**CLOUD DEVELOPER**

Udacity Nanodegree

**Author:**

* Vương Hữu Phúc - PhucVH17
* Đoàn Phước Thi – ThiDP1

Giới thiệu tổng quan và một số thông tin cần chú ý khi học và tham gia làm project.

# **Overview**

Khóa học bao gồm 4 nội dung chính với 5 project sẽ được thực hiện:

* **Cloud Fundamentals**
  + Project 1: Deploy Static Website on AWS
* **Full Stack Apps on AWS**
  + Project 2: Your Own Instagram on AWS
* **Monolith to Microservices at Scale**
  + Project 3: Refactor Monolith to Microservices and Deploy
* **Develop & Deploy Serverless App**
  + Project 4: Serverless Application
* **Capstone**
  + Project 5: Choose either microservices or serverless design

Về tổng quan, khóa học sẽ cho bạn kiến thức về: Điện toán đám mây, Microservices và Serverless.

Nắm vững kiến thức cơ bản về AWS, xây dựng và triển khai các ứng dụng trên cloud và nắm bắt các nguyên tắc của kiến trúc đám mây.

Đi sâu vào microservices, hiểu về kiến trúc các mẫu, sao chép dịch vụ và mở rộng độc lập.

Tìm hiểu về serverless, tạo REST API và triển khai các phương pháp hay nhất về bảo mật.

Khóa học ở mức độ dễ, giúp bạn hiểu biết hơn về AWS, cung cấp kinh nghiệm thực tế, trang bị cho bạn thiết kế, xây dựng và triển khai các ứng dụng phức tạp.

# **Prerequisite**

Khóa học sẽ cần một số kiến thức về các mảng sau:

* AWS CLI, AWS account, Node.JS, NPM, ElasticBeanstalk CLI, Docker, and Iconic CLI
* IDE: Visual studio code

Tất cả các project sẽ làm lab thực tế trên AWS, account do Udacity cung cấp.

