

Bộ Giáo Dục Và Đào Tạo
Trường Đại Học Ngoại Ngữ - Tin Học Thành Phố Hồ Chí Minh
Khoa Công Nghệ Thông Tin



**BÁO CÁO KẾT THÚC HỌC PHẦN
ĐIỆN TOÁN ĐÁM MÂY**

**ĐỀ TÀI : XÂY DỰNG VÀ TRIỂN KHAI ỨNG DỤNG WEB TĨNH
TRÊN GOOGLE CLOUD PLATFORM**

Giảng Viên Hướng Dẫn : ThS. Cao Tiên Thành

Học viên thực hiện:

- | | | |
|----|--------------------|------------|
| 1. | Phạm Đức Long | 22DH114617 |
| 2. | Nguyễn Vũ Anh Khoa | 22DH111684 |

Tp. Hồ Chí Minh - Ngày tháng năm 2025

LỜI CẢM ƠN

Để hoàn thành bản báo cáo đồ án tốt nghiệp với đề tài "Triển khai Ứng dụng Web Tính trên Google Cloud Platform" này, em đã nhận được sự quan tâm, giúp đỡ tận tình từ nhiều cá nhân và tổ chức.

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc đến thầy Th.S Cao Tiến Thành đã luôn tận tình hướng dẫn, chỉ bảo, truyền đạt kiến thức và kinh nghiệm quý báu trong suốt quá trình em thực hiện đồ án. Những góp ý và định hướng kịp thời của Thầy là nguồn động lực và nền tảng vững chắc giúp em vượt qua những khó khăn, hoàn thiện đồ án một cách tốt nhất.

Em cũng xin chân thành cảm ơn quý Thầy/Cô trong khoa Công nghệ Thông Tin và Trường Đại Học Ngoại - Ngữ Tin Học đã tạo điều kiện thuận lợi về cơ sở vật chất, tài liệu và môi trường học tập, nghiên cứu để em/tôi có thể hoàn thành đồ án này.

Cuối cùng, em xin gửi lời cảm ơn đến gia đình, bạn bè và những người thân đã luôn động viên, khích lệ và là chỗ dựa tinh thần vững chắc trong suốt quá trình em học tập và thực hiện đồ án.

Mặc dù đã có nhiều cố gắng, nhưng chắc chắn bản báo cáo này vẫn không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ quý Thầy và bạn bè để đồ án được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIẢNG VIÊN

Điểm: _____ (bằng chữ: _____)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án ?

, ngày tháng năm 2024

GIÁO VIÊN HƯỚNG DẪN

(ky, ho tên)

LƯU TRỮ CÁC THAY ĐỔI

*A - Added M - Modified D - Deleted

Ngày tháng	Phần tử thay đổi	A* M, D	Mô tả thay đổi	New Version

BẢNG CHỮ KÝ

Tác giả:

Tên: _____

Chữ ký: _____

Vị trí: _____

Ngày: _____

Tên:

Chữ ký: _____

Vị trí: _____

Ngày: _____

Tên:

Chữ ký: _____

Vị trí: _____

Ngày: _____

Người điều chỉnh:

Tên: _____

Chữ ký: _____

Vị trí: _____

Ngày: _____

Người duyệt:

Tên: _____

Chữ ký: _____

Vị trí: _____

Ngày: _____

MỤC LỤC

LỜI CẢM ƠN	2
CHƯƠNG I. TỔNG QUAN VỀ ĐỒ ÁN VÀ CÔNG NGHỆ CLOUD COMPUTING	12
1.1. Giới thiệu Đồ án	12
1.1.1. Mục tiêu của đồ án	12
1.1.2. Phạm vi đồ án	12
1.1.3. Các công nghệ chính được sử dụng	13
1.2. Tổng quan về Cloud Computing	15
1.2.1. Định nghĩa Cloud Computing	15
1.2.2. Các mô hình dịch vụ (IaaS, PaaS, SaaS)	16
1.2.3. Các mô hình triển khai (Public, Private, Hybrid)	18
1.3. Giới thiệu về Google Cloud Platform (GCP)	22
1.3.1. Vai trò của GCP trong Cloud Computing	22
1.3.2. Các dịch vụ chính của GCP liên quan đến đồ án	23
1.3.3. Google Cloud SQL/Firebase: Dịch vụ database quản lý	24
1.4. Các công nghệ và công cụ hỗ trợ sử dụng	25
1.4.1. Ubuntu Server	25
1.4.2. Nginx Web Server	26
1.4.3. HTML/CSS/JavaScript	26
1.4.4. SSH Client và Google Cloud SDK	28
CHƯƠNG II. CƠ SỞ LÝ THUYẾT VÀ CẤU HÌNH HỆ THỐNG	30
2.1. Kiến trúc tổng thể hệ thống	30
2.1.1. Sơ đồ kiến trúc đề xuất	30
2.1.2. Giải thích các thành phần trong kiến trúc	30
2.2. Các khái niệm cơ bản trên Google Cloud Platform	31
2.2.1. Google Compute Engine (GCE)	31
2.2.2. Google Cloud SQL/ Firestore	37
2.2.3. Virtual Private Cloud (VPC)	38
2.2.4. Google Cloud Storage (GCS)	39
2.2.5. Identity and Access Management (IAM)	41
2.3. Cơ sở lý thuyết về Web Server Nginx	42
2.3.1. Cấu hình cơ bản của Nginx cho trang web tĩnh	42
2.3.2. Giới thiệu về ngôn ngữ/framework Backend được sử dụng	43
2.4. Chuẩn bị môi trường phát triển và triển khai	44
2.4.1. Các bước tạo tài khoản GCP và dự án mới	44
2.4.2. Cài đặt và cấu hình Google Cloud SDK	45
2.4.3. Chuẩn bị mã nguồn ứng dụng web động	45

CHƯƠNG III. TRIỂN KHAI DỊCH VỤ WEB TRÊN GCP	47
3.1. Tạo và cấu hình Virtual Machine trên Compute Engine	47
3.1.1. Lựa chọn Region, Zone, Machine type và Disk Image	47
3.1.2. Cấu hình Firewall rules cho VM	50
3.1.3. Thiết lập địa chỉ IP tĩnh (External IP)	51
3.2. Triển khai và cấu hình Database trên GCP	51
3.2.1. Tạo và cấu hình Instance Cloud SQL (hoặc Firestore Database)	51
3.2.2. Thiết lập kết nối an toàn từ VM đến Database	54
3.3. Cài đặt và cấu hình Nginx trên VM Ubuntu	61
3.3.1. Kết nối SSH đến VM và cập nhật hệ thống	61
3.3.2. Cài đặt Nginx Web Server	64
3.4. Tải mã nguồn web tĩnh lên VM	65
3.4.1. Phương pháp tải mã nguồn (scp, gcloud compute scp)	65
3.4.2. Đặt mã nguồn vào thư mục được cấu hình bởi Nginx	66
3.5. Triển khai Máy chủ Admin riêng biệt và Kết nối Database	69
3.5.1. Tạo và cấu hình VM cho máy chủ Admin	69
3.5.2. Cài đặt và cấu hình Cloud SQL Auth Proxy trên VM Admin	69
3.5.3. Triển khai Ứng dụng Admin Panel (Node.js/AdminJS)	71
3.6. Kiểm tra chức năng qua giao diện Website chính	74
3.5.1. Truy cập ứng dụng web thông qua trình duyệt	74
3.5.2. Hiển thị sản phẩm từ Database	74
3.5.2. Kiểm tra log Nginx và hệ thống	75
CHƯƠNG IV. BẢO MẬT VÀ SAO LUU/PHỤC HỒI DỮ LIỆU TRÊN GOOGLE CLOUD PLATFORM	76
4.1. Các công cụ bảo mật trên Google Cloud Platform	76
4.1.1. VPC Firewall Rules	76
4.1.2. Triển khai HTTPS với chứng chỉ SSL/TLS và tên miền	76
4.1.3. Một số gợi ý bảo mật khác	81
4.2. Backup và Restore dữ liệu trên Cloud Platform	84
4.2.1. Logic chung về Backup và Restore	84
4.2.2. Chiến lược Backup cho ứng dụng web tĩnh trên GCP	84
4.2.3. Backup Database Cloud SQL (Tính năng tự động của GCP):	86
CHƯƠNG V. ĐÁNH GIÁ, KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	89
5.1. Đánh giá kết quả đạt được	89
5.1.1. Những mục tiêu đề án đã hoàn thành	89
5.1.2. Đánh giá hiệu suất và độ ổn định của hệ thống	89
5.1.3. Ưu điểm và nhược điểm khi triển khai trên GCP	90
5.2. Khó khăn gặp phải và giải pháp	90

5.2.1. Các vấn đề kỹ thuật phát sinh	90
5.2.2. Cách thức giải quyết các vấn đề	91
5.3. Hướng phát triển trong tương lai	91
5.3.1. Tích hợp và triển khai Database trên Cloud (Cloud SQL, Firestore)	92
5.3.2. Mở rộng hệ thống với Load Balancing và Auto-scaling	92
5.3.3. Tự động hóa quy trình triển khai (CI/CD với Cloud Build)	93
KẾT LUẬN	94
TÀI LIỆU THAM KHẢO	94
VIDEO MINH HỌA VÀ TRIỂN KHAI HỆ THỐNG	94
BẢNG PHÂN CÔNG	94

DANH MỤC BẢNG

BẢNG 1. BẢNG SO SÁNH IAAS, PAAS, SAAS	17
BẢNG 2. SO SÁNH CÁC MÔ HÌNH TRIỂN KHAI TRÊN CLOUD COMPUTING	21
BẢNG 3. SO SÁNH CÁC DÒNG MÁY ẢO CHÍNH CỦA GCE	33
BẢNG 4. SO SÁNH CÁC LOẠI DISK VÀ IMAGE TRONG GCE	36

