

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



HCMUTE

BÁO CÁO

XÂY DỰNG ỨNG DỤNG QUẢN LÝ THÔNG TIN SINH VIÊN SỬ DỤNG DỊCH VỤ SERVERLESS CỦA AWS

MÃ MÔN HỌC: CLCO332779_22_1_03CLC

THỰC HIỆN: Nhóm 10

GIÁO VIÊN HƯỚNG DẪN: TS. Huỳnh Xuân Phụng

Sinh viên thực hiện: Đàm Vinh Quang - 20110548

Nguyễn Quốc Huy - 20110089

Hồ Duy Hoàng - 20110487

Tp. Hồ Chí Minh, tháng 12 năm 2022

MỤC LỤC

PHẦN MỞ ĐẦU	1
GIỚI THIỆU ĐỀ TÀI.....	2
1. Các chức năng.....	2
2. Các service cần phải thực hiện trong đề tài.....	2
3. Áp dụng đối với đề tài tạo ứng dụng quản lý sinh viên sử dụng mô hình serverless.....	2
3.1. AWS LAMBDA	3
3.2. API gateway.....	3
3.3. DynamoDB	3
3.4. EC2.....	4
4. Xây dựng ứng dụng	4
TÀI LIỆU THAM KHẢO.....	5

PHẦN MỞ ĐẦU

Ngày nay, tin học đã có những tiến bước lớn và phát triển về ứng dụng của nó trong nhiều lĩnh vực đối với đời sống trên phạm vi thế giới nói chung và Việt Nam nói riêng. Những công nghệ tin học cũng được nhiều người quan tâm và nhắc đến nhiều hơn bao giờ hết vì nó là một phần không thể thiếu trong cuộc sống văn minh, góp phần đẩy mạnh cho công cuộc công nghiệp hoá hiện đại hoá các nước. Đặc biệt là với điện toán đám mây, nó giúp phân phối các tài nguyên CNTT theo nhu cầu qua internet với chính sách thanh toán theo mức dùng và được tiếp cận các dịch vụ công nghệ như năng lượng điện toán, lưu trữ và cơ sở dữ liệu khi cần thiết từ nhà cung cấp dịch vụ đám mây Amazon Web Services (AWS).

GIỚI THIỆU ĐỀ TÀI

Đề tài quản lý sinh viên không còn là đề tài quá mới mẻ với các bài toán quản lý, đó là công việc nhằm quản lý tất cả quá trình hoạt động và học tập của sinh viên trong các Đại học cũng để nâng cao về công nghệ. Đề tài Xây dựng ứng dụng quản lý thông tin sinh viên sử dụng dịch vụ serverless của AWS là sự kết hợp những dịch vụ của điện toán đám mây để tạo nên một ứng dụng quản lý thông tin sinh viên, trong đó có serverless là mô hình lưu trữ dữ liệu trên máy chủ ảo nhưng không cần quan tâm đến máy chủ.

1. Các chức năng:

- Thêm, xoá, sửa, truy xuất thông tin sinh viên

2. Các service cần phải thực hiện trong đề tài:

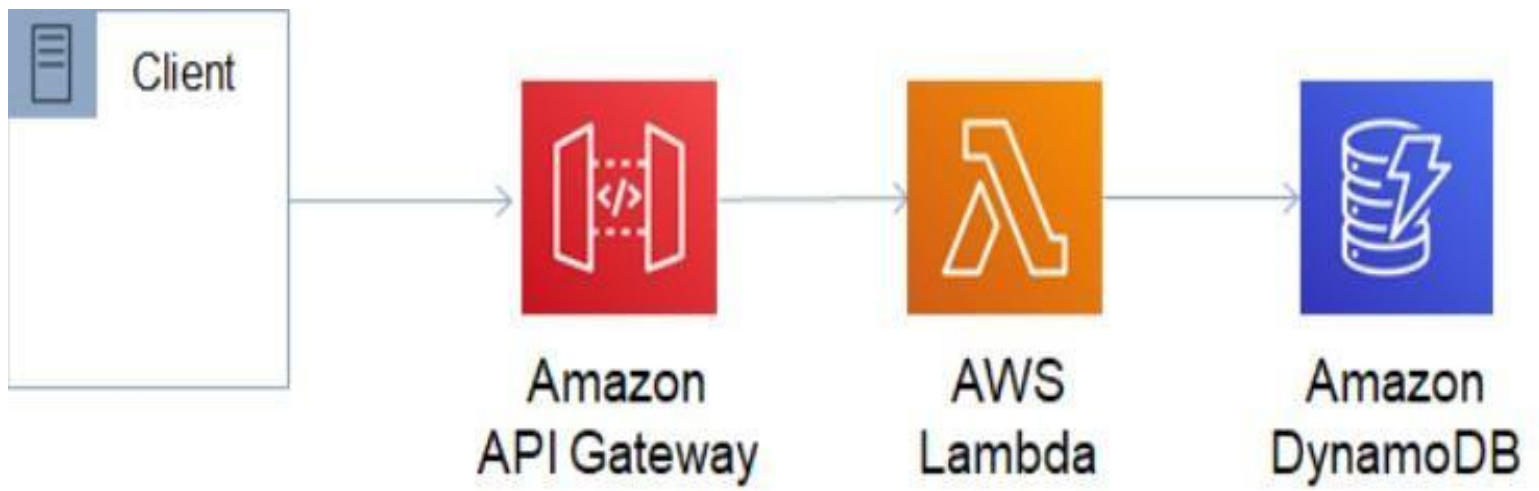
- Back end viết bằng Nodejs sử dụng dịch vụ của AWS lambda
- Front end: Reactjs, HTML, CSS, JS, BOOTSTRAP
- API gateway
- Database sử dụng dịch vụ DynamoDB