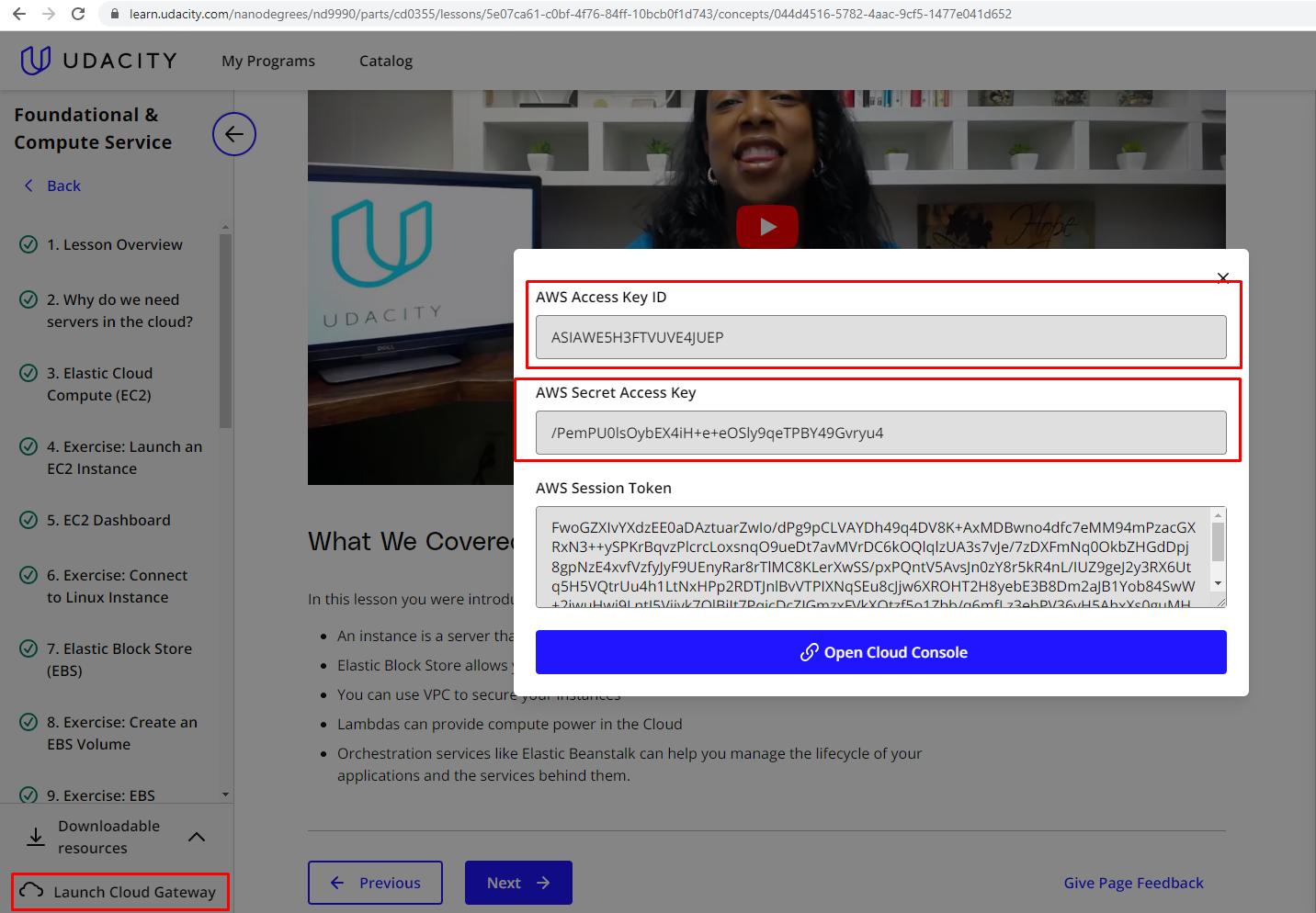
# **Attention**

Để có thể sử dụng <https://us-east-1.console.aws.amazon.com/> bạn có thể sử dụng tài khoản AWS do udacity cung cấp hoặc đăng ký tài khoản cá nhân.

Lưu ý khi sử dụng tài khoản AWS do udacity cung cấp thì chỉ có ngân sách cho mỗi project là $25 và sẽ bị giới hạn dịch vụ có thể sử dụng nhưng vẫn đảm bảo hoàn thành khóa học. Nên hãy cố gắng tận dụng và hoàn thành nhanh nhất. **Sau khi sử dụng xong, luôn luôn xóa hết tài nguyên**.

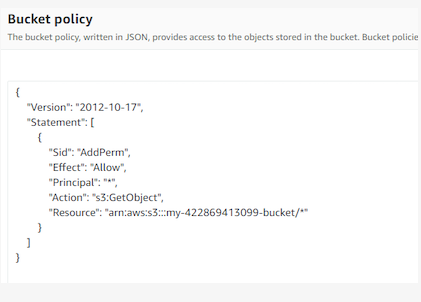
Nếu bạn muốn sử dụng tài khoản cá nhân thì có thể đăng ký trên AWS.

Khi chuẩn bị submit project để review, hãy dành thời gian đọc phần mô tả project và kiểm tra lại phần rubic xem mình đã làm đủ các yêu cầu trong đó chưa. Bạn cần phải đáp ứng hết thì mới có thể **PASS** được.

**Đối với 4 project bạn sẽ được cung cấp 4 account khác nhau. Mỗi account sẽ có key khác nhau.**  


# **Project 01: Deploy Static Website on AWS**

Project đầu tiên này là phần dễ nhất trong 5 project mà bạn sẽ làm, bạn cứ làm theo những bước ở phần mô tả project 1 và **cần đặc biệt chú ý đến phần policy của S3 bucket** (Phần này được xem là phần quan trọng nhất để pass)