DANH MỤC HÌNH ẢNH

HÌNH 1. TỔNG QUAN VỀ GOOGLE CLOUD PLATFORM	12
HÌNH 2. CÁC CÔNG NGHỆ ĐƯỢC SỬ DỤNG TRONG GCP	13
HÌNH 3. CLOUD COMPUTING	15
HÌNH 4. CÁC MÔ HÌNH DỊCH VỤ CLOUD COMPUTING	16
HÌNH 5. CÁC MÔ HÌNH TRIỂN KHAI TRÊN CLOUD COMPUTING	18
HÌNH 6. GIỚI THIỆU VAI TRÒ CỦA GCP TRONG CLOUD COMPUTING	22
HÌNH 7. DỊCH VỤ DATABASE TRONG GOOGLE CLOUD SQL	24
HÌNH 8. UBUNTU TRONG GOOGLE CLOUD PLATFORM	25
HÌNH 9. HTML, CSS, JAVASCRIPT TRONG GCP	27
HÌNH 10. SSH CLIENT VÀ GOOGLE CLOUD SDK TRONG GCP	28
HÌNH 11. SƠ ĐỒ KIẾN TRÚC	30
HÌNH 12. HÌNH ẢNH GOOGLE COMPUTE ENGINE	31
HÌNH 13. CÁC DỊCH VỤ CLOUD STORAGE TRÊN GCP	37
HÌNH 14. VIRTUAL PRIVATE CLOUD	38
HÌNH 15. STORAGE CLASSES TRONG GCP	40
HÌNH 16. NGÔN NGỮ NODE.JS TRONG GOOGLE CLOUD PLATFORM	43
HÌNH 17. CÁC BƯỚC TẠO DỰ ÁN MỚI TRONG GCP	44
HÌNH 18. THIẾT LẬP KHU VỰC (REGION) VÀ VÙNG (ZONE)	47
HÌNH 19. LỰA CHỌN LOẠI MÁY ĐỂ THIẾT LẬP MÁY ẢO	48
HÌNH 20. THIẾT LẬP BOOT DISK IMAGE UBUNTU	49
HÌNH 21. CẤU HÌNH THIẾT LẬP FIREWALL RULES TRONG GCP	50
HÌNH 22. THIẾT LẬP ĐỊA CHỈ IP TÍNH VM INSTANCES	51
HÌNH 23. CREATE INSTANCE CLOUD SQL	51
HÌNH 24. CẤU HÌNH CỦA SQL CLOUD	52
HÌNH 25. TẠO BẢNG DATABASE TRONG CLOUD SQL	53
HÌNH 26. TẠO USER CHO NGƯỜI DÙNG TRONG SQL CLOUD	53
HÌNH 27. CÀI ĐẶT VÀ THIẾT LẬP KẾT NỐI AUTH-PROXY.	55
HÌNH 28. CẤU HÌNH DỊCH VỤ SYSTEMD CHO CLOUD SQL AUTH PROXY	56
HÌNH 29. KIỂM TRA TRẠNG THÁI DỊCH VỤ CLOUD SQL AUTH PROXY	56
HÌNH 30. QUẢN LÝ VÀ KIỂM TRA HOẠT ĐỘNG TRẠNG THÁI ỨNG DỤNG VỚI PM2.	57
HÌNH 31. CẤU HÌNH FILE .ENV CHO WEBSITE	58
HÌNH 32. GIAO DIỆN WEBSITE VÀ DỮ LIỆU NGƯỜI DÙNG CHỨNG MINH KẾT NỐI DATABASE	58
HÌNH 33. KHỞI CHẠY CLOUD SQL AUTH PROXY TRÊN MÁY TÍNH CỤC BỘ	59
HÌNH 34. KẾT NỐI DATABASE CLOUD SQL VỚI MYSQL MÁY CỤC BỘ	60
HÌNH 35. GIAO DIỆN QUẢN LÝ DATABASE USER_AUTH_DB TRONG MYSQL WORKBENCH	61
HÌNH 36. SỬ DỤNG GOOGLE CLOUD SDK KẾT NỐI ĐẾN SSH CỦA VM INSTANCE	62
HÌNH 37. SỬ DỤNG SSH TRONG GOOGLE CLOUD CONSOLE	63

HÌNH 38. CẬP NHẬP VÀ NÂNG CẤP CÁC GÓI TRÊN VM INSTANCE	63
HÌNH 39. CÀI ĐẶT NGINX TRÊN SSH	64
HÌNH 40. TẢI MÃ NGUỒN WEB TĨNH BẰNG GCLOUD LÊN VM	65
HÌNH 41. MÃ NGUỒN WEB TĨNH TRÊN NGINX	66
HÌNH 42. SỬA TỆP CẤU HÌNH NGINX MẶC ĐỊNH	67
HÌNH 43. CÁC ĐẶC QUYỀN SỞ HỮU VÀ QUYỀN TRUY CẬP CHO NGINX	68
HÌNH 44. TẠO MÁY ẢO UBUNTU CHO TRANG ADMIN	69
HÌNH 45. KHỞI CHẠY CLOUD SQL AUTH PROXY TRÊN MÁY CHỦ ADMIN	70
HÌNH 46. TẠO KẾT NỐI AUTH-PROXY ĐẾN ADMINSERER.JS	70
HÌNH 47. TRIỂN KHAI ADMINSERVER.JS TRÊN MÁY UBUNTU CỦA GOOGLE	71
HÌNH 48. KHỞI CHẠY VÀ CẤU HÌNH TỰ ĐỘNG KHỞI ĐỘNG TIẾN TRÌNH NODE.JS BẰNG PM2	71
HÌNH 49. CẤU HÌNH FILE .ENV CHO TRANG AMIN	72
HÌNH 50. KIỂM TRA KẾT NỐI DATABASE VỚI TRANG ADMIN	73
HÌNH 51. TRANG ADMIN PANEL ĐÃ ĐƯỢC TRUY XUẤT THÀNH CÔNG	73
HÌNH 52. TRUY CẬP ỨNG DỤNG WEB THÔNG QUA TRÌNH DUYỆT	74
HÌNH 53. HIỂN THỊ SẢN PHẨM Ở WEBSITE CÓ TRONG DATABASE	74
HÌNH 54. KIỂM TRA LOG TRUY CẬP	75
HÌNH 55. CẤU HÌNH IP PRIVATE CHO CLOUD SQL	76
HÌNH 56. BẢNG ĐIỀU KHIỂN QUẢN LÝ CHỨNG CHỈ SSL (LET'S ENCRYPT) CHO TÊN MIỀN	77
HÌNH 57. CHỨNG CHỈ SSL CERT CẤP CHO TÊN MIỀN	78
HÌNH 58. KHÓA RIÊNG TƯ SSL PRIVATE KEY CẤP CHO TÊN MIỀN	78
HÌNH 59. CẤU HÌNH NGINX (DEFAULT) TRÊN MÁY CHỦ UBUNTU	79
HÌNH 60. KẾT QUẢ THỰC THI ĐƯỢC HTTPS CHO TRANG WEB	80
HÌNH 61. KẾT QUẢ THỰC THI ĐƯỢC HTTPS CHO TRANG ADMIN	80
HÌNH 62. LỆNH CÀI ĐẶT SURUCATA TRÊN UBUNTU TRONG GCP	81
HÌNH 63. CẤU HÌNH SURICATA.YAML VỚI VIỆC THIẾT LẬP GIAO DIỆN MẠNG	82
HÌNH 64. FILE MY.RULES CHỨA CÁC QUY TẮC CHO SURICATA	82
HÌNH 65. KẾT QUẢ PING ĐƯỢC THỰC HIỆN TỪ WINDOWS	83
HÌNH 66. SURICATA CHẠY VÀ LOG CẢNH BÁO	83
HÌNH 67. HIỂN THỊ NỘI DUNG BACKUP.SH VÀ LƯU TRỮ MÃ NGUỒN ỨNG DỤNG	84
HÌNH 68. CẤU HÌNH CRONTAB VÀ LÊN LỊCH CHẠY SCRIPT BACKUP.	85
HÌNH 69. THƯ MỤC BACKUP, CHỨNG MINH CÁC FILE BACKUP ĐÃ ĐƯỢC TẠO THÀNH CÔNG.	86
HÌNH 70. TỔNG QUAN CÁC BẢN SAO LƯU CỦA CLOUD SQL, LIỆT KÊ CÁC INSTANCE	86
HÌNH 71. CÀI ĐẶT SAO LƯU TỰ ĐỘNG CHO MỘT INSTANCE CLOUD SQL	87
HÌNH 72. CÁC BẢN SAO LƯU ĐÃ TẠO CHO MỘT INSTANCE CLOUD SQL VÀ GỒM THỜI GIAN TẠO	88
HÌNH 73. GIAO DIỆN PHỤC HỒI INSTANCE TỪ MỘT BẢN SAO LƯU.	88
HÌNH 74 : HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI	92

CHƯƠNG I. TỔNG QUAN VỀ ĐỒ ÁN VÀ CÔNG NGHỆ CLOUD COMPUTING

1.1. Giới thiệu Đồ án

1.1.1. Mục tiêu của đồ án



Hình 1. Tổng quan về Google Cloud Platform

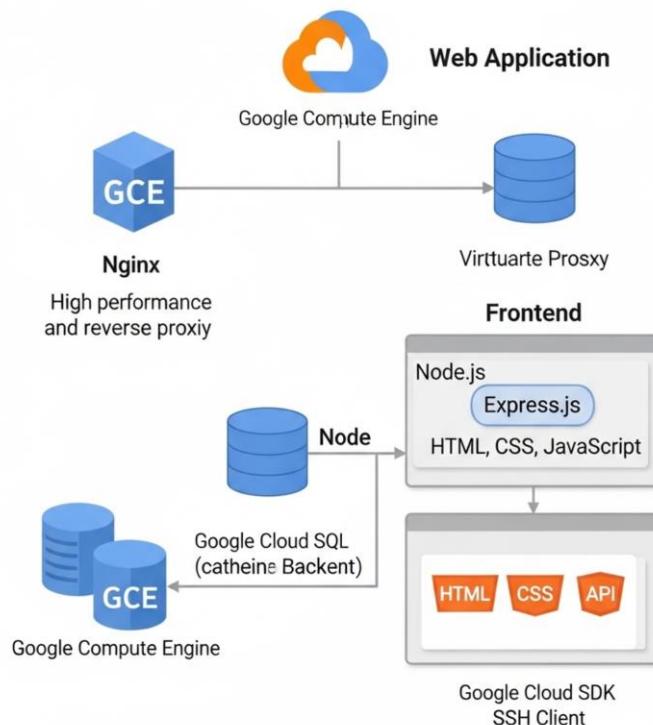
- Mục tiêu chính của đồ án điện toán đám mây là giúp người học hiểu về điện toán đám mây, các khái niệm, kiến trúc, và ứng dụng thực tế của nó. Đồng thời, đồ án cũng có thể tập trung vào việc triển khai các dịch vụ đám mây, như cơ sở hạ tầng (IaaS), nền tảng (PaaS), hoặc phần mềm (SaaS), trên các môi trường ảo.

