CHƯƠNG TRÌNH THỰC TẬP AWS FCJ INTERNSHIP 2025

BÁO CÁO ĐỀ XUẤT DỰ ÁN (PROJECT PROPOSAL)

ĐỀ TÀI: PHÁT HIỆN BẤT THƯỜNG CHI PHÍ TRÊN NỀN TẢNG AWS SỬ DỤNG MACHINE LEARNING

(COST ANOMALY DETECTION WITH MACHINE LEARNING)

Sinh viên thực hiện: PHẠM CÔNG NHẬT TRIỀU

MSSV: 2180609467

Email: trieu1292003@gmail.com

Thành phố Hồ Chí Minh, tháng 07 năm 2025

CHƯƠNG TRÌNH THỰC TẬP AWS FCJ INTERNSHIP 2025

BÁO CÁO ĐỀ XUẤT DỰ ÁN (PROJECT PROPOSAL)

ĐỀ TÀI: PHÁT HIỆN BẤT THƯỜNG CHI PHÍ TRÊN NỀN TẢNG AWS SỬ DỤNG MACHINE LEARNING

(COST ANOMALY DETECTION WITH MACHINE LEARNING)

Sinh viên thực hiện: PHẠM CÔNG NHẬT TRIỀU

MSSV: 2180609467

Email: trieu1292003@gmail.com

Người hướng dẫn:

Thời gian thực hiện: 01/07/2025 – 15/07/2025

Thành phố Hồ Chí Minh, tháng 07 năm 2025

MỤC LỤC

PHẦN 1: TÓM TẮT ĐIỀU HÀNH (EXECUTIVE SUMMARY)

Trong bối cảnh các doanh nghiệp và tổ chức ngày càng phụ thuộc vào hạ tầng điện toán đám mây, việc giám sát chi phí vận hành trên nền tảng AWS trở thành một nhu cầu cấp thiết nhằm đảm bảo tính hiệu quả tài chính và khả năng kiểm soát ngân sách. Tuy nhiên, phương thức giám sát truyền thống chủ yếu dựa trên ngưỡng tĩnh hoặc phát hiện thủ công không còn đáp ứng kịp tốc độ biến động của hệ thống hiện đại. Điều này dẫn đến nguy cơ phát sinh các chi phí bất thường mà không được phát hiện kịp thời, gây tổn thất về tài chính và ảnh hưởng đến hoạt động vận hành.

Dự án ***"Phát hiện bất thường chi phí sử dụng Machine Learning trên nền tảng AWS"*** được xây dựng nhằm giải quyết vấn đề nêu trên thông qua việc ứng dụng các kỹ thuật học máy hiện đại vào dữ liệu chi phí AWS. Mục tiêu của dự án là phát triển một hệ thống có khả năng phân tích dữ liệu chi tiêu, nhận diện sớm các hành vi bất thường, gửi cảnh báo tự động và hỗ trợ công tác điều tra nguyên nhân một cách hiệu quả. Giải pháp được thiết kế tối giản về mặt hạ tầng, có thể triển khai trong thời gian ngắn và phù hợp với sinh viên hoặc tổ chức có ngân sách giới hạn.

1) Tổng quan giải pháp đề xuất

Giải pháp bao gồm một quy trình xử lý dữ liệu chi phí AWS được tự động hóa từ khâu thu thập, làm sạch, đến huấn luyện và triển khai mô hình học máy (Isolation Forest hoặc AutoEncoder). Mô hình sẽ phân tích các mẫu chi tiêu theo thời gian và gắn cờ các điểm bất thường. Khi phát hiện sự kiện vượt ngưỡng, hệ thống sẽ sử dụng Amazon SNS để gửi cảnh báo đến người quản lý thông qua email. Các sự kiện và xu hướng chi phí cũng sẽ được trực quan hóa thông qua dashboard, hỗ trợ quá trình điều tra và ra quyết định.

2) Các tính năng chính

- Thu thập và xử lý dữ liệu từ AWS Cost and Usage Report (CUR)
- Phân tích chuỗi thời gian và nhận dạng bất thường chi phí
- Gửi cảnh báo tự động qua Amazon SNS hoặc email

- Dashboard phân tích chi phí và hiển thị điểm bất thường
- Hệ thống có khả năng huấn luyện lại mô hình định kỳ để nâng cao độ chính xác

3) Giá trị mang lại

- Tăng cường khả năng kiểm soát và minh bạch tài chính trên nền tảng AWS
- Giảm tổn thất chi phí do phát hiện sớm các sự cố và lạm dụng tài nguyên
- Hỗ trợ bộ phận tài chính và vận hành trong việc đánh giá, phân tích và lập kế hoạch ngân sách
- Góp phần xây dựng quy trình giám sát chi phí chủ động, phù hợp với định hướng FinOps hiện đại

4) Thời gian và chi phí triển khai

Toàn bộ dự án được thiết kế với thời gian thực hiện **dưới 14 ngày**, sử dụng các dịch vụ AWS thuộc Free Tier hoặc có chi phí rất thấp như Amazon S3, AWS Lambda, Amazon SNS và Amazon SageMaker. **Tổng chi phí dự kiến dưới 10 USD**, đảm bảo phù hợp với nguồn lực của sinh viên thực tập sinh trong khuôn khổ chương trình FCJ.