3. Áp dụng đối với đề tài tạo ứng dụng quản lý sinh viên sử dụng mô hình serverless:

Serverless là mô hình phát triển ứng dụng trên cloud mà cho phép ta xây dựng và chạy applications của chúng ta một cách dễ dàng, mà không cần quản lý server.

Cách thức hoạt động mô hình serverless để xây dựng web.

- Thao khảo mô hình Serverless Framework:



3.1. AWS LAMBDA:

Sơ lược:

AWS Lambda là một service của AWS cho phép ta chạy và quản lý function. Đây là phần chính của mô hình Serverless. Ta sẽ viết code dưới máy local và deploy nó lên AWS Lambda để chạy.

Ưu điểm:

Chạy mã mà không cần cung cấp hay quản lý cơ sở hạ tầng. Chỉ cần viết và tải lên mã dưới dạng một tệp có đuôi .zip hoặc hình ảnh bộ chứa.

Tự động đáp ứng các yêu cầu thực thi mã ở bất kỳ quy mô nào, từ hàng chục sự kiện mỗi ngày đến hàng trăm nghìn sự kiện mỗi giây.

Tiết kiệm chi phí bằng cách chỉ trả tiền cho thời gian sử dụng điện toán của bạn—mỗi mili giây—thay vì phải cung cấp trước cơ sở hạ tầng để đạt công suất tối đa.

Tối ưu hóa thời gian và hiệu suất thực thi mã với kích thước bộ nhớ hàm phù hợp. Đáp ứng nhu cầu cao trong khoảng 10 đến 99 mili giây với tính Đồng thời được cung cấp.

Các trường hợp sử dụng:

- Xử lý dữ liệu trên quy mô lớn
- Chạy các backend web và di động tương tác
- Hỗ trợ thông tin chuyên sâu mạnh mẽ về Machine learning.
- Tạo các ứng dụng theo định hướng sự kiện.

3.2. API gateway:

Amazon API Gateway là dịch vụ được quản lý hoàn toàn giúp các nhà phát triển dễ dàng tạo, phát hành, duy trì, giám sát và bảo vệ API ở mọi quy mô. API đóng vai trò là "cửa trước" cho các ứng dụng để truy cập dữ liệu, logic nghiệp vụ hoặc chức năng từ các dịch vụ backend của bạn.

Nhờ sử dụng dịch vụ này, không cần phải tracking nhiều endpoints, phần code frontend sẽ được tinh gọn hơn.

3.3. DynamoDB:

Sử dụng lưu trữ data trên DynamoDB.

Amazon DynamoDB là một cơ sở dữ liệu NoSQL khóa-giá trị được quản lý hoàn toàn, phi máy chủ và được thiết kế để chạy các ứng dụng hiệu suất cao trên mọi quy mô. DynamoDB cung cấp tính năng bảo mật tích hợp, sao lưu liên tục, sao chép đa Khu vực tự động, lưu đệm trong bộ nhớ và các công cụ tải và xuất dữ liệu.

3.4. Amazon EC2:

Amazon EC2 (Elastic Compute Cloud), một trong những dịch vụ nổi tiếng nhất của Amazon Web Services, cung cấp cho các doanh nghiệp khả năng chạy các ứng dụng trên đám mây công cộng.