1.1.2. Phạm vi đồ án

- Phạm vi của đồ án này được xác định rõ ràng để tập trung vào các mục tiêu chính về xây dựng và triển khai ứng dụng web động trên Google Cloud Platform (GCP). Cụ thể, đồ án sẽ bao gồm các khía cạnh sau:
 - Ứng dụng Web động: Phát triển một ứng dụng web với cả frontend (HTML, CSS, JavaScript) và backend (sử dụng Node.js/Express.js) có khả năng tương tác dữ liệu.
 - Triển khai trên GCP: Sử dụng Google Compute Engine (GCE) làm máy chủ ứng dụng với Nginx làm reverse proxy. Google Cloud SQL (hoặc Firestore) sẽ được dùng làm dịch vụ cơ sở dữ liệu chính.

- Hạ tầng và Bảo mật cơ bản: Thiết lập mạng Virtual Private Cloud (VPC) với các quy tắc tường lửa (Firewall Rules) và quản lý quyền truy cập bằng Identity and Access Management (IAM).
- Sao lưu và Phục hồi: Triển khai các chiến lược sao lưu cho mã nguồn và cơ sở dữ liệu, cùng quy trình khôi phục cơ bản.

1.1.3. Các công nghệ chính được sử dụng



Hình 2. Các công nghệ được sử dụng trong GCP

- Đề án này được xây dựng và triển khai dựa trên sự kết hợp của các công nghệ và dịch vụ hàng đầu trong lĩnh vực phát triển web và điện toán đám mây. Các công nghệ chính bao gồm:
 - **Google Cloud Platform (GCP):** Là nền tảng điện toán đám mây trung tâm, cung cấp các dịch vụ hạ tầng cần thiết.
 - Google Compute Engine (GCE): Dịch vụ máy ảo (VM) được sử dụng để host ứng dụng backend và web server.

- Google Cloud SQL: Dịch vụ cơ sở dữ liệu quan hệ được quản lý hoàn toàn, đóng vai trò lưu trữ và quản lý dữ liệu cho ứng dụng (ví dụ: MySQL hoặc PostgreSQL).
- Virtual Private Cloud (VPC): Cung cấp khả năng cấu hình mạng ảo, bao gồm các quy tắc tường lửa (Firewall Rules) để kiểm soát truy cập và bảo mật hệ thống.
- Identity and Access Management (IAM): Quản lý quyền truy cập và vai trò cho người dùng và các dịch vụ trong dự án GCP.
- Google Cloud Storage (GCS): Dịch vụ lưu trữ đối tượng được sử dụng để sao lưu mã nguồn và dữ liệu (nếu cần).

- **Hệ điều hành :**

- Ubuntu Server: Một bản phân phối Linux phổ biến, được lựa chọn làm hệ điều hành cho máy ảo trên GCE nhờ tính ổn định, bảo mật và cộng đồng hỗ trợ mạnh mẽ.
- Web Server: Công cụ Nginx đóng vai trò là web server hiệu suất cao và reverse proxy, tiếp nhận các yêu cầu từ người dùng, phục vụ nội dung tĩnh và chuyển tiếp các yêu cầu động đến ứng dụng backend.

- **Ngôn ngữ và Framework Phát triển Ứng dụng:**

- Node.js : Môi trường runtime chính để phát triển logic xử lý phía máy chủ của ứng dụng.
- Express.js: Framework web được sử dụng để xây dựng API, quản lý routing và xử lý các yêu cầu HTTP trong ứng dụng backend.
- HTML/CSS/JavaScript: Các công nghệ nền tảng dùng để xây dựng giao diện người dùng (frontend) của ứng dụng, đảm bảo tính tương tác và hiển thị trên trình duyệt web.

- **Công cụ Hỗ trợ Triển khai:**

- SSH Client: Công cụ cần thiết để kết nối bảo mật từ máy tính cục bộ đến máy ảo trên GCE để cấu hình và quản lý.

- Google Cloud SDK: Bộ công cụ dòng lệnh (CLI) cung cấp các lệnh **gcloud** mạnh mẽ để tương tác và quản lý các dịch vụ GCP một cách hiệu quả.

1.2. Tổng quan về Cloud Computing

1.2.1. Định nghĩa Cloud Computing



Hình 3. Cloud Computing

- **Cloud Computing** là mô hình cung cấp các dịch vụ điện toán thông qua internet. Là một trong những xu hướng công nghệ làm thay đổi cách thức chúng ta lưu trữ và truy cập dữ liệu, ứng dụng và các dịch vụ trực tuyến, thay vì phải đầu tư xây dựng và quản lý các máy chủ vật lý và hạ tầng riêng (on-premise). [1]

1.2.2. Các mô hình dịch vụ (IaaS, PaaS, SaaS)



Hình 4. Các mô hình dịch vụ Cloud Computing

- IaaS, PaaS và Saas là ba mô hình phổ biến trong lĩnh vực điện toán đám mây (Cloud Computing). Mỗi mô hình đều có đặc điểm và lợi ích riêng, giúp các doanh nghiệp tùy chọn giải pháp phù hợp theo nhu cầu. Doanh nghiệp cũng có thể sử dụng kết hợp cả ba mô hình này để đáp ứng nhu cầu kinh doanh linh hoạt.[2]
- **IaaS (Infrastructure as a Service)**
 - IaaS là mô hình dịch vụ điện toán đám mây cung cấp cơ sở hạ tầng như máy chủ ảo, lưu trữ, và mạng qua internet. Người dùng có thể quản lý các tài nguyên này mà không cần đầu tư vào phần cứng hay duy trì hạ tầng tại chỗ. Điều này giúp doanh nghiệp linh hoạt hơn trong việc mở rộng và quản lý tài nguyên CNTT.[2]
- **PaaS (Platform as a Service)**
 - PaaS là mô hình dịch vụ điện toán đám mây cung cấp một nền tảng phát triển và triển khai ứng dụng. Nó giúp các nhà phát triển tập trung vào việc xây dựng sản phẩm mà không phải lo lắng về hạ tầng. Bên cạnh đó, mô hình này còn cung cấp công cụ lập trình, thư viện, và môi trường để phát triển ứng dụng dễ dàng và nhanh chóng.[2]

- **SaaS (Software as a Service)**

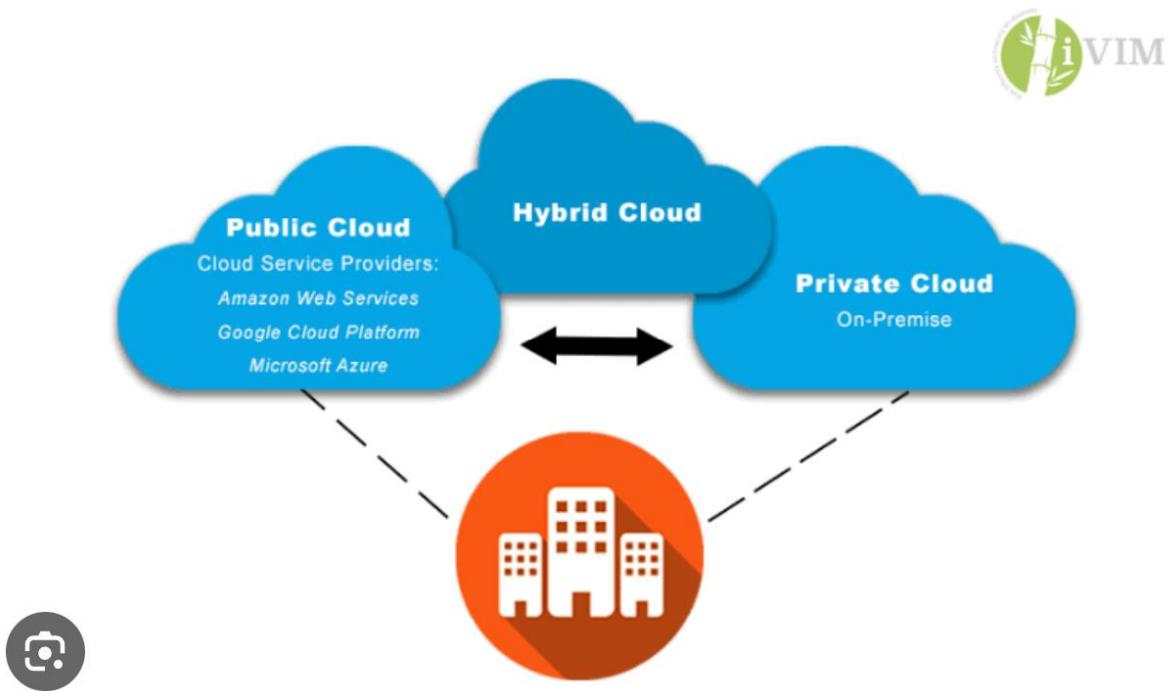
- SaaS là mô hình dịch vụ điện toán đám mây cung cấp các ứng dụng phần mềm qua internet. Người dùng chỉ cần truy cập ứng dụng qua trình duyệt web mà không cần cài đặt hoặc quản lý phần mềm trên máy tính cá nhân. Nhà cung cấp dịch vụ sẽ chịu trách nhiệm quản lý, cập nhật, và bảo trì ứng dụng.[2]

So sánh các mô hình dịch vụ đám mây (IaaS, PaaS, SaaS)