5) Chỉ số đánh giá thành công

- Độ chính xác phát hiện bất thường (Precision/Recall) đạt tối thiểu 85%
- Thời gian phản hồi của cảnh báo dưới 5 phút tính từ thời điểm chi phí tăng đột biến
- Tỷ lệ tiết kiệm chi phí trung bình kỳ vọng đạt 10–20% trong vòng 3 tháng
- Hệ thống có khả năng mở rộng áp dụng với nhiều tài khoản AWS hoặc các dự án lớn hơn

6) Kết luận

Đề tài này thể hiện một giải pháp có tính thực tiễn cao, dễ triển khai, chi phí thấp, nhưng vẫn đáp ứng được đầy đủ các tiêu chí kỹ thuật, tài chính và nghiệp vụ. Dự án không chỉ phục vụ mục tiêu học thuật trong khuôn khổ thực tập mà còn có thể được áp dụng trong môi trường doanh nghiệp thực tế, góp phần nâng cao hiệu quả vận hành và quản lý chi phí trên nền tảng AWS.

PHẦN 2: TUYÊN BỐ VẤN ĐỀ (PROBLEM STATEMENT)

2.1. Bối cảnh và động lực nghiên cứu

Trong bối cảnh điện toán đám mây trở thành nền tảng hạ tầng chủ đạo cho các hệ thống thông tin hiện đại, quản trị chi phí hiệu quả trong môi trường như Amazon Web Services (AWS) đang là mối quan tâm hàng đầu của cả doanh nghiệp lớn lẫn các tổ chức giáo dục, nhóm nghiên cứu hoặc người dùng cá nhân như sinh viên. Khi dịch vụ được mở rộng theo mô hình pay-as-you-go, rủi ro phát sinh chi phí đột biến là hoàn toàn có thể xảy ra – dù là từ lỗi hệ thống, tấn công mạng, cấu hình sai, hay đơn giản là người dùng quên tắt tài nguyên.

Mặc dù AWS cung cấp các công cụ như AWS Budgets, Cost Explorer hay cảnh báo ngưỡng chi phí tĩnh (static thresholds), nhưng những công cụ này không đủ thông minh để phát hiện các mẫu bất thường phức tạp trong chi tiêu – đặc biệt là trong các môi trường sử dụng biến động cao hoặc nhiều tài khoản AWS đồng thời. Do đó, nhu cầu về một hệ thống có khả năng học tập hành vi chi tiêu bình thường, từ đó tự động phát hiện và cảnh báo sớm những sai lệch là một nhu cầu cấp thiết.

2.2. Định nghĩa vấn đề

Bài toán đặt ra là: Làm thế nào để xây dựng một hệ thống phát hiện bất thường chi phí AWS có thể hoạt động gần thời gian thực, tự động nhận diện các hành vi chi tiêu không bình thường và đưa ra cảnh báo phù hợp, sử dụng Machine Learning nhưng vẫn đảm bảo chi phí thấp, dễ triển khai và phù hợp với năng lực của sinh viên trong thời gian ngắn.

Cụ thể, bài toán tập trung vào:

- Thu thập và phân tích dữ liệu Cost and Usage Report (CUR) từ AWS.
- Xây dựng mô hình phát hiện bất thường sử dụng thuật toán học không giám sát như Isolation Forest hoặc AutoEncoder.
- Thiết lập pipeline cảnh báo tức thời khi phát hiện bất thường.
- Trực quan hóa dữ liệu nhằm hỗ trợ điều tra nguyên nhân gốc.

2.3. Phân tích khó khăn hiện hữu

Vấn đề kỹ thuật	**Biểu hiện cụ thể**	**Tác động**
Phát hiện thủ công	Dựa vào kiểm tra chi tiết từng hóa đơn hoặc bảng tính	Mất thời gian, dễ bỏ sót
Cảnh báo tĩnh	Chỉ phát hiện khi vượt ngưỡng đặt trước	Không phát hiện được hành vi bất thường tinh vi
Thiếu khả năng điều tra	Không có bảng điều khiển hoặc phân tích nguyên nhân	Khó khăn trong xử lý sự cố kịp thời

Thống kê từ báo cáo của CloudZero (2024) cho thấy, **trung bình 30–40% chi phí bị lãng phí trong môi trường đám mây do thiếu kiểm soát chủ động**, và việc phát hiện bất thường muộn thường dẫn đến thiệt hại tài chính không nhỏ, đặc biệt với các nhóm không có chuyên gia tài chính chuyên trách – như sinh viên hoặc startup giai đoạn đầu.

2.4. Các bên liên quan và ảnh hưởng

Bên liên quan	**Vai trò**	**Lợi ích từ giải pháp**
Sinh viên/nhóm phát triển	Người trực tiếp sử dụng và triển khai	Kiểm soát tài nguyên và chi phí hiệu quả
Giảng viên/hướng dẫn dự án	Đánh giá chất lượng học thuật và kỹ thuật	Dễ theo dõi tiến độ và kiểm tra tính khả thi

Bên liên quan	Vai trò	Lợi ích từ giải pháp
Quản lý tài khoản AWS (nếu có)	Quản trị tài nguyên trên nhiều lớp	Có công cụ cảnh báo tức thời

Ví dụ minh họa thực tế: Một nhóm sinh viên triển khai đồ án sử dụng EC2 để huấn luyện mô hình AI nhưng quên tắt máy chủ sau giờ làm. Sau 3 ngày, hóa đơn tăng gấp 10 lần dự kiến mà không được phát hiện kịp thời. Việc có hệ thống cảnh báo bất thường sẽ giúp nhóm phát hiện chỉ sau 15–30 phút và xử lý ngay, tránh tổn thất.