# **Project 02: Udagram: Your Own Instagram on AWS**

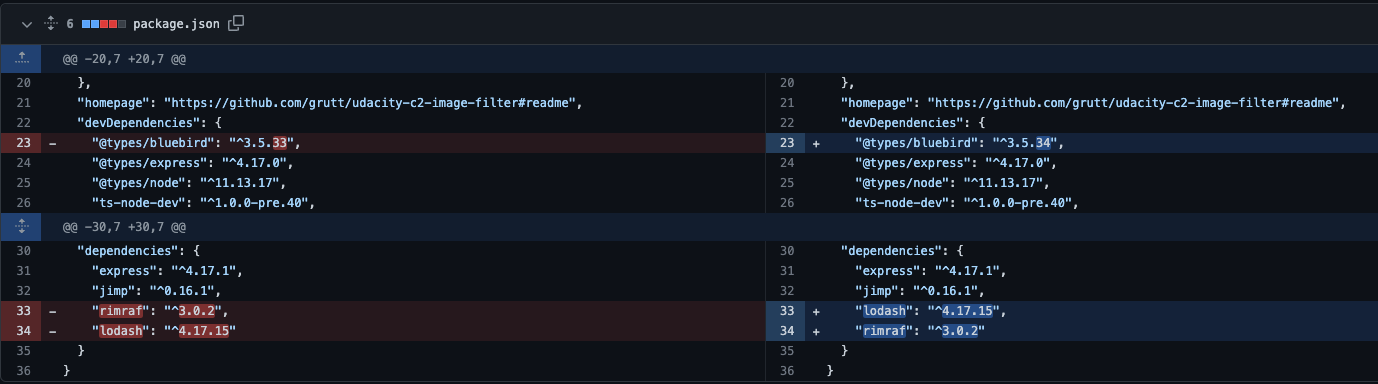
Trong project này bạn sẽ phát triển một ứng dụng dựa trên đám mây để tải lên, liệt kê và lọc hình ảnh. Họ sẽ sử dụng Node.js/Express, một khung JavaScript phổ biến để phát triển ứng dụng nối mạng để phát triển ứng dụng này.

Bạn phải triển khai API REST để ra lệnh bằng HTTP, lưu trữ dữ liệu trong Dữ liệu quan hệ của Amazon Web Services Service (RDS) và S3, mở rộng cơ sở mã bằng các tính năng đăng nhập xác thực an toàn và triển khai cho Amazon Web.

* **Những lưu ý khi thực hiện project 02**

Khi build project cần chú ý đến version của các package. Ví dụ: node, bluebird…

Đối với node js thì version chủ yếu là 16.x



Khi làm việc với git thì bạn phải tạo ít nhất 2 branch để làm việc (master, develop,...).  
Khi thực hiện những @TODO thì chú ý đến phần type code:

- Bất kỳ biến nào được khai báo thì cần có tên biến, kiểu dữ liệu và hàm rõ ràng.

*Ví dụ*: **let image\_local: string[] = [];**

Đối với mục 4. deletes any files on the server on finish of the response thì nên sử dụng promise và then.

Link tham khảo: [Promise.prototype.then() - JavaScript | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/then)