EC2 cho phép người dùng xây dựng ứng dụng để tự động mở rộng quy mô theo nhu cầu thay đổi và thời kỳ cao điểm, đồng thời giúp việc triển khai máy chủ ảo và quản lý lưu trữ trở nên đơn giản, giảm thiểu nhu cầu đầu tư vào phần cứng và giúp hợp lý hóa các quy trình phát triển.

Các tính năng của EC2:

+ **Khả năng đáp ứng các yêu cầu về dung lượng thay đổi:** Việc mở rộng quy mô dễ dàng của EC2 giúp loại bỏ các trở ngại phát triển xảy ra khi các ứng dụng yêu cầu nhiều tài nguyên hơn.

+ **Linh hoạt trong cấu hình:** Người dùng có thể chọn kích thước bộ nhớ, CPU và kích thước phân vùng khởi động được tối ưu hóa cho hệ điều hành họ chọn.

+ **Kiểm soát chính xác:** Người dùng có quyền truy cập quản trị vào các phiên bản của họ, có thể dừng và bắt đầu các phiên bản trong khi vẫn giữ lại dữ liệu phân vùng khởi động.

+ **Bảo mật:** EC2 tận dụng Đám mây riêng ảo của Amazon (VPC) để

bảo mật và các doanh nghiệp có thể kết nối cơ sở hạ tầng CNTT an toàn của họ với các tài nguyên trong VPC.

4. Xây dựng ứng dụng:

- **Bước 1: Cài đặt sẵn phần mềm GitBash**
- **Bước 2: Tạo database sử dụng dịch vụ DynamoDB**
- Truy cập vào dịch vụ DynamoDB của AWS Console
- Tạo một table như sau Name: "SinhVien", Partition key: "mssv" & sau đó nhấn Create table

Create table

Table details [Info](#)

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

1 to 255 characters and case sensitive.

Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

1 to 255 characters and case sensitive.

- Sau khi Table tạo xong, nhấn vào table -> chọn "Explore table items"
- Create item -> Add new attribute, sau đó tạo các dòng ví dụ như sau, sau đó Create item (Lưu ý các Attributes phải đặt đúng tên)

Attribute name	Value	Type	
mssv - Partition key	20110548	String	
sinhviename	Đàm Vinh Quang	String	Remove
sinhviendate	06-08-2002	String	Remove
sinhviensex	Nam	String	Remove
sinhvienaddress	120 Phạm Văn Chiêu, Đắk Nông	String	Remove
sinhvienphone	0999377469	String	Remove
sinhvienmajors	Công nghệ thông tin	String	Remove
sinhviengroup	20110CL3	String	Remove
sinhvienethnic	Nùng	String	Remove

✖ You must provide the partition key attribute mssv with a non-empty value.

Cancel Create item

- Sau khi tạo thành công các items sẽ như thế này

✔ Completed Read capacity units consumed: 0.5

Items returned (1)					
		Actions ▼		Create item	
		< 1 >		⚙️ ✖	
<input type="checkbox"/>	mssv ▼	sinhvienaddress ▼	sinhviendate ▼	sinhvienethnic ▼	sinhvienphone ▼
<input type="checkbox"/>	20110548	120 Phạm Văn Chi...	06-08-2002	Nùng	20110CL3

- **Bước 3: Tạo các Function sử dụng dịch AWS Lambda**

- Truy cập vào dịch vụ AWS Lambda của AWS Console và tạo 4 function trong folder Lambda của Project
- Nhấn Create Function
- Ví dụ tạo Function: sinhvienGet để xuất thông tin sinh viên có trong database

The screenshot shows the AWS Lambda 'Create function' page. At the top, there's a breadcrumb 'Lambda > Functions > Create function'. Below it, the title 'Create function' is followed by an 'Info' link. A note states: 'AWS Serverless Application Repository applications have moved to [Create application](#).' There are three tabs: 'Author from scratch' (selected, with a blue circle icon), 'Use a blueprint', and 'Container image'. The 'Author from scratch' tab has the text 'Start with a simple Hello World example.' The 'Basic information' section contains the following fields:

- Function name:** A text input field containing 'sinhvienGet'. Below it, a note says 'Enter a name that describes the purpose of your function.' and 'Use only letters, numbers, hyphens, or underscores with no spaces.'
- Runtime:** A dropdown menu showing 'Node.js 12.x'. Above it, there's an 'Info' link and a note: 'Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.'
- Architecture:** Radio buttons for 'x86_64' (selected) and 'arm64'. Above them, there's an 'Info' link and a note: 'Choose the instruction set architecture you want for your function code.'

At the bottom of the 'Basic information' section, there is a 'Permissions' section with an 'Info' link.