2.5. Hệ quả nếu không có giải pháp

- **Tài chính:** Mất kiểm soát chi tiêu, gây áp lực lên ngân sách dự án, thậm chí ngưng triển khai giữa chừng.
- **Kỹ thuật:** Tài nguyên bị sử dụng lãng phí kéo dài làm giảm hiệu suất toàn hệ thống.
- **Học thuật:** Thiếu công cụ giám sát hiện đại ảnh hưởng đến chất lượng đào tạo kỹ năng FinOps và CloudOps.
- **Tinh thần:** Mất động lực triển khai do lo ngại rủi ro tài chính không kiểm soát được.

2.6. Cơ hội thị trường và khả năng mở rộng

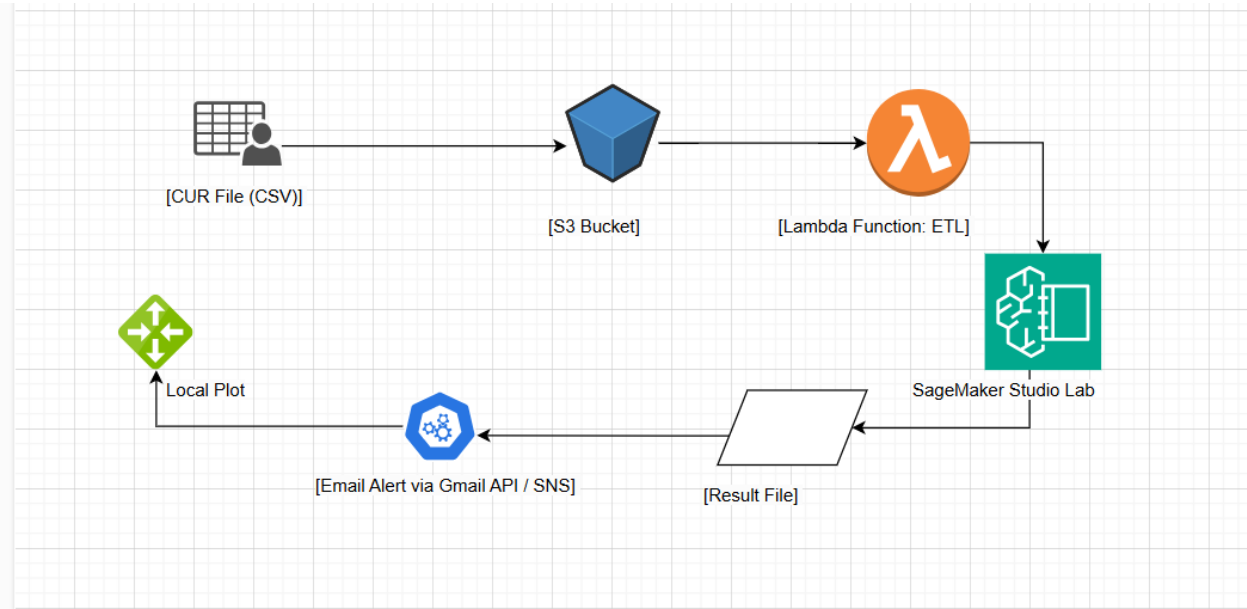
- **Tiềm năng áp dụng:** Không chỉ dừng ở sinh viên, giải pháp này có thể mở rộng sang các nhóm khởi nghiệp, tổ chức giáo dục hoặc doanh nghiệp nhỏ – nơi chưa có đủ nguồn lực tài chính để triển khai các công cụ FinOps thương mại.
- **Khả năng mở rộng kỹ thuật:** Sau khi thử nghiệm thành công ở quy mô nhỏ, hệ thống có thể nâng cấp để xử lý dữ liệu thời gian thực với AWS Kinesis, mở rộng sang đa tài khoản bằng Cost Explorer API và tích hợp với hệ thống CI/CD.

PHẦN 3: KIẾN TRÚC GIẢI PHÁP (SOLUTION ARCHITECTURE)

3.1. Tổng quan kiến trúc hệ thống

Hệ thống phát hiện bất thường chi phí AWS được thiết kế với kiến trúc tối giản, hướng đến mục tiêu dễ triển khai, tận dụng dịch vụ miễn phí hoặc trong Free Tier, đồng thời đảm bảo các yếu tố sau:

- Thu thập dữ liệu chi phí từ AWS Cost and Usage Report (CUR).
- Tiền xử lý dữ liệu để định dạng phù hợp cho mô hình Machine Learning.
- Phát hiện bất thường bằng mô hình học không giám sát như Isolation Forest.
- Gửi cảnh báo ngay khi phát hiện điểm bất thường.
- Trực quan hóa chi phí và log cảnh báo.



Tổng thể hệ thống là **serverless**, **chi phí thấp**, sử dụng các pipeline thủ công/b bán tự động dễ quản lý bởi sinh viên trong môi trường học thuật.

3.2. Lựa chọn dịch vụ AWS và lý do sử dụng

Dịch vụ AWS	Vai trò trong hệ thống	Lý do lựa chọn
Amazon S3	Lưu trữ dữ liệu đầu vào (CUR) và output kết quả	Free Tier, dễ tích hợp, bền vững
AWS Lambda	Thực hiện tiền xử lý dữ liệu (ETL), kiểm tra định dạng	Miễn phí trong Free Tier, không cần quản lý server
Amazon SNS *(tuỳ chọn)*	Gửi cảnh báo đến email/SMS người quản lý	Dễ tích hợp, realtime, không cần cron job
SageMaker Studio Lab *(hoặc local)*	Huấn luyện mô hình ML trên Jupyter Notebook	Miễn phí, phù hợp sinh viên, không tốn chi phí tài nguyên compute

3.3. Thiết kế các thành phần chính

3.3.1. Thu thập dữ liệu (Data Ingestion)

- Người dùng tải CUR thủ công từ AWS Billing Console hoặc thiết lập cho tự động gửi vào S3 định kỳ.
- File dạng .csv chứa thông tin dịch vụ, loại sử dụng, tài khoản, khu vực, thời gian và chi phí.