Tiêu chí	IaaS	PaaS	SaaS
Bạn tự quản lý	Ứng dụng, dữ liệu, runtime, Middleware, HĐH	Ứng dụng, Dữ liệu	Dữ liệu (tức là nội dung bạn tạo ra)
Đám mây quản lý	Ảo hóa, máy chủ, lưu trữ, mạng	Runtime, Middleware, HĐH, Ảo hóa, Máy chủ, Lưu trữ, Mạng	Mọi thứ (Ứng dụng, Runtime, Middleware, HĐH, Ảo hóa, Máy chủ, Lưu trữ, Mạng)
Ví dụ	Google Cloud Engine, AWS EC2	Google App Engine, Heroku	Gmail, Google Docs, Salesforce, Zoom
Ai là người dùng?	Kỹ sư hệ thống, Quản trị viên	Lập trình viên, Nhà phát triển	Người dùng cuối, Doanh nghiệp
Ưu điểm	Linh hoạt cao, kiểm soát toàn diện	Phát triển nhanh, không lo hạ tầng	Dễ dàng, sẵn có, không cần cài đặt
Nhược điểm	Cần kiến thức quản trị, tự bảo trì HĐH	Ít kiểm soát, có thể bị "khóa" với nhà cung cấp	Hạn chế tùy biến, phụ thuộc hoàn toàn vào nhà cung cấp

Bảng 1. Bảng so sánh IaaS, Paas, Saas

1.2.3. Các mô hình triển khai (Public, Private, Hybrid)



Hình 5. Các mô hình triển khai trên Cloud Computing

- Mô hình triển khai điện toán đám mây là cách mà các doanh nghiệp/ tổ chức thiết lập và sử dụng các tài nguyên đám mây như máy chủ, lưu trữ dữ liệu, phần mềm,... Các mô hình này quyết định nơi đặt các tài nguyên (có thể là bên trong công ty hoặc bên ngoài, ở một trung tâm dữ liệu của nhà cung cấp dịch vụ đám mây) và ai sẽ sở hữu, quản lý những tài nguyên đó (có thể là tổ chức hoặc nhà cung cấp dịch vụ). Đồng thời còn quyết định mục đích sử dụng và cách thức vận hành của hệ thống đám mây, ví dụ như liệu doanh nghiệp có thể chia sẻ dữ liệu với các tổ chức khác hay không.
- Hiện nay, có 4 mô hình triển khai điện toán đám mây phổ biến gồm Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud. Mỗi mô hình sẽ có những ưu và nhược điểm khác nhau. Việc hiểu rõ các mô hình này sẽ giúp tổ chức chọn lựa phương án phù hợp.[3]

- **Public Cloud – Đám mây công cộng**

- Public Cloud là loại đám mây mà bất kỳ ai cũng có thể sử dụng, miễn là có kết nối Internet. Điều này có nghĩa là các dịch vụ của điện toán đám mây và tài nguyên như máy chủ, lưu trữ dữ liệu, phần mềm... đều được cung cấp rộng rãi cho mọi người, doanh nghiệp, hoặc các nhóm ngành lớn. Tuy nhiên, vì đám mây công cộng mở cho tất cả mọi người, nên tính bảo mật không cao như các mô hình đám mây khác.[3]
- Trong mô hình này, cơ sở hạ tầng đám mây (máy chủ, phần cứng, phần mềm) do nhà cung cấp dịch vụ đám mây sở hữu và quản lý, không phải do người dùng sở hữu. Người dùng chỉ cần trả phí để sử dụng dịch vụ mà không phải lo về việc duy trì và bảo trì hạ tầng.

- **Private Cloud – Đám mây riêng**

- Private Cloud là mô hình đối ngược với đám mây công cộng. Đám mây riêng là một mô hình triển khai điện toán đám mây được thiết kế dành riêng cho một tổ chức duy nhất, đảm bảo rằng mọi tài nguyên, từ phần cứng, phần mềm đến dữ liệu, đều thuộc quyền kiểm soát và sử dụng của tổ chức đó.[3]
- Với mô hình này, tổ chức không phải chia sẻ tài nguyên với bất kỳ ai khác, giúp tăng cường bảo mật và quyền riêng tư. Tất cả hệ thống trong đám mây riêng đều được quản lý trong một môi trường an toàn, thường được bảo vệ bởi các tường lửa mạnh mẽ và được giám sát bởi bộ phận công nghệ thông tin (CNTT) nội bộ của tổ chức. Điều này cho phép tổ chức kiểm soát hoàn toàn hạ tầng và dữ liệu của mình, phù hợp cho những tổ chức cần xử lý thông tin nhạy cảm hoặc tuân thủ các quy định nghiêm ngặt về bảo mật.

- **Hybrid Cloud – Đám mây lai**

- Hybrid Cloud là mô hình kết hợp giữa đám mây công cộng và đám mây riêng, cho phép các tổ chức di chuyển dữ liệu và ứng dụng giữa các môi trường công cộng và riêng biệt. Mô hình này mang lại sự linh hoạt tối đa, giúp tổ chức có

thể tận dụng các ưu điểm của cả hai mô hình. Mô hình điện toán đám mây Hybrid Cloud giúp tối ưu hóa hiệu quả hoạt động cho doanh nghiệp.[3]

- **Community Cloud – Đám mây công cộng**

- Community Cloud là một mô hình đám mây mà tài nguyên được chia sẻ giữa các tổ chức có cùng lợi ích, yêu cầu bảo mật hoặc mục tiêu kinh doanh. Các tổ chức này có thể cùng nhau đầu tư vào cơ sở hạ tầng, hoặc nhờ một nhà cung cấp đám mây thứ ba quản lý hạ tầng chung. Đám mây cộng đồng thường được sử dụng trong các ngành công nghiệp có yêu cầu bảo mật cao như chăm sóc sức khỏe, giáo dục, và chính phủ.[3]

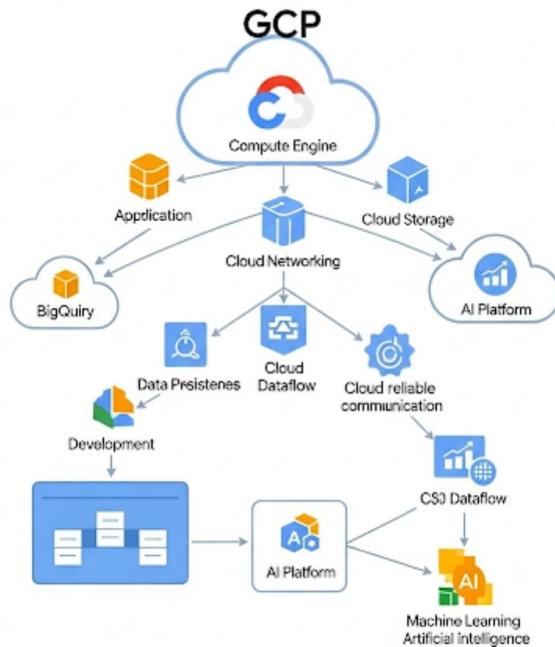
Bảng so sánh các mô hình triển khai điện toán đám mây:

Tiêu chí	Public Cloud	Private Cloud	Hybrid Cloud
Bản chất	Chia sẻ hạ tầng với nhiều khách hàng qua internet.	Hạ tầng đám mây dành riêng cho một tổ chức duy nhất.	Kết hợp Public và Private Cloud, linh hoạt di chuyển giữa hai.
Sở hữu / Quản lý	Nhà cung cấp đám mây.	Tổ chức sử dụng hoặc bên thứ ba quản lý riêng.	Tổ chức và nhà cung cấp đám mây cùng quản lý.
Kiểm soát	Thấp. Ít tùy chỉnh hạ tầng.	Cao nhất. Toàn quyền kiểm soát mọi thứ.	Trung bình. Kiểm soát cao phần riêng, linh hoạt phần chung.
Chi phí	Thấp ban đầu, trả theo dùng. Tối ưu cho mở rộng nhanh.	Rất cao ban đầu, chi phí vận hành cố định.	Kết hợp. Tối ưu chi phí bằng cách tận dụng cả hai.
Bảo mật	Phụ thuộc nhà cung cấp, phù hợp dữ liệu ít nhạy cảm.	Rất cao. Đáp ứng quy định nghiêm ngặt.	Tối ưu: Dữ liệu nhạy cảm ở Private, ít nhạy cảm ở Public.
Mở rộng	Rất cao và nhanh chóng.	Hạn chế, cần đầu tư thêm phần cứng.	Linh hoạt. Dùng Public cho tải đột biến, giữ Private cho ổn định.
Ví dụ	Gmail, website thông thường, Google Cloud, AWS, Azure.	Trung tâm dữ liệu riêng của ngân hàng, chính phủ.	Ứng dụng web với dữ liệu bí mật (Private) và web server công cộng (Public).
Ưu điểm	Rẻ, dễ dùng, mở rộng siêu nhanh.	Bảo mật cao nhất, kiểm soát toàn diện.	Linh hoạt tối đa, tối ưu cả bảo mật và chi phí.
Nhược Điểm	Ít kiểm soát, phụ thuộc nhà cung cấp.	Đắt đỏ, cần tự quản lý phức tạp.	Phức tạp khi thiết lập và quản lý tích hợp.