Lưu ý: Nên chọn phiên bản Nodejs cũ nhất để thích hợp với Syntax code

- Change default execution role: dùng các quyền có sẵn

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LabRole

[View the LabRole role on the IAM console.](#)

► Advanced settings

Please correct the errors above.

Cancel Create function

- Create function
- Paste code `sinhvienGet.js` trong folder Lambda vào phần này & sau đó bấm Test và tạo một event có tên bất kỳ (không quan trọng) sau đó nhấn Save

Code source [Info](#) Upload from ▼

File Edit Find View Go Tools Window Test ▼ Deploy Changes not deployed

Go to Anything (Ctrl-P)

Environment

sinhvienGet /
index.js

```

1 'use strict';
2 const AWS = require('aws-sdk');
3
4 exports.handler = async(event, context) =>{
5   const documentClient = new AWS.DynamoDB.DocumentClient();
6
7   let responseBody = "";
8   let statusCode = 0;
9
10  const params = {
11    TableName: "SinhVien",
12  };
13
14  try{
15    const data = await documentClient.scan(params).promise();
16    responseBody = JSON.stringify(data.Items);
17    statusCode = 200;
18  }
19  catch(err){
20    responseBody = `Unable to select sinhvien: ${err}`;
21    statusCode = 403;
22  }
23
24  const response = {
25    statusCode: statusCode,
26    headers: {
27      "Content-Type": "application/json",
28      "access-control-allow-origin": "*"
29    },
30    body: responseBody
31  };
32  return response;
33 };
  
```

33.3 JavaScript Spaces: 4

- Các Function còn lại làm tương tự

- **Bước 4: Tạo các API Gateway & cấu hình cho đồ án**

git clone <https://github.com/NguyenQuocHuy2k2/ProjectDientoandammay>

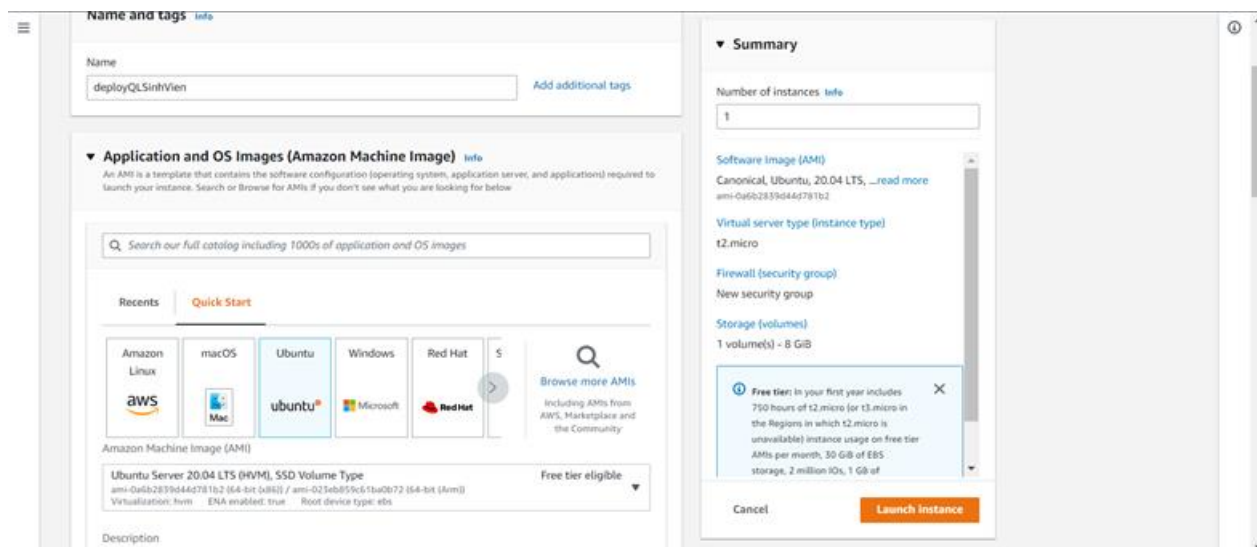
Hướng dẫn tạo các API và cấu hình: <https://youtu.be/fVNlcqoPM7Y>

- **Bước 5: Mở GitBash tại Desktop và tiến hành chạy**

- Sau khi git, tại Desktop sẽ xuất hiện Folder ProjectDientoandammay
- cd ProjectDientoandammay
- npm i
- npm start