3.3.2. Tiền xử lý dữ liệu (Data Preprocessing)

- Sử dụng AWS Lambda để lọc ra các trường cần thiết (e.g., UsageStartDate, ServiceName, Cost).
- Loại bỏ outliers không liên quan, chuẩn hoá theo khoảng thời gian (ngày/tuần).

3.3.3. Mô hình học máy (ML Model)

- Mô hình sử dụng thuật toán học không giám sát như **Isolation Forest** hoặc **AutoEncoder**.
- Huấn luyện offline trên SageMaker Studio Lab hoặc Google Colab.
- Đầu ra là nhãn bất thường cho từng dòng chi phí.

3.3.4. Cảnh báo (Alerting)

- Khi phát hiện dòng bất thường, hệ thống gửi cảnh báo tới email bằng:
  - Amazon SNS (nếu có thời gian cấu hình IAM)
  - Hoặc đơn giản hơn: dùng Python script gửi email qua Gmail SMTP API.

3.3.5. Trực quan hoá (Visualization)

- Dữ liệu kết quả được vẽ biểu đồ (line chart, anomaly point) bằng:
  - matplotlib trong Jupyter Notebook
  - Hoặc cập nhật Google Sheet rồi dùng Google Chart API

3.4. Kiến trúc bảo mật và tuân thủ

Phân quyền truy cập tối thiểu (Least Privilege):

- Lambda chỉ có quyền truy cập S3 Bucket cụ thể.
- Không cấp quyền đọc toàn bộ S3 hoặc tài khoản AWS.

Không lưu trữ thông tin nhạy cảm:

- CUR không chứa dữ liệu người dùng, không cần mã hoá cấp độ cao.

Đăng nhập và kiểm soát truy cập:

- Nếu sử dụng Studio Lab, cần bật xác thực 2 bước.

Tuân thủ:

- Giải pháp mang tính học thuật, nhưng vẫn theo chuẩn cơ bản AWS Well-Architected: chi phí thấp – an toàn – linh hoạt.

3.5. Khả năng mở rộng và hiệu năng hệ thống

- **Khả năng mở rộng:**
  - Có thể triển khai thêm các thành phần:
    - Trigger Lambda theo lịch (EventBridge)
    - Dùng SageMaker real-time endpoint để phân tích liên tục
    - Kết nối đa tài khoản AWS nếu chuyển sang môi trường doanh nghiệp
- **Hiệu năng:**
  - File CUR mỗi ngày thường nhỏ (<5 MB) → xử lý nhanh trong vài giây
  - Lambda có thể xử lý hàng ngàn dòng CUR trong vòng vài trăm mili-giây
  - Hệ thống cảnh báo có thể gửi thông báo sau khi mô hình chạy chưa đến 10 phút

Ví dụ mở rộng thực tiễn:

- Hệ thống hiện tại dùng bản local Jupyter Notebook để huấn luyện.
- Sau này có thể mở rộng sang CI/CD pipeline: mỗi lần có CUR mới → tự động chạy tiền xử lý → chạy model → cảnh báo + dashboard.

PHẦN 4: CHIẾN LƯỢC TRIỂN KHAI KỸ THUẬT (TECHNICAL IMPLEMENTATION)

4.1. Các giai đoạn thực hiện và sản phẩm bàn giao

Để đảm bảo khả năng hoàn thành trong vòng 2 tuần và phù hợp với mức độ phức tạp trung bình, quá trình triển khai được chia thành 6 giai đoạn chính, tương ứng với các sản phẩm bàn giao cụ thể:

Giai đoạn	**Mô tả ngắn gọn**	**Sản phẩm bàn giao**
1. Phân tích & xác định yêu cầu	Phân tích CUR và xác định logic phát hiện bất thường	Tài liệu phân tích dữ liệu đầu vào
2. Tiền xử lý dữ liệu	Viết hàm làm sạch, chuẩn hóa và lọc dữ liệu từ CUR	Script xử lý dữ liệu (Lambda hoặc local)
3. Huấn luyện mô hình ML	Dùng Isolation Forest hoặc AutoEncoder để huấn luyện mô hình	Notebook mô hình với kết quả đánh giá
4. Hệ thống cảnh báo	Gửi email khi phát hiện điểm bất thường	Script gửi mail cảnh báo (SNS hoặc SMTP)
5. Trực quan hóa dữ liệu	Biểu đồ chi phí và điểm bất thường	Biểu đồ matplotlib / Google Sheets
6. Kiểm thử và hoàn thiện	Thử pipeline toàn bộ, hiệu chỉnh nếu cần	Báo cáo tổng hợp, hướng dẫn sử dụng

4.2. Yêu cầu kỹ thuật (tính toán, lưu trữ, mạng)

Hạng mục	**Yêu cầu cụ thể**
Tính toán (Compute)	Không yêu cầu cao, sử dụng local hoặc SageMaker Studio Lab (CPU 2-core là đủ)
Lưu trữ (Storage)	Dữ liệu CUR dạng CSV, dung lượng nhỏ: ~5MB/ngày
Mạng (Network)	Cần quyền truy cập internet để:
	- Tải CUR từ AWS Console
	- Gửi email qua SMTP API hoặc SNS
	- Lưu trữ/đọc dữ liệu từ S3

Do hệ thống không xử lý dữ liệu real-time lớn, nên toàn bộ pipeline có thể chạy trên laptop cá nhân hoặc hạ tầng miễn phí.