Bảng 2. So sánh các mô hình triển khai trên Cloud Computing

1.3. Giới thiệu về Google Cloud Platform (GCP)

1.3.1. Vai trò của GCP trong Cloud Computing



Hình 6. Giới thiệu vai trò của GCP trong Cloud Computing

- GCP cho phép bạn lựa chọn mức độ kiểm soát phù hợp với nhu cầu. Bạn có thể thuê máy ảo (IaaS) với Compute Engine để kiểm soát mọi thứ, hoặc dùng nền tảng phát triển (PaaS) như App Engine hay dịch vụ không máy chủ (Serverless) như Cloud Functions để tập trung hoàn toàn vào mã nguồn mà không cần quản lý hạ tầng. Điều này giúp các doanh nghiệp và nhà phát triển nhanh chóng đưa ý tưởng vào thực tế.
- GCP được xây dựng trên cùng hạ tầng bảo mật mà Google sử dụng cho các sản phẩm tỷ người dùng như Gmail hay YouTube. Với hệ thống bảo mật đa lớp, mã hóa dữ liệu mặc định, cùng các công cụ quản lý danh tính (IAM) và mạng (VPC Firewall Rules), GCP giúp bảo vệ dữ liệu và ứng dụng của bạn. Mạng lưới toàn cầu rộng khắp cũng đảm bảo ứng dụng luôn có sẵn và hoạt động ổn định.
- GCP áp dụng mô hình "trả tiền theo mức sử dụng" (pay-as-you-go), giúp bạn chỉ trả cho những gì bạn dùng, tránh lãng phí. Nền tảng này cũng giảm đáng kể gánh nặng

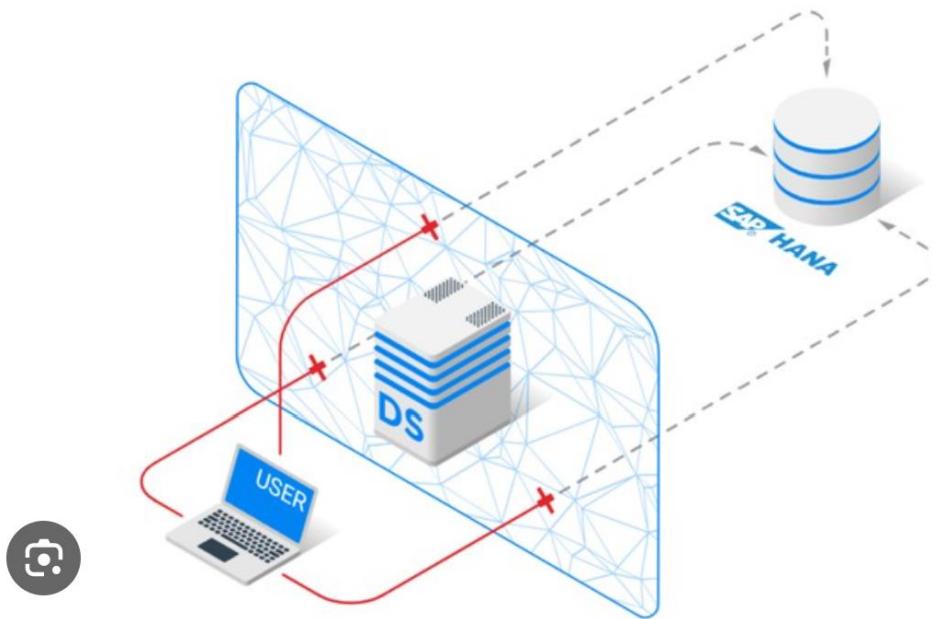
quản lý và bảo trì hạ tầng vật lý, cho phép đội ngũ IT tập trung vào đổi mới thay vì các công việc vận hành tồn thời gian.

- **Tóm lại**, GCP tập trung vào việc trao quyền cho người dùng khả năng xây dựng và mở rộng các giải pháp kỹ thuật số một cách nhanh chóng, bảo mật, và hiệu quả chi phí, tận dụng tối đa sức mạnh từ công nghệ của Google.

1.3.2. Các dịch vụ chính của GCP liên quan đến đồ án

1. **Google Compute Engine (GCE)** : Là nơi bạn chạy máy chủ ảo (VM) để host ứng dụng web backend và Nginx web server. Đây là "trái tim" tính toán của ứng dụng bạn.
2. **Google Cloud SQL**: Dịch vụ database được quản lý hoàn toàn (ví dụ: MySQL). Giúp bạn lưu trữ dữ liệu cho ứng dụng mà không cần lo lắng về việc cài đặt hay bảo trì database phức tạp.
3. **Virtual Private Cloud (VPC)** : Xây dựng mạng riêng an toàn cho tài nguyên của bạn trên đám mây. Giúp bạn định nghĩa các quy tắc tường lửa (Firewall Rules) để kiểm soát chẽ ai/gì có thể truy cập vào VM và database của bạn.
4. **Identity and Access Management (IAM)**: Quản lý quyền truy cập. Đảm bảo chỉ những người/dịch vụ được phép mới có thể truy cập và thao tác với các tài nguyên GCP của bạn (như VM, database, lưu trữ), giúp tăng cường bảo mật.
5. **Google Cloud Storage (GCS)**: Dịch vụ lưu trữ dữ liệu bền vững và có khả năng mở rộng. Lý tưởng để sao lưu mã nguồn và dữ liệu, hoặc lưu trữ các tệp tĩnh lớn của website.

1.3.3. Google Cloud SQL/Firestore: Dịch vụ database quản lý.



Hình 7. Dịch vụ database trong Google Cloud SQL

- Đây là nhóm các dịch vụ database cốt lõi trên Google Cloud Platform, được thiết kế để đơn giản hóa hoàn toàn việc quản lý cơ sở dữ liệu cho bạn. Thay vì phải tự tay cài đặt, cấu hình, bảo trì, vá lỗi, và sao lưu một database trên máy chủ của mình (một công việc rất tốn thời gian và đòi hỏi chuyên môn cao), GCP sẽ lo tất cả những việc này.

1. Google Cloud SQL:

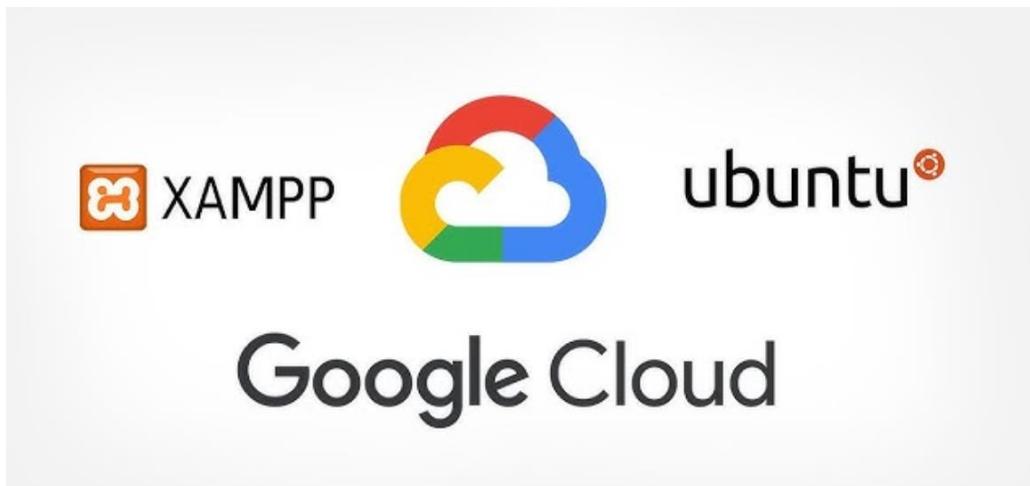
- Là dịch vụ dành cho database quan hệ (Relational Databases). Nó hỗ trợ các hệ quản trị database phổ biến như MySQL, PostgreSQL, và SQL Server.
- Vai trò trong đồ án: Cloud SQL sẽ là nơi lưu trữ tất cả dữ liệu có cấu trúc của ứng dụng web động của bạn (ví dụ: thông tin người dùng, sản phẩm, đơn hàng, v.v.). Bạn chỉ cần tạo một instance, thiết lập người dùng, và ứng dụng của bạn có thể kết nối ngay lập tức. GCP sẽ tự động lo việc sao lưu, phục hồi, và đảm bảo tính khả dụng cao.

2. Firestore:

- Là một dịch vụ database NoSQL (không quan hệ) dạng tài liệu (Document Database). Nó được thiết kế cho các ứng dụng web, di động, và IoT, nơi bạn cần lưu trữ dữ liệu linh hoạt, phi cấu trúc và đồng bộ hóa thời gian thực.
- Vai trò trong đồ án (nếu bạn chọn): Firestore sẽ là nơi lưu trữ dữ liệu của ứng dụng web nếu bạn ưu tiên sự linh hoạt về cấu trúc dữ liệu, khả năng đồng bộ hóa thời gian thực, và mở rộng dễ dàng cho các ứng dụng hiện đại.

1.4. Các công nghệ và công cụ hỗ trợ sử dụng

1.4.1. Ubuntu Server



Hình 8. Ubuntu trong Google Cloud Platform

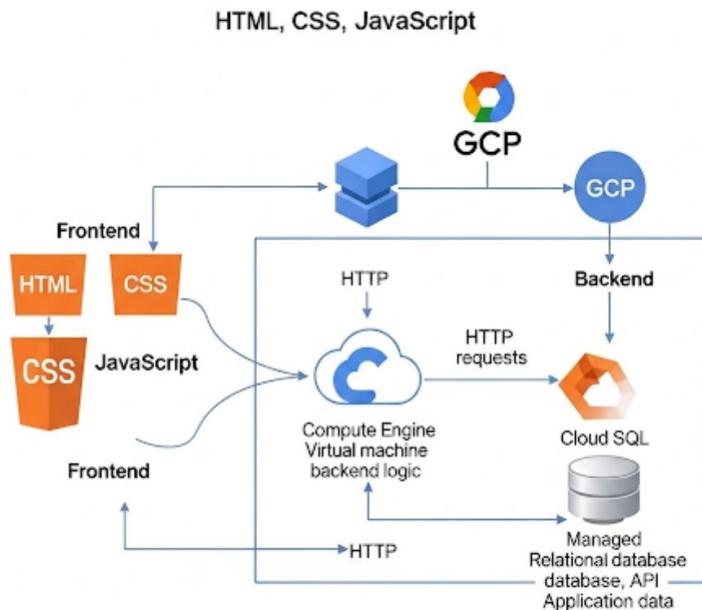
- **Ubuntu Server** là hệ điều hành Linux được lựa chọn để chạy trên các máy ảo (VM instances) của Google Compute Engine (GCE) trong đồ án này. Vai trò của nó là cung cấp một môi trường hoạt động ổn định và mạnh mẽ để triển khai và vận hành ứng dụng web động của chúng tôi.