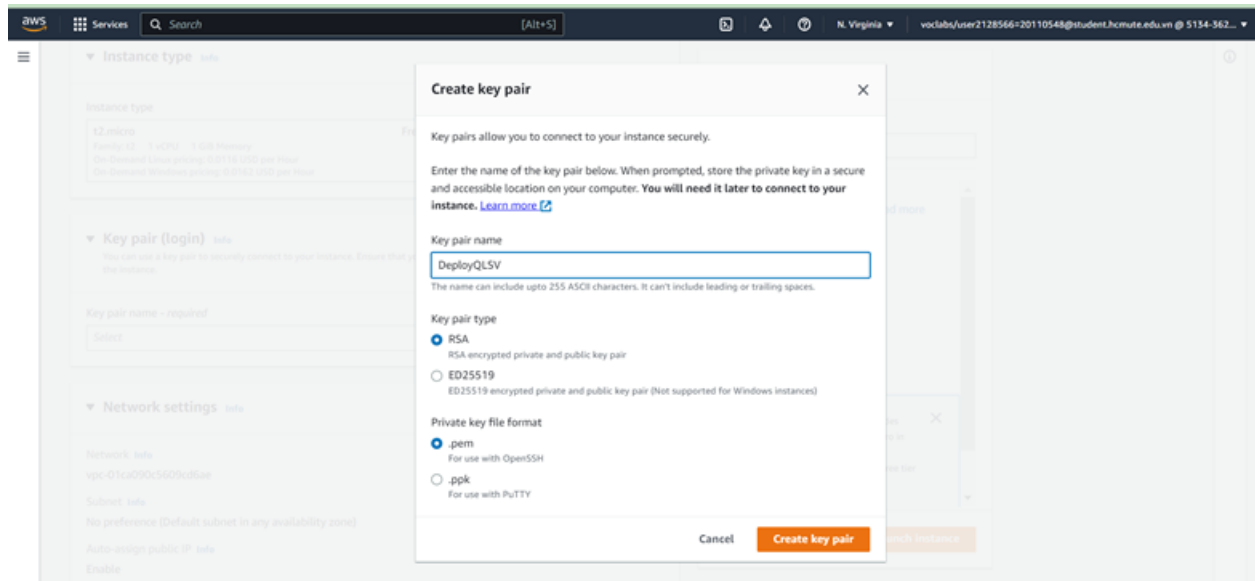
- **Bước 6: Tiến hành Deploy ứng dụng lên AWS sử dụng dịch vụ Amazon EC2**

- Chạy EC2:
- Bước 1. Mở EC2 và tạo launch Instances
- --> Chọn ubuntu server 20.04
- <https://user-images.githubusercontent.com/113155221/207244638-f56ad7d8-3907-451e-9ac1-091c4c22f3b5.png>

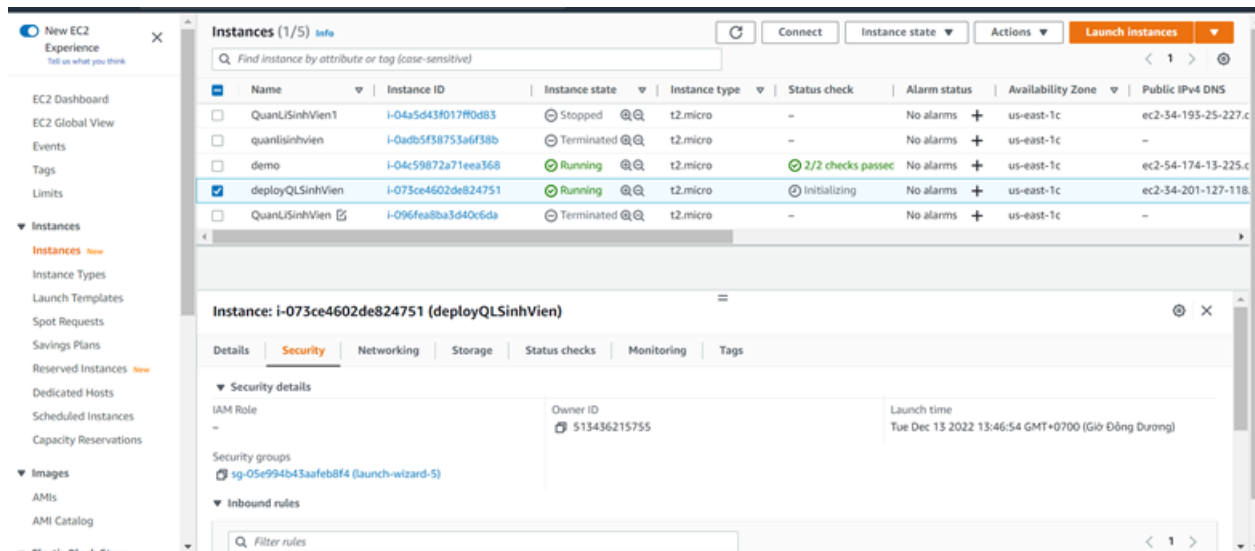


- --> Tiến hành tạo 1 key mới, tạo thư mục deployaws và lưu file vào một thư mục đó