4.3. Phương pháp phát triển và kỹ thuật áp dụng

- Phương pháp phát triển: Iterative / Agile thu nhỏ, theo chu kỳ kiểm tra – hiệu chỉnh ngắn (2–3 ngày/lần).
- Ngôn ngữ lập trình chính: Python 3.10
- Môi trường phát triển:
  - Jupyter Notebook (Google Colab hoặc SageMaker Studio Lab)
  - AWS Lambda (Python runtime)
- Kỹ thuật chính áp dụng:

- Xử lý dữ liệu: pandas, numpy, boto3
- Học máy: sklearn với IsolationForest hoặc AutoEncoder (nếu cần)
- Trực quan hóa: matplotlib, seaborn
- Gửi email: smtplib, hoặc tích hợp SNS SDK

Mọi phần đều được kiểm soát qua các script nhỏ, dễ kiểm thử độc lập.

4.4. Kế hoạch kiểm thử (đơn vị, tích hợp, hiệu năng)

Loại kiểm thử	Mục tiêu	Cách thực hiện
Kiểm thử đơn vị	Đảm bảo mỗi module chạy đúng logic	Test script xử lý CUR, script gửi mail, hàm đánh giá model
Kiểm thử tích hợp	Kiểm tra pipeline hoàn chỉnh	Dữ liệu từ CUR → qua Lambda → mô hình → cảnh báo email
Kiểm thử hiệu năng	Đo độ trễ xử lý pipeline	Tính thời gian từ nạp file → đến gửi cảnh báo; mục tiêu < 10 phút
Kiểm thử mô hình	Đánh giá độ chính xác mô hình phát hiện bất thường	Sử dụng dữ liệu mẫu có gán nhãn, so sánh Precision, Recall

4.5. Kế hoạch triển khai và quy trình khôi phục

4.5.1. Kế hoạch triển khai

- Toàn bộ pipeline có thể triển khai bằng tay hoặc script CLI đơn giản.

- Không cần CI/CD, nhưng mọi phần có thể tích hợp vào shell script hoặc Python CLI để tự động hóa sau này.
- Các bước triển khai gồm:

1. Đưa CUR vào thư mục /data
2. Chạy script preprocess.py
3. Chạy train_model.ipynb nếu chưa có mô hình
4. Chạy detect_anomaly.py để phát hiện và gửi email
5. Lưu kết quả log vào file hoặc bảng Google Sheet

4.5.2. Quy trình khôi phục

Do hệ thống nhỏ, chiến lược khôi phục đơn giản:

- Nếu lỗi ở mô hình → retrain lại bằng train_model.ipynb
- Nếu lỗi ở Lambda → cập nhật lại hàm hoặc chuyển sang local script
- Nếu lỗi cảnh báo → kiểm tra kết nối mạng / xác thực SMTP
- Toàn bộ mã nguồn nên lưu trữ trên GitHub cá nhân để backup và khôi phục dễ dàng

PHẦN 5: KẾ HOẠCH THỜI GIAN VÀ CÁC MỐC CHÍNH (TIMELINE AND MILESTONES)

5.1. Lộ trình thực hiện theo giai đoạn

Dự án được thiết kế để hoàn thành trong vòng **14 ngày làm việc**, tương ứng với thời gian thực tập thực tế của sinh viên. Lộ trình thực hiện được chia thành 6 giai đoạn chính như sau:

Giai đoạn	**Mô tả công việc chính**	**Thời lượng (ngày)**
Giai đoạn 1: Khảo sát & Phân tích	Phân tích CUR, xác định mục tiêu, ràng buộc kỹ thuật	1
Giai đoạn 2: Thu thập & Tiền xử lý	Xử lý dữ liệu CSV từ CUR, kiểm tra định dạng và chuẩn hóa dữ liệu	2
Giai đoạn 3: Huấn luyện mô hình	Áp dụng Isolation Forest hoặc AutoEncoder, đánh giá trên dữ liệu mẫu	4
Giai đoạn 4: Phát hiện & cảnh báo	Xây dựng hệ thống cảnh báo bất thường qua email/SNS	2
Giai đoạn 5: Trực quan hóa & UX	Trực quan hóa dữ liệu chi phí và điểm bất thường qua biểu đồ	2
Giai đoạn 6: Kiểm thử & Báo cáo	Viết báo cáo, hoàn thiện demo, kiểm thử hệ thống	2
Dự phòng	Dự trù lỗi kỹ thuật, trì hoãn	**1**