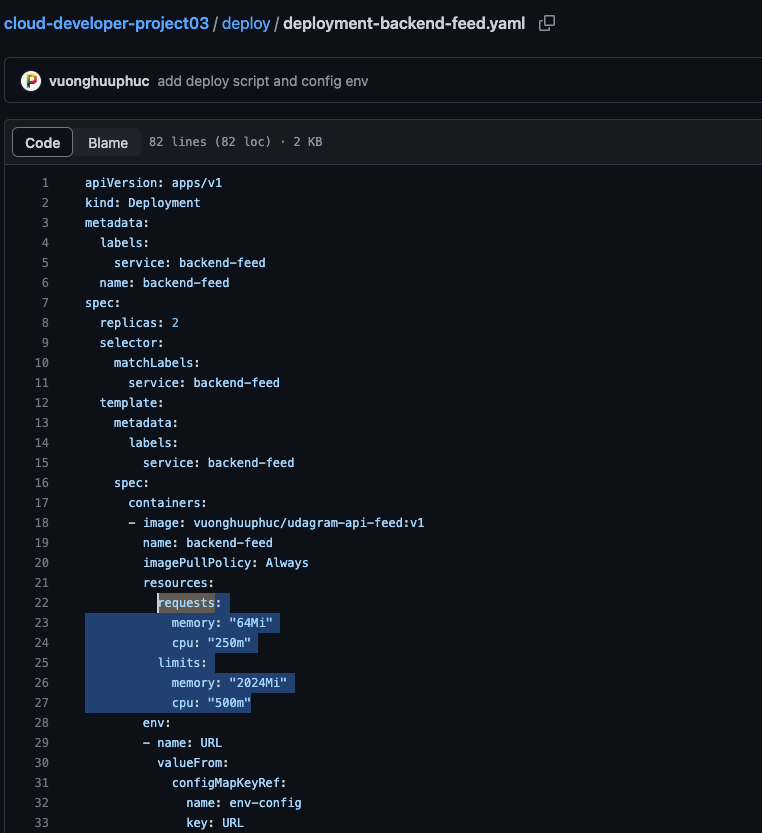
# **Project 03: Refactor Udagram App into Microservices and Deploy** Trong project này, bạn sẽ lấy một ứng dụng hiện có tên là Udagram và cấu trúc lại nó thành một kiến trúc microservice với dịch vụ tinh gọn. Xây dựng quy trình CI/CD tự động xây dựng và triển khai hình ảnh Docker cho cụm Kubernetes. Các Cụm Kubernetes sẽ được định cấu hình để giúp giải quyết các thách thức chung liên quan đến quy mô và bảo mật.

* **Những lưu ý khi thực hiện project 03**

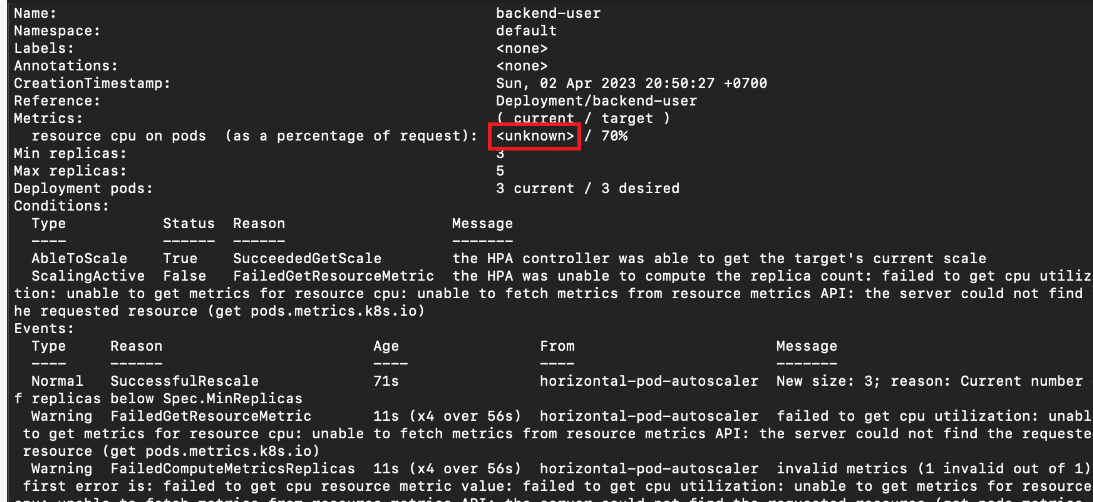
Đây là project khó nhất trong Nanodegree này.

Bạn nên dành nhiều thời gian để tham khảo, tìm hiểu nhiều hơn để hoàn thành project.