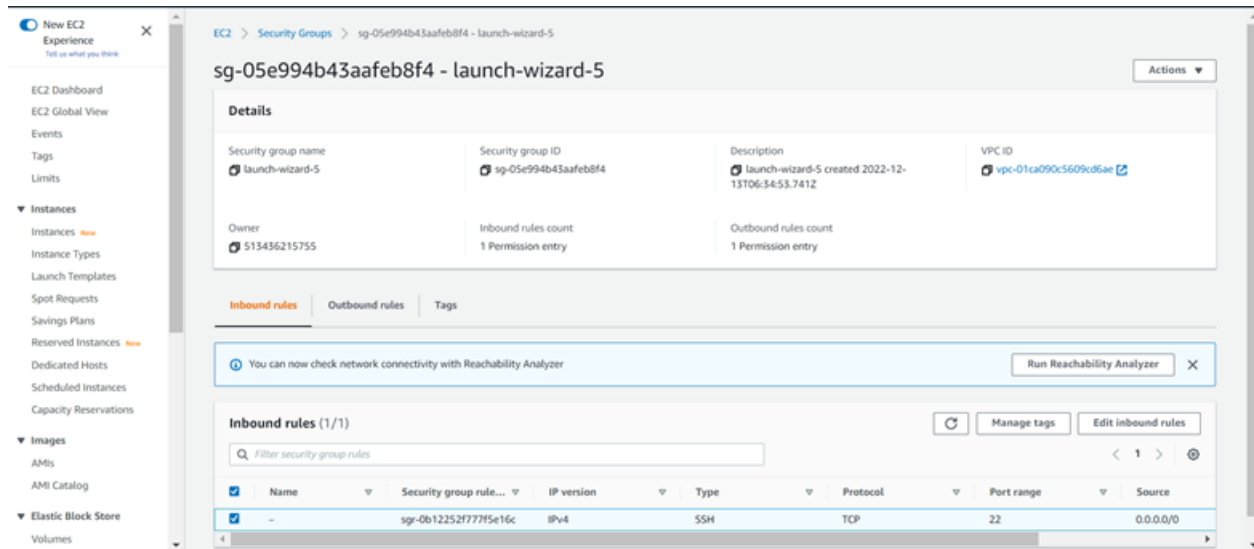
- <https://user-images.githubusercontent.com/113155221/207245306-4ef9ce5c-b5c3-4a14-8e41-e8b28b26a78d.png>



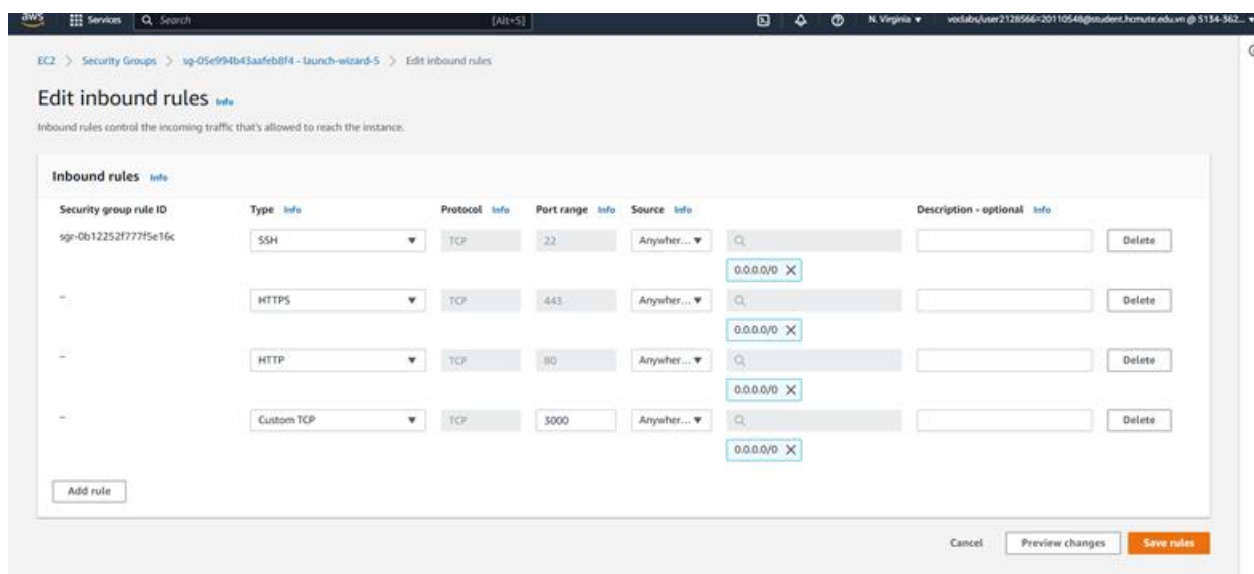
- --> Launch instance
- Bước 2:
- --> Chọn name: deployQLSinhVien -> chọn security-> chọn sg-05e994b43aafef8f4
- <https://user-images.githubusercontent.com/113155221/207250727-ccfb7795-77c0-4c67-a67f-abad719f7334.png>