Giai đoạn	Mô tả công việc chính	Thời lượng (ngày)
Tổng cộng		**14 ngày**

5.2. Các mốc chính và tiêu chí đánh giá thành công

Mốc (Milestone)	Thời điểm (Ngày)	Tiêu chí đánh giá thành công
M1 – Phân tích dữ liệu hoàn tất	Ngày 1	Hiểu rõ định dạng CUR, xác định biến đầu vào mô hình
M2 – Dữ liệu tiền xử lý sẵn sàng	Ngày 3	Có tập dữ liệu sạch, cấu trúc phù hợp cho ML
M3 – Mô hình phát hiện hoạt động	Ngày 7	Huấn luyện xong mô hình với độ chính xác đạt ≥ 80% (trên dữ liệu mẫu)
M4 – Hệ thống cảnh báo vận hành	Ngày 9	Gửi được email cảnh báo khi phát hiện điểm bất thường
M5 – Dashboard trực quan hoàn chỉnh	Ngày 11	Biểu đồ chi phí và điểm bất thường được hiển thị chính xác
M6 – Báo cáo và demo hoàn tất	Ngày 14	Báo cáo 20 trang hoàn chỉnh, demo chạy được từ đầu đến cuối không lỗi

5.3. Phân tích phụ thuộc và đường găng

Công việc	**Phụ thuộc vào**	**Ghi chú**
Tiền xử lý dữ liệu	Phân tích dữ liệu	CUR phải hiểu đúng mới xử lý được
Huấn luyện mô hình	Dữ liệu đã xử lý	Input cần sạch, không bị lỗi định dạng
Cảnh báo bất thường	Mô hình đã triển khai	Phụ thuộc đầu ra của mô hình
Trực quan hóa	Cảnh báo hoặc dữ liệu đầu ra từ mô hình	Lấy dữ liệu để hiển thị
Viết báo cáo & demo	Tất cả các phần trên	Là bước cuối, đòi hỏi hệ thống hoàn chỉnh

5.4. Kế hoạch phân bổ nguồn lực và thời gian dự phòng

PHẦN 6: ƯỚC TÍNH CHI PHÍ VÀ PHÂN TÍCH TÀI CHÍNH (BUDGET ESTIMATION AND ROI ANALYSIS)

6.1. Chi phí hạ tầng AWS

Chi phí hạ tầng trong dự án được tối ưu theo hướng sử dụng **Free Tier**, tận dụng dịch vụ miễn phí và công cụ mã nguồn mở. Ước tính chi phí hàng tháng (đơn vị: USD):

Dịch vụ AWS	**Mục đích sử dụng**	**Dung lượng/Thời gian**	**Ước tính chi phí**
Amazon S3	Lưu trữ dữ liệu CUR và kết quả xử lý	1–2 GB	0.05–1.00
AWS Lambda	Tiền xử lý và tính toán nhẹ	<1 triệu request/tháng	0.00 (Free Tier)
Amazon SNS (hoặc Gmail SMTP)	Gửi cảnh báo email	<100 email/tháng	0.00
AWS CloudWatch (nếu dùng)	Ghi log cảnh báo, theo dõi lỗi	Dưới ngưỡng miễn phí	0.00

Tổng chi phí hạ tầng ước tính: < 2 USD/tháng

(với mức sử dụng phù hợp sinh viên, giới hạn lưu trữ và số lần truy cập nhỏ)

6.2. Chi phí phát triển và tích hợp hệ thống

Dự án được phát triển bởi một thực tập sinh sử dụng máy cá nhân và tài nguyên học thuật. Do đó chi phí phát triển không tính theo đơn vị tiền tệ mà được quy đổi theo thời gian công sức.

Hạng mục	Mô tả	Ước tính chi phí
Phát triển mã nguồn	Viết script Python, huấn luyện ML	0 USD (Tự thực hiện)
Môi trường phát triển	Google Colab / SageMaker Studio Lab	Miễn phí
Công cụ trực quan hóa	Matplotlib / Google Sheets	Miễn phí
Trình soạn thảo và IDE	VSCode, Jupyter Notebook	Miễn phí

6.3. Chi phí vận hành định kỳ

Vì hệ thống không chạy liên tục theo thời gian thực mà chỉ cần kiểm tra theo chu kỳ (ví dụ mỗi ngày hoặc mỗi tuần), chi phí vận hành định kỳ gần như không đáng kể.

Hoạt động	Tần suất	Ước tính chi phí
Tiền xử lý dữ liệu CUR	1 lần/ngày hoặc tuần	0.00
Huấn luyện lại mô hình	1 lần/tháng hoặc khi cần	0.00
Gửi email cảnh báo	5–10 email/tháng	0.00

Tổng chi phí vận hành: ≈ 0 USD/tháng

6.4. Phân tích lợi tức đầu tư và thời gian hoàn vốn

Giả định:

- Sinh viên hoặc startup nhỏ có hóa đơn AWS hàng tháng khoảng **50 USD**
- Nếu không giám sát, có thể chi tiêu bất hợp lý hoặc lỗi cấu hình gây vượt ngưỡng đến **20% chi phí** (≈10 USD)

- Hệ thống phát hiện bất thường giúp **giảm thiểu >80% lãng phí phát sinh**

Tính toán ROI:

- **Chi phí đầu tư:** < 10 USD (1 lần, không định kỳ)
- **Chi phí vận hành:** ≈ 0 USD
- **Lợi ích hằng tháng tiềm năng:** tiết kiệm 8–10 USD nếu có lỗi phát sinh

Thời gian hoàn vốn (Break-even): 1–2 tháng sử dụng

ROI theo công thức:

ROI = (Lợi ích – Chi phí đầu tư) / Chi phí đầu tư

= (10 – 10) / 10 = 0% (tháng đầu)

= (10 – 0) / 10 = 100% (tháng thứ 2)

Kết luận: Dự án hoàn vốn ngay từ tháng thứ 2 nếu có 1 lỗi chi tiêu bất thường được phát hiện.