- **Tương thích tốt với Cloud Platform:** Ubuntu Server được hỗ trợ và tích hợp rất tốt trên Google Cloud Platform, giúp quá trình tạo VM, cài đặt và cấu hình các dịch vụ khác (như Nginx, môi trường ứng dụng) trở nên dễ dàng và nhanh chóng.

1.4.2. Nginx Web Server

- Trong đồ án này, **Nginx** được lựa chọn làm web server và reverse proxy chính trên máy ảo Ubuntu Server của Google Compute Engine (GCE). Nginx nổi tiếng với khả năng xử lý hiệu suất cao, ổn định và khả năng mở rộng tốt, đặc biệt phù hợp cho các ứng dụng web hiện đại.
- Nginx được cấu hình để phục vụ trực tiếp các tệp tĩnh của ứng dụng web (HTML, CSS, JavaScript, hình ảnh, font chữ, v.v.). Việc này giúp giảm tải cho ứng dụng backend, vì Nginx rất hiệu quả trong việc phân phối các tệp này một cách nhanh chóng
- Trong kiến trúc của đồ án, khi người dùng yêu cầu một trang web tĩnh hoặc tài nguyên tĩnh, Nginx sẽ trực tiếp trả về các tệp đó từ thư mục lưu trữ trên VM.

1.4.3. HTML/CSS/JavaScript



Hình 9. HTML, CSS, JavaScript trong GCP

- HTML, CSS và JavaScript là ba công nghệ nền tảng và cốt lõi của bất kỳ ứng dụng web nào, đặc biệt là phần giao diện người dùng (Frontend). Mặc dù chúng không chạy trực tiếp trên các dịch vụ của GCP như máy ảo hay database, mà được tải xuống và thực thi trên trình duyệt web của người dùng, chúng vẫn đóng vai trò cực kỳ quan trọng trong đồ án này.

1. HTML

- Vai trò: HTML định nghĩa cấu trúc và nội dung của mọi trang web. Nó quy định các thành phần như tiêu đề, đoạn văn, hình ảnh, liên kết, bảng biểu, các trường nhập liệu (form) mà người dùng sẽ thấy và tương tác.
- Trong đồ án: Là nền tảng để xây dựng các trang của ứng dụng web. Mã HTML sẽ được phục vụ bởi Nginx (trên GCE) hoặc từ Google Cloud Storage đến trình duyệt người dùng.

2. CSS

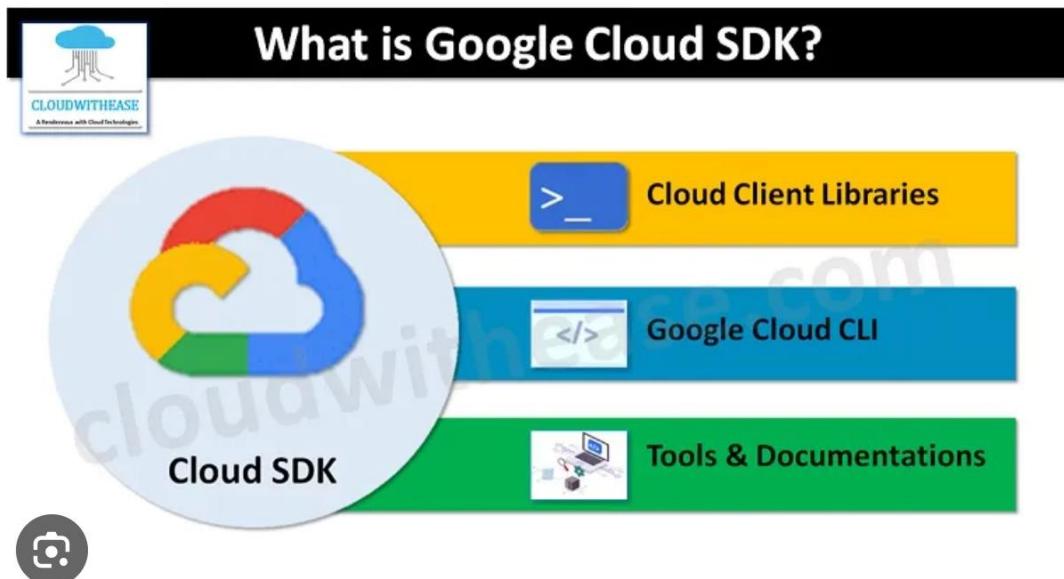
- Vai trò: CSS được sử dụng để tạo kiểu và định dạng cho các thành phần HTML. Nó kiểm soát màu sắc, font chữ, bố cục, khoảng cách, hiệu ứng... biến một trang web đơn giản thành một giao diện hấp dẫn và thân thiện với người dùng.
- Trong đồ án: Đảm bảo ứng dụng web có giao diện đẹp mắt, dễ nhìn và nhất quán trên các thiết bị khác nhau. Các tệp CSS cũng sẽ được phục vụ từ Nginx hoặc Cloud Storage.

3. JavaScript

- Vai trò: JavaScript mang lại tính tương tác và động lực cho trang web. Nó cho phép trang web phản hồi lại hành động của người dùng và quan trọng nhất là giao tiếp với ứng dụng Backend thông qua các API.
- Trong đồ án:
 - Xử lý các sự kiện trên giao diện người dùng.
 - Gửi yêu cầu (requests) đến ứng dụng Backend để lấy dữ liệu, lưu dữ liệu, hoặc thực hiện các chức năng khác (ví dụ: gửi thông tin đăng nhập, hiển thị danh sách sản phẩm từ database).

- Cập nhật nội dung trang web mà không cần tải lại toàn bộ trang (sử dụng AJAX/Fetch API).

1.4.4. SSH Client và Google Cloud SDK



Hình 10. SSH Client và Google Cloud SDK trong GCP

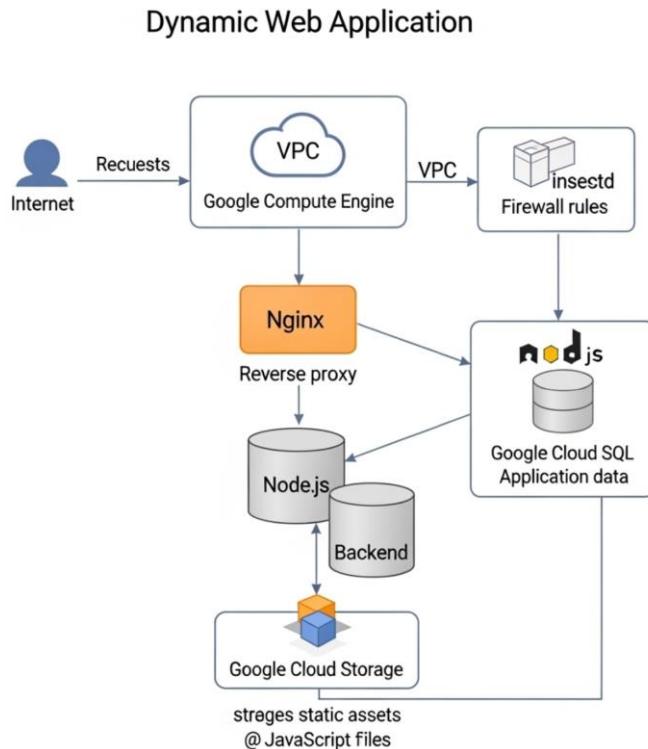
- Trong quá trình triển khai và quản lý ứng dụng web động trên Google Cloud Platform (GCP), SSH Client và Google Cloud SDK là hai công cụ hỗ trợ quan trọng, đóng vai trò cầu nối giữa máy tính cục bộ của bạn và các tài nguyên trên đám mây.
 1. SSH Client (Secure Shell Client)
 - **Định nghĩa:** SSH Client là một chương trình ứng dụng cho phép bạn kết nối bảo mật đến một máy chủ từ xa thông qua giao thức Secure Shell (SSH). Giao thức này mã hóa tất cả dữ liệu được trao đổi giữa client và server, bao gồm cả mật khẩu, đảm bảo an toàn trong quá trình quản trị.
 - **Vai trò :** Truy cập và quản lý máy ảo (VM): Đây là công cụ chính để bạn có thể kết nối từ máy tính cá nhân vào máy ảo Ubuntu Server trên Google Compute Engine (GCE). Sau khi kết nối, bạn có thể thực hiện các tác vụ quản trị như cài đặt phần mềm (Nginx, Node.js), tải mã nguồn ứng dụng, cấu hình hệ thống, kiểm tra log, và khắc phục sự cố.

- **Bảo mật:** SSH sử dụng mã hóa để bảo vệ phiên làm việc của bạn, ngăn chặn việc nghe lén hoặc giả mạo. Việc sử dụng SSH key pairs là phương pháp khuyến nghị để xác thực, an toàn hơn nhiều so với mật khẩu.
2. Google Cloud SDK (Software Development Kit)
- **Định nghĩa:** Google Cloud SDK là một bộ công cụ dòng lệnh (Command-Line Interface - CLI) mạnh mẽ được cung cấp bởi Google. Nó bao gồm các lệnh gcloud, gsutil, bq và các thư viện cần thiết để tương tác với hầu hết các dịch vụ của GCP một cách lập trình hoặc thủ công thông qua terminal.
 - Vai trò trong đồ án:
 - Tạo, xóa, sửa đổi VM trên Compute Engine.
 - Quản lý instance Cloud SQL và database.
 - Cấu hình mạng VPC và Firewall Rules.
 - Quản lý quyền IAM.
 - Tải lên/tải xuống tệp từ Google Cloud Storage.
 - **Tích hợp với SSH Client:** Lệnh gcloud compute ssh trong Cloud SDK cung cấp một cách tiện lợi và an toàn để kết nối SSH đến VM trên GCE mà không cần quản lý khóa SSH thủ công. Nó tự động quản lý các khóa và cấu hình phiên SSH cho bạn.
 - **Tự động hóa tác vụ:** Cloud SDK rất hữu ích cho việc viết các tập lệnh (scripts) để tự động hóa các tác vụ triển khai và quản lý, giúp tiết kiệm thời gian và giảm lỗi.