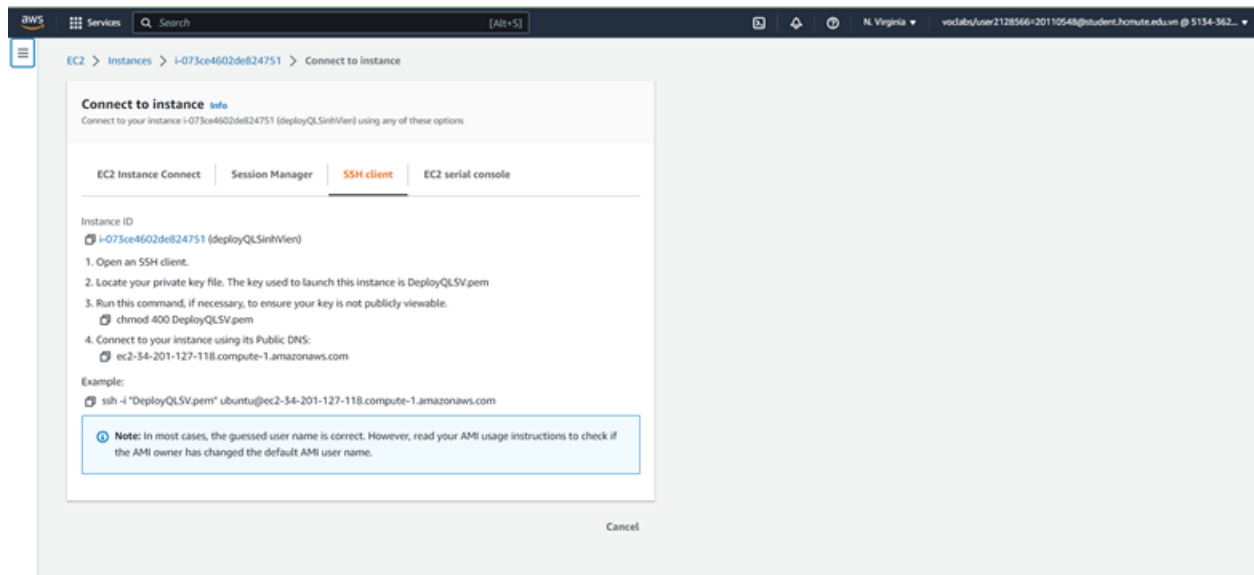
- --> Chọn Edit inbound rules để thêm rule
- <https://user-images.githubusercontent.com/113155221/207250756-d151ac48-b7bf-4c0a-ba4b-08ff950188df.png>