6.5. Chiến lược tối ưu chi phí

Để đảm bảo chi phí luôn nằm dưới mức ngân sách, dự án áp dụng các chiến lược:

- **Sử dụng dịch vụ trong Free Tier**: Tận dụng Lambda, S3, SNS miễn phí dưới mức ngưỡng
- **Huấn luyện mô hình cục bộ hoặc trên Studio Lab**: Tránh phát sinh chi phí từ SageMaker thật
- **Thực thi theo lịch định kỳ**: Chạy mô hình 1–2 lần/ngày thay vì real-time
- **Xóa tài nguyên sau khi dùng**: Tắt log, xóa file không cần thiết, tránh lưu trữ dư thừa
- **Giới hạn alert và log**: Chỉ gửi email khi vượt ngưỡng nghiêm trọng, tránh spam không cần thiết

Những biện pháp này giúp đảm bảo hệ thống luôn vận hành hiệu quả trong phạm vi ngân sách <10 USD.

PHẦN 7: ĐÁNH GIÁ VÀ QUẢN LÝ RỦI RO (RISK ASSESSMENT AND MITIGATION PLAN)

7.1 Nhận diện rủi ro kỹ thuật và vận hành

Trong quá trình triển khai một hệ thống phát hiện bất thường chi phí trên nền tảng AWS, đặc biệt với quy mô nhỏ và giới hạn nguồn lực (sinh viên thực hiện), các rủi ro có thể phát sinh bao gồm:

Nhóm rủi ro	Mô tả cụ thể
Rủi ro kỹ thuật	- Không tương thích định dạng CUR - Mô hình ML phát hiện sai - Giới hạn tài nguyên máy cá nhân
Rủi ro vận hành	- Email cảnh báo không gửi được - Lỗi khi triển khai Lambda - Không cập nhật dữ liệu đúng lịch
Rủi ro bảo mật	- Rò rỉ thông tin thanh toán nếu xử lý không đúng quyền hạn - Thiếu kiểm soát truy cập

Việc nhận diện sớm rủi ro giúp xây dựng kế hoạch dự phòng phù hợp, giảm thiểu ảnh hưởng tiêu cực đến tiến độ và chất lượng dự án.

7.2. Đánh giá tác động và xác suất xảy ra

Dưới đây là bảng đánh giá sơ bộ xác suất xảy ra và mức độ ảnh hưởng của từng rủi ro được xác định:

Rủi ro chính	Tác động (Impact)	Xác suất (Likelihood)	Ghi chú bổ sung
Không hiểu định dạng CUR của AWS	Trung bình	Cao	Gây chậm tiến độ xử lý dữ liệu
Mô hình ML phát hiện sai bất thường	Cao	Trung bình	Gây mất tin cậy hệ thống, cảnh báo sai
Không gửi được email cảnh báo	Trung bình	Thấp	Người dùng không được thông báo kịp thời
Lỗi triển khai Lambda / xử lý CSV	Trung bình	Trung bình	Dừng pipeline phát hiện bất thường
Giới hạn tài nguyên máy (RAM/CPU)	Thấp	Cao	Làm chậm huấn luyện mô hình
Sai quyền IAM, rò rỉ dữ liệu chi phí	Cao	Thấp	Vi phạm bảo mật, đặc biệt nếu dùng tài khoản thật

Phân loại mức độ rủi ro:

- **Cao:** Gây gián đoạn nghiêm trọng hoặc ảnh hưởng trực tiếp đến kết quả cuối cùng
- **Trung bình:** Có thể ảnh hưởng cục bộ nhưng có thể khắc phục
- **Thấp:** Ít ảnh hưởng, dễ xử lý bằng công cụ có sẵn

7.3. Ma trận ưu tiên rủi ro và chiến lược giảm thiểu

Ma trận ưu tiên rủi ro (Risk Matrix)

	Xác suất cao	Xác suất trung bình	Xác suất thấp
Tác động cao	[1] Mô hình sai		[2] Sai quyền IAM
Tác động trung bình	[3] Không hiểu CUR	[4] Lỗi Lambda	[5] Gửi mail lỗi

	Xác suất cao	Xác suất trung bình	Xác suất thấp
Tác động thấp	[6] Máy yếu		

Chiến lược giảm thiểu (Mitigation Strategies)

Mã Rủi ro chính		Biện pháp giảm thiểu cụ thể
1	Mô hình phát hiện sai	- Tinh chỉnh threshold bất thường - Dùng tập dữ liệu kiểm thử đa dạng
2	Sai quyền IAM	- Gán quyền tối thiểu (least privilege) - Hạn chế dùng root account
3	Không hiểu CUR format	- Học từ tài liệu AWS Billing - Xem mẫu thực tế - Thử nghiệm từng phần
4	Lỗi Lambda hoặc CSV	- Debug với log CloudWatch - Viết test case cho input
5	Gửi email thất bại	- Tích hợp Gmail SMTP dự phòng - Ghi log gửi lại sau
6	Tài nguyên máy hạn chế	- Dùng Studio Lab hoặc Colab để huấn luyện thay vì máy cá nhân

7.4. Kế hoạch dự phòng và quy trình giám sát

Kế hoạch dự phòng (Contingency Plan)

- Thay thế công cụ: Nếu SageMaker Studio Lab ngừng hoạt động → chuyển sang Google Colab
- Backup mô hình: Lưu checkpoint mô hình đã huấn luyện định kỳ

- Thủ công hóa: Khi pipeline lỗi → có thể chạy script phát hiện offline trên máy tính cá nhân
- Gửi lại thông báo: Khi email lỗi → có thể gửi lại bằng tay hoặc sử dụng backup SMTP

Quy trình giám sát và phản hồi

Hoạt động giám sát	**Công cụ/Phương pháp**	**Tần suất**
Ghi log hoạt động Lambda	AWS CloudWatch	Mỗi lần chạy
Theo dõi mô hình ML	Kiểm tra bằng file test mẫu	Sau mỗi huấn luyện
Theo dõi email	Xem log SMTP hoặc SNS	Hằng tuần
Theo dõi chi phí AWS	Billing Dashboard + alert (nếu có)	Theo dõi hàng ngày hoặc tuần

Nguyên tắc: Mọi cảnh báo lỗi phải được log lại và phản hồi trong vòng 24 giờ để đảm bảo tính liên tục của hệ thống.