CHƯƠNG II. CƠ SỞ LÝ THUYẾT VÀ CẤU HÌNH HỆ THỐNG

2.1. Kiến trúc tổng thể hệ thống

2.1.1. Sơ đồ kiến trúc đề xuất



Hình 11. Sơ đồ kiến trúc

2.1.2. Giải thích các thành phần trong kiến trúc

- Sơ đồ trên mô tả tổng quan về kiến trúc của ứng dụng web động khi được triển khai trên Google Cloud Platform (GCP). Kiến trúc này được thiết kế để đảm bảo ứng dụng hoạt động ổn định, bảo mật và có khả năng mở rộng. Các thành phần chính và vai trò của chúng như sau:

- **Người dùng (User):** Là đối tượng cuối cùng tương tác với ứng dụng web thông qua trình duyệt (Web Browser) trên Internet.
- **Internet:** Cầu nối trung gian cho phép người dùng kết nối đến các dịch vụ trên đám mây của GCP.

- **Virtual Private Cloud (VPC):**
 - VPC định nghĩa một môi trường mạng ảo riêng biệt và an toàn trên GCP, cô lập tài nguyên của đồ án khỏi các tài nguyên khác trên internet.
 - Firewall Rules: Là các quy tắc được cấu hình trong VPC để kiểm soát chặt chẽ luồng truy cập
- **Google Compute Engine (GCE) Instance:**
 - Nginx Web Server tiếp nhận, phục vụ các tệp tĩnh (HTML/CSS/JS) và chuyển tiếp các yêu cầu động đến PHP Backend. PHP Backend: Ứng dụng xử lý logic chính.
 - PHP Backend xử lý logic ứng dụng và tương tác với Google Cloud SQL để truy vấn dữ liệu.
- **Google Cloud Storage** có thể được dùng để lưu trữ và phục vụ các tệp tĩnh lớn, giảm tải cho GCE.

2.2. Các khái niệm cơ bản trên Google Cloud Platform

2.2.1. Google Compute Engine (GCE)

- Máy ảo (VM Instances)



Hình 12. Hình ảnh Google Compute Engine

- Google Cloud cung cấp nhiều dòng máy (machine family) khác nhau, mỗi dòng được tối ưu hóa cho một nhu cầu sử dụng cụ thể về CPU, bộ nhớ và hiệu suất mạng. Việc lựa chọn dòng máy phù hợp giúp tối ưu hiệu suất và chi phí.[2]

1. General-purpose :

- **Mục đích:** Cân bằng giữa hiệu suất và chi phí, phù hợp với hầu hết các tải công việc hàng ngày. Đây là lựa chọn linh hoạt cho nhiều ứng dụng.
- **Series tiêu biểu:** E2, N2, N2D, C4 (và N1 cũ hơn).
 - **E2:** Tối ưu hóa chi phí nhất, phù hợp cho các workload không quá nhạy cảm về độ trễ.
 - **N2/N2D:** Cung cấp hiệu suất cân bằng tốt hơn E2, phù hợp cho nhiều ứng dụng phổ biến. N2 dùng CPU Intel, N2D dùng CPU AMD (thường có giá tốt hơn cho cùng hiệu năng).
 - **C4 (mới nhất):** Cung cấp hiệu suất cao và nhất quán hơn cho các workload đa năng đòi hỏi hiệu suất cao, với công nghệ offload nâng cao.

2. Compute-optimized :

- **Mục đích:** Được thiết kế cho các tải công việc yêu cầu hiệu suất CPU cực cao và độ trễ thấp.
- **Series tiêu biểu:** C2, C2D, C3, C3D (và C2A).
 - **C2/C2D:** Tối ưu cho các tác vụ nặng về tính toán, như phân tích gen, mô phỏng khoa học, thiết kế chip điện tử, hoặc các game server yêu cầu CPU mạnh. C2 dùng Intel, C2D dùng AMD.
 - **C3/C3D (mới nhất):** Cung cấp hiệu suất cao hơn nữa với công nghệ tùy chỉnh của Google

3. Memory-optimized :

- **Mục đích:** Dành cho các ứng dụng yêu cầu lượng bộ nhớ (RAM) khổng lồ, thường có tỷ lệ bộ nhớ trên vCPU rất cao.
- **Series tiêu biểu:** M1, M2, M3, M4, X4.

- **M3/M4:** Lý tưởng cho các cơ sở dữ liệu trong bộ nhớ (in-memory databases) như SAP HANA, phân tích dữ liệu lớn, hoặc các workload cần nhiều RAM.

4. Accelerator-optimized :

- **Mục đích:** Được thiết kế đặc biệt để tận dụng tối đa sức mạnh của các bộ tăng tốc phần cứng như GPU (Graphics Processing Units).
- **Series tiêu biểu:** A2, A3, G2.
 - **A2/A3:** Phù hợp cho các tải công việc học máy (Machine Learning), trí tuệ nhân tạo (AI), phân tích dữ liệu chuyên sâu, hoặc render đồ họa nặng, nơi GPU đóng vai trò then chốt.
 - **G2:** Thường được sử dụng cho các ứng dụng ảo hóa đồ họa hoặc phát trực tuyến trò chơi.

So sánh Ưu và Nhược điểm các Dòng Máy Ảo Chính của GCE :

Dòng máy ảo	Mục đích chính	Ưu điểm nổi bật	Nhược điểm
1. Đa năng(E2, N2, N2D, C4)	Cân bằng hiệu suất và chi phí. Phù hợp cho đa số ứng dụng.	Linh hoạt, chi phí tốt thích hợp cho nhiều loại workload.	Không tối ưu cho các tác vụ chuyên biệt cực nặng.
2. Tối ưu Tính toán C2, C2D, C3, C3D)	Yêu cầu hiệu suất CPU cực cao, độ trễ thấp.	CPU mạnh nhất, hiệu năng vượt trội.	Chi phí cao không tối ưu RAM.
3. Tối ưu Bộ nhớ (M1, M2, M3, M4, X4)	Cần lượng bộ nhớ (RAM) rất lớn.	RAM cực khủng lý tưởng cho database in-memory.	Chi phí rất cao CPU không phải mạnh nhất.
4. Tối ưu Bộ tăng tốc (A2, A3, G2)	Sử dụng GPU cho AI/ML, phân tích dữ liệu lớn, render.	Sức mạnh GPU không lò tuyệt vời cho AI/ML.	Chi phí rất cao yêu cầu cấu hình phức tạp.

Bảng 3. So sánh các dòng máy ảo chính của GCE

1. Disk (Persistent Disk)

- Persistent Disk là ổ đĩa lưu trữ khối (block storage) được gắn vào các máy ảo GCE, hoạt động tương tự như một ổ cứng vật lý. Dữ liệu trên Persistent Disk vẫn còn nguyên vẹn ngay cả khi VM bị dừng hoặc xóa. Google Cloud cung cấp nhiều loại Persistent Disk khác nhau, tối ưu cho các nhu cầu hiệu suất và chi phí đa dạng.

Các loại Persistent Disk chính:

- **Standard Persistent Disk (HDD):** Phù hợp cho các tải công việc có yêu cầu I/O thấp đến trung bình, như ổ đĩa khởi động (boot disk) hoặc lưu trữ ít truy cập.
- **SSD Persistent Disk:** Dành cho các tải công việc yêu cầu hiệu suất cao, độ trễ thấp, lý tưởng cho cơ sở dữ liệu hoặc phân tích dữ liệu lớn.
- **Balanced Persistent Disk:** Cung cấp sự cân bằng tốt giữa hiệu suất và chi phí, phù hợp cho nhiều tải công việc đa năng cần hiệu suất tốt hơn Standard. Thường là lựa chọn mặc định.
- **Extreme Persistent Disk:** Cung cấp hiệu suất I/O cực cao, dành riêng cho các ứng dụng cực kỳ nhạy cảm về hiệu suất và độ trễ như database trong bộ nhớ.

2. Image

- Một Image (Ảnh máy ảo) là một khuôn mẫu (template) được cấu hình sẵn, chứa hệ điều hành và có thể bao gồm cả các phần mềm đã cài đặt trước. Image được sử dụng để khởi tạo (tạo) các máy ảo một cách nhanh chóng và nhất quán. Khi bạn tạo một VM từ một Image, nó sẽ tạo ra một Persistent Disk mới dựa trên nội dung của Image đó.
- Các loại Image chính:
 - **Public Images :** Do Google Cloud hoặc các nhà cung cấp bên thứ ba (như các bản phân phối Linux, Windows Server) cung cấp và quản lý. Chúng có sẵn để mọi người sử dụng.

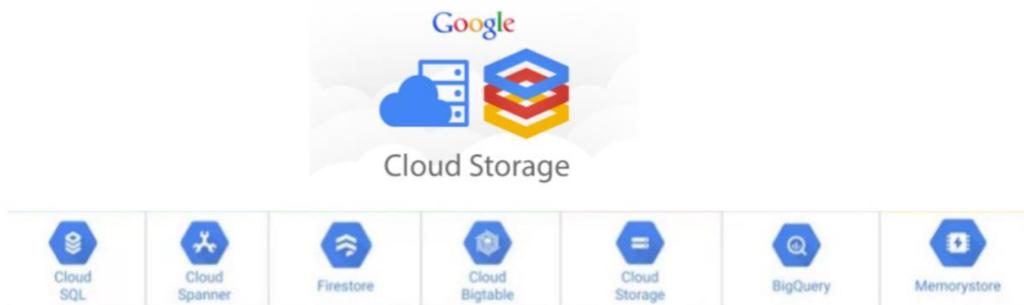
- Custom Images : Do chính bạn tạo ra từ một VM instance hiện có. Custom Image chứa hệ điều hành của VM đó và tất cả các ứng dụng, cấu hình, dữ liệu đã được cài đặt lên VM trước khi tạo Image.