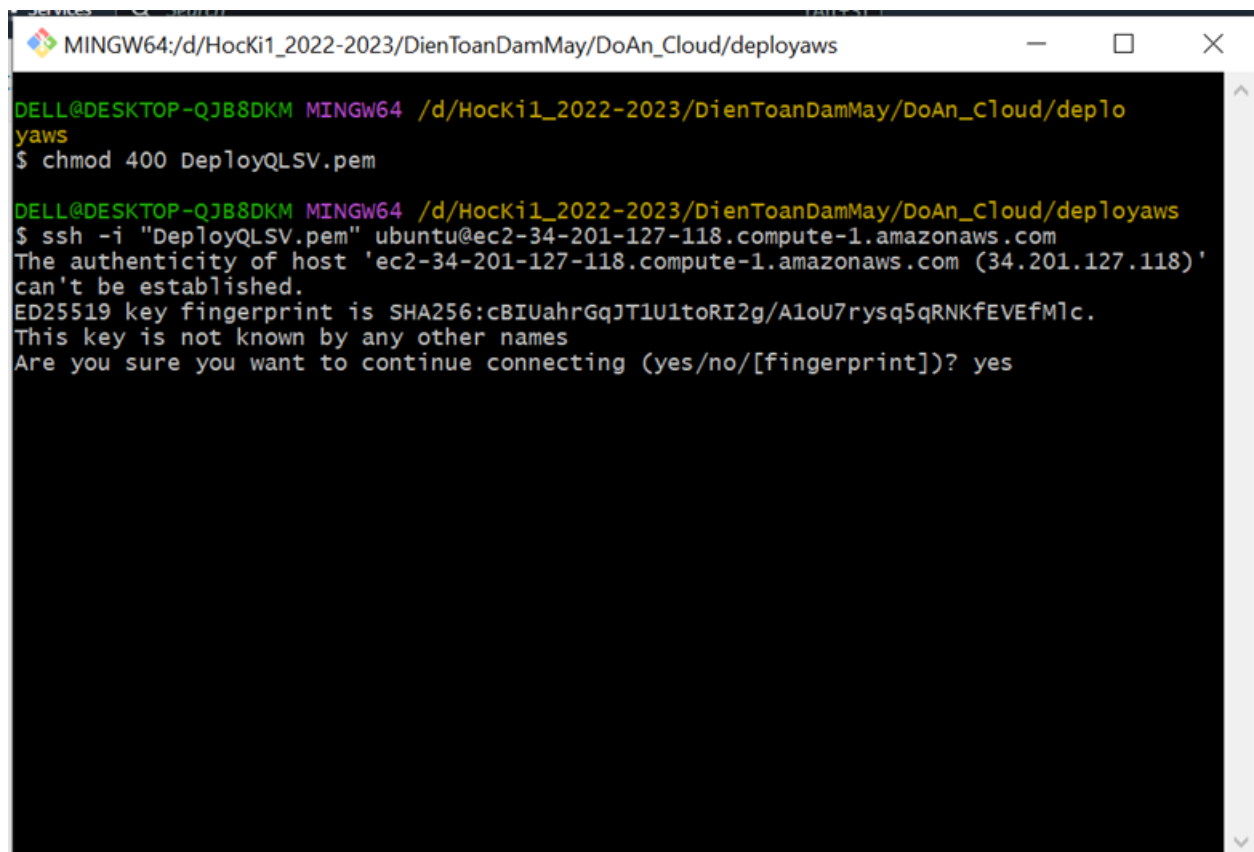
- --> Tạo thêm 3 rule
- https 443 Anywhere-IPv4
- http 80 Anywhere-IPv4
- custom TCP 3000 Anywhere-IPv4
- <https://user-images.githubusercontent.com/113155221/207250807-e7f22643-792a-452e-b56a-563a58ada1eb.png>



- --> save rules để lưu
- Bước 3:
- --> Instances / deployQLSinhVien / Connect
- <https://user-images.githubusercontent.com/113155221/207250847-ca735e4f-3f55-4ebd-8dec-3dbb8fc64efd.png>



- --> Tiếp theo ta vào thư mục deployaws có chứa file key vừa này tạo bất gitbash gõ các lệnh
- - chmod 400 DeployQLSV.pem
- - ssh -i "DeployQLSV.pem" [ubuntu@ec2-34-201-127-118.compute-1.amazonaws.com](https://user-images.githubusercontent.com/113155221/207250932-2cb901ec-49c2-4205-bcf6-0d1d1f96b49a.png) và gõ yes
- <https://user-images.githubusercontent.com/113155221/207250932-2cb901ec-49c2-4205-bcf6-0d1d1f96b49a.png>



- --> Sau đó ta gõ các lệnh theo thứ tự
- - sudo apt install curl
- - curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash
- - source ~/.bashrc
- - nvm install 14.21.1
- - nvm use 14.21.1
- - git clone <https://github.com/NguyenQuocHuy2k2/ProjectDientoandammay.git>
- - cd ProjectDientoandammay
- - npm install react-scripts --save
- - npm install
- - npm start
- - sau khi chạy xong ta copy Public IPv4 address: 34.201.127.118 với port 3000
- => đường dẫn là http://34.201.127.118:3000/

TÀI LIỆU THAM KHẢO

1. <https://viblo.asia/p/tim-hieu-ve-amazon-ec2-maGK7jRe5j2>
2. <https://viblo.asia/p/tim-hieu-ve-serverless-framework-3P0IP898lox>
3. <https://wiki.tino.org/serverless-la-gi/>
4. <https://topdev.vn/blog/lam-quen-voi-kien-truc-serverless/>
5. <https://vtitech.vn/tim-hieu-ve-serverless-framework/>