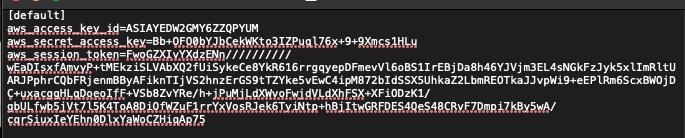
* Bạn sẽ phải đăng ký 1 account Travis CI để Travis CI hoạt động thì chúng ta cần add thẻ tín dụng (-1$ và sẽ hoàn trả sau 24h) sau khi add xong thì cần báo với bộ hỗ trợ Travis Ci thông qua email:[support@travis-ci.com](mailto:support@travis-ci.com)
* Tại file yaml bạn phải nâng memory và CPU lên thì mới hoạt động được, và số lượng replica sẽ là 1; riêng con **backend-feed-deployment.yaml** sẽ để là số lượng replica là 2, để tránh trường hợp bị crash sạch

  
- Khi tạo node EKS thì bạn nên chọn là **m5.large** nhưng lưu ý loại này sẽ tốn nhiều tiền hơn khi chạy. Sau khi học hoặc làm project thì chúng ta xóa đi để tránh những phát sinh không cần thiết

Sau khi chạy lệnh **kubectl get hpa** thì giá trị metrics phải có số % tránh trường hợp <unknown> như bên dưới:



* Chú ý phần mã hóa base64 cho những KEY của AWS. Chúng ta cần encrypt đủ 3 phần: aws\_access\_key\_id; aws\_secret\_access\_key; aws\_session\_token nằm trong file credentials rồi mã hóa base64



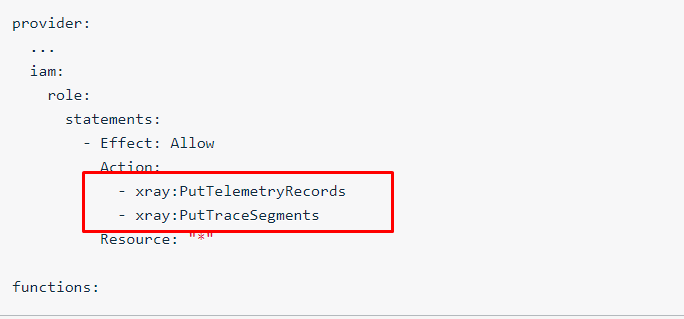
Base64 Decode and Encode – Online

**Project 04: Serverless Application**

# Trong dự án này, học viên sẽ phát triển một ứng dụng serverless để tải lên, list và lọc hình ảnh và bạn sẽ bắt đầu với việc xây dựng API REST serverless bằng API Gateway và AWS Lambda. Sau đó, bạn sẽ sử dụng các dịch vụ AWS khác (DynamoDB, S3, OpenSearch) để lưu trữ dữ liệu và bảo mật ứng dụng bằng xác thực.

# Cuối cùng, người học sẽ triển khai ứng dụng của họ lên AWS Serverless Framework.

* **Những lưu ý khi thực hiện project 04**
* Phần này tập trung nhiều nhất vào coding type\_script
* Ở 2 function create/update thì phải validate (required...)
* Tổ chức cấu trúc thư mục phải đúng với mô hình 3 lớp như trong bài học đã có mô tả
* Về phần config policy ở file serverless.yml
* Khi set permission thì không nên khai báo nhiều function trong 1 statements



* Để tránh lỗi CORS cần chú ý phần set CorsRules – AllowedOrigins cho S3 bucket

# **Project 05: Choose either microservices or serverless design**

# Phần này bạn sẽ chọn 1 trong 2 project 03 hoặc project 04 sau đó phát triển thêm 1 hoặc vài chức năng là sẽ pass.

# **Ví dụ**: Ở Project 04 có thể thêm tính năng xóa ảnh

# **Support Channel**

* **Sử dụng kênh support của Udacity**:
* Knowledge Portal Udacity/ Mentor Help cho các vấn đề về nội dung
* GPT
* Account Help cho các vấn đề kĩ thuật: <https://udacityenterprise.zendesk.com/hc/en-us/requests/new>
* Cloud Developer Teams Group
* **Contact Point**: DanPNL (HN+OB)/ ThaoNT39 (ĐN+ Hue +QN)/ VyTT4 (HCM+CT)