PHẦN 8. KẾT QUẢ KỲ VỌNG VÀ TÁC ĐỘNG DÀI HẠN (EXPECTED OUTCOMES AND STRATEGIC IMPACT)

8.1. Các chỉ số đo lường thành công

Để đánh giá hiệu quả thực tiễn của hệ thống phát hiện bất thường chi phí trên AWS, một tập hợp các chỉ số định lượng và định tính được sử dụng như sau:

Chỉ số đo lường	**Mục tiêu**	**Cách đo**
Độ chính xác mô hình phát hiện	≥ 80% (Precision và Recall)	So sánh nhãn gán thủ công với kết quả mô hình

Chỉ số đo lường	Mục tiêu	Cách đo
Thời gian phản hồi cảnh báo	≤ 10 phút từ lúc CUR được cập nhật	Tính bằng timestamp cảnh báo
Tỷ lệ giảm chi phí bất thường	≥ 10% chi phí bị lãng phí được phát hiện và xử lý	Trước/sau khi áp dụng
Tính khả dụng của hệ thống	≥ 95% thời gian hoạt động không lỗi	Log giám sát CloudWatch
Tỷ lệ cảnh báo sai (false positive)	≤ 15%	Phân tích thủ công kết quả mô hình

Các chỉ số trên giúp đảm bảo hệ thống không chỉ hoạt động về mặt kỹ thuật mà còn mang lại giá trị đo lường được về tài chính và hiệu quả vận hành.

8.2. Lợi ích ngắn hạn và trung hạn

a) Ngắn hạn (0–6 tháng)

- Tự động hóa giám sát chi phí: Giảm phụ thuộc vào thao tác thủ công trong việc theo dõi bảng chi phí.
- Tăng tốc phát hiện bất thường: Thay vì 2–3 ngày phát hiện sự cố chi tiêu, hệ thống cho phép nhận diện trong vài phút.
- Hạn chế thất thoát ngân sách: Loại bỏ các khoản tiêu dùng do lỗi cấu hình (ví dụ: instance không tắt).
- Tăng nhận thức FinOps cho sinh viên/cá nhân: Hiểu rõ hơn về dòng chi phí AWS, từ đó tối ưu hóa sử dụng.

b) Trung hạn (6–18 tháng)

- Mở rộng đa người dùng: Cho phép tích hợp giám sát nhiều tài khoản nhỏ cho nhóm sinh viên hoặc dự án startup.
- Báo cáo chi tiết theo lịch: Hệ thống sinh báo cáo định kỳ (PDF hoặc Google Sheet) giúp dễ kiểm toán và theo dõi.
- Tinh chỉnh mô hình nâng cao: Sử dụng thêm thuật toán học sâu (Deep Learning) hoặc clustering cho các tập dữ liệu phức tạp hơn.

8.3. Giá trị dài hạn đối với tổ chức

Hệ thống không chỉ giải quyết vấn đề ngắn hạn mà còn đóng vai trò quan trọng trong chiến lược quản lý chi phí dài hạn trên đám mây:

- Tích hợp vào hệ thống FinOps toàn diện: Có thể mở rộng thành một module trong hệ thống quản lý ngân sách AWS quy mô lớn.
- Cơ sở dữ liệu học máy nội bộ: Qua thời gian, hệ thống sẽ tích lũy tập dữ liệu về chi phí, phục vụ cho phân tích xu hướng hoặc dự báo ngân sách.
- Chuẩn hóa quy trình phát hiện chi phí bất thường: Giảm thiểu phụ thuộc vào từng cá nhân, giúp quy trình minh bạch và có thể lặp lại.

8.4. Nâng cao trải nghiệm người dùng và năng lực chiến lược

a) Nâng cao trải nghiệm người dùng:

- Giao diện thân thiện: Qua bảng báo cáo trực quan hoặc tích hợp Google Sheets/Colab, người dùng dễ hiểu kết quả phát hiện.
- Tương tác tức thời: Khi có cảnh báo, người dùng nhận được email giải thích chi tiết về dịch vụ AWS gây bất thường.
- Không cần chuyên môn sâu: Người không chuyên về ML hay Cloud vẫn có thể sử dụng hệ thống một cách hiệu quả.

b) Gia tăng năng lực chiến lược:

- Tăng năng lực kiểm soát tài chính công nghệ: Đặc biệt với các tổ chức mới hoặc sinh viên đang làm quen AWS.
- Giảm rủi ro lạm phát ngân sách trong học tập và R&D: Đặc biệt hữu ích trong môi trường giáo dục hoặc tổ chức học thuật.
- Khả năng nhân rộng và đào tạo: Dự án có thể dùng làm mẫu để hướng dẫn sinh viên khác triển khai, nâng cao kỹ năng về AWS, Python, ML.