Public EC2 Instance](https://aws.amazon.com/vi/premiumsupport/knowledge-center/rds-mysql-ssh-workbench-connect-ec2/)

DynamoDB:

[AWS How to Configure a Private Network Environment for Amazon DynamoDB Using VPC Endpoints | AWS Database Blog](https://aws.amazon.com/blogs/database/how-to-configure-a-private-network-environment-for-amazon-dynamodb-using-vpc-endpoints/#:~:text=VPC%20Endpoints%20for%20DynamoDB%20enables,t%20leave%20the%20Amazon%20network.)

[DynamoDB Example Application Using the AWS SDK for Python (Boto): Tic-Tac-Toe - Amazon DynamoDB](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TicTacToe.html)