Bảng So sánh Các Loại Disk và Image trong GCE

Tiêu chí so sánh	Disk (Persistent Disk)	Disk (Persistent Disk)
Bản chất	Ổ đĩa lưu trữ dữ liệu cho VM. Dữ liệu trên Disk vẫn còn khi VM tắt/xóa.	Khuôn mẫu để tạo VM. Không chứa dữ liệu ứng dụng trực tiếp, chỉ là "bản sao" của hệ điều hành và phần mềm.
Mục đích	Lưu trữ hệ điều hành (boot disk) . Lưu trữ dữ liệu ứng dụng , cấu hình.	Khởi tạo nhanh chóng và nhất quán các VM. Tái tạo môi trường giống hệt nhau.
Loại chính	Standard (HDD): Phổ thông, chi phí thấp. SSD: Hiệu suất cao, độ trễ thấp.Balanced: Cân bằng giữa hiệu suất và chi phí. Extreme: Hiệu suất cực cao.	Public Images: Do Google/bên thứ ba cung cấp. Custom Images: Do người dùng tự tạo từ VM hiện có.
Mối liên hệ	Khi tạo VM từ một Image, một Persistent Disk mới sẽ được tạo dựa trên nội dung của Image đó.	Persistent Disk là nơi lưu trữ nội dung của Image sau khi VM được tạo.
Giá thành	Tính theo dung lượng (GB/tháng) và hiệu suất/IOPS (tùy loại).	Tính theo dung lượng lưu trữ của Image hoặc miễn phí (Public).
Độ bền dữ liệu	Có. Dữ liệu vẫn còn nguyên.	Không áp dụng. Chỉ là khuôn mẫu, không phải nơi lưu trữ dữ liệu động.

Bảng 4. So sánh các loại Disk và Image trong GCE

2.2.2. Google Cloud SQL/ Firestore



Hình 13. Các dịch vụ Cloud Storage trên GCP

a. Google Cloud SQL (Đối với Database Quan hệ)

- **Google Cloud SQL** là dịch vụ database quan hệ được quản lý hoàn toàn của GCP. Điều này có nghĩa là Google sẽ tự động lo các tác vụ phức tạp như thiết lập, vá lỗi, sao lưu, phục hồi, và mở rộng quy mô cho cơ sở dữ liệu của bạn, cho phép bạn tập trung hoàn toàn vào việc phát triển ứng dụng.

- **Lý do lựa chọn Cloud SQL:**

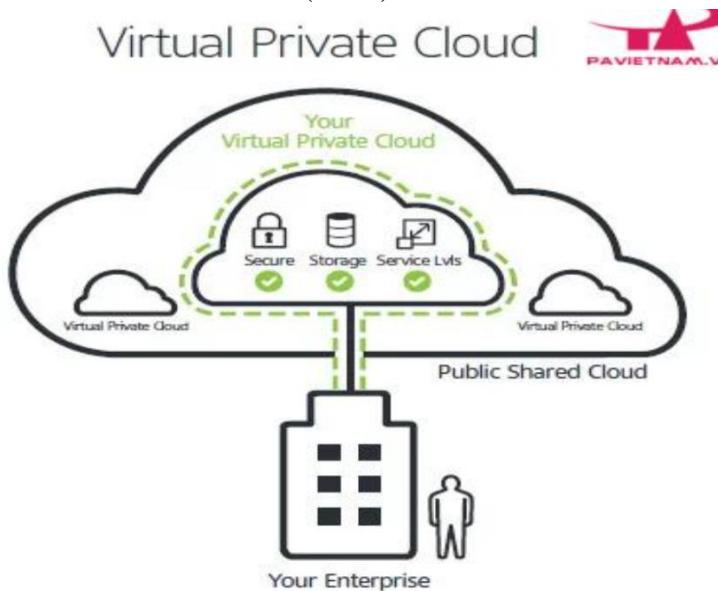
- **Giảm gánh nặng quản lý:** Chúng tôi không cần dành thời gian và nguồn lực để cài đặt, cấu hình hay bảo trì máy chủ database, vốn là một công việc phức tạp và dễ phát sinh lỗi.
- **Tính sẵn sàng cao:** Cloud SQL được thiết kế với tính khả dụng cao, đảm bảo database luôn hoạt động và dữ liệu luôn có sẵn cho ứng dụng.
- **Bảo mật:** GCP tích hợp nhiều lớp bảo mật cho Cloud SQL, bao gồm mã hóa dữ liệu khi lưu trữ và truyền tải, cùng với các tùy chọn mạng riêng ảo.
- **Khả năng mở rộng:** Dễ dàng điều chỉnh dung lượng lưu trữ và hiệu suất (CPU/RAM) của database instance theo nhu cầu phát triển của ứng dụng.

- **Tương thích với PHP:** Cloud SQL hỗ trợ MySQL (hoặc PostgreSQL), là loại database tương thích hoàn hảo với ứng dụng backend PHP của chúng tôi.

b. Firestore (Đối với Database NoSQL - Tùy chọn mở rộng)

- Firestore là một database NoSQL dạng tài liệu (document database) của Google, được xây dựng cho các ứng dụng web, di động và IoT với khả năng đồng bộ hóa dữ liệu thời gian thực và khả năng mở rộng linh hoạt.
- Mục đích/Lý do:
 - Firestore sẽ là lựa chọn tiềm năng nếu chúng tôi cần lưu trữ dữ liệu phi cấu trúc hoặc yêu cầu đồng bộ hóa dữ liệu theo thời gian thực cho các tính năng như chat, thông báo đẩy, hoặc dữ liệu người dùng không có cấu trúc cố định.
 - Nó rất dễ tích hợp với các ứng dụng frontend (ví dụ: qua thư viện Firebase SDK) và tự động mở rộng theo nhu cầu.

2.2.3. Virtual Private Cloud (VPC)



Hình 14. Virtual Private Cloud

- **Virtual Private Cloud (VPC)** trong Google Cloud Platform (GCP) cung cấp một mạng ảo riêng biệt, an toàn và có khả năng mở rộng cho các tài nguyên đám mây của bạn. VPC hoạt động như một "data center ảo" của riêng bạn trên Google Cloud, cho phép bạn định nghĩa và kiểm soát môi trường mạng của mình.

- **Mạng ảo và Subnet**

- **Mạng ảo (Virtual Network)** trong GCP giống như một "mạng riêng ảo" của bạn trên Google Cloud, nơi tất cả tài nguyên (VM, database) sẽ kết nối. Mạng VPC này có phạm vi toàn cầu, cho phép bạn kết nối tài nguyên ở nhiều khu vực khác nhau.
 - Vai trò trong đồ án: Chúng tôi sẽ tạo một mạng VPC riêng để đảm bảo các tài nguyên của mình được cô lập an toàn khỏi Internet và các dự án khác, sử dụng không gian địa chỉ IP riêng.
- **Subnet** là các dải địa chỉ IP nhỏ hơn bên trong mạng VPC của bạn, nằm trong một khu vực cụ thể.
 - Vai trò trong đồ án: Chúng tôi sẽ tạo một subnet trong khu vực đã chọn để nhóm các tài nguyên như VM của GCE và Cloud SQL instance.
 - Firewall Rules : là các quy tắc mạng xác định luồng truy cập vào và ra khỏi tài nguyên trong VPC của bạn, đóng vai trò là lớp bảo mật đầu tiên.

2.2.4. Google Cloud Storage (GCS).

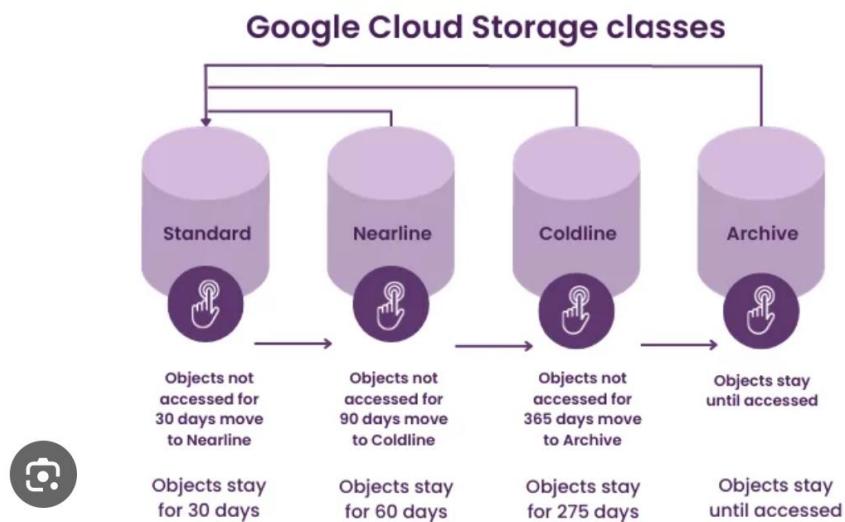
- Vai trò trong đồ án: Để đảm bảo an toàn và chỉ cho phép các kết nối cần thiết, chúng tôi cấu hình các quy tắc cụ thể:
 - Mở cổng HTTP/HTTPS (80/443): Cho phép người dùng truy cập ứng dụng web từ Internet
 - Mở cổng SSH (22): Cho phép quản trị viên kết nối bảo mật vào VM, giới hạn theo dải IP cụ thể để tăng cường bảo mật.
 - Mở cổng Database (3306 cho MySQL): Cho phép PHP backend trên VM kết nối với Cloud SQL, chỉ từ địa chỉ IP nội bộ của VM để database không bị lộ ra ngoài.

- Mặc định chặn: Tất cả các cổng và luồng truy cập không được phép sẽ bị chặn để giảm thiểu rủi ro.

- **Khái niệm Bucket và Object**

- **Bucket** : Là container cơ bản dùng để tổ chức dữ liệu trong GCS. Mọi dữ liệu bạn lưu trữ đều phải nằm trong một bucket. Tên bucket phải là duy nhất trên toàn cầu.
- **Object** : Là dữ liệu thực tế được lưu trữ trong bucket. Một object bao gồm dữ liệu của tệp và các metadata (siêu dữ liệu) mô tả tệp đó. Object là các đơn vị dữ liệu bất biến (không thể sửa đổi sau khi tạo, chỉ có thể ghi đè bằng một object mới).

- **Các lớp lưu trữ (Storage Classes)**



Hình 15. Storage Classes trong GCP

- GCS cung cấp các lớp lưu trữ khác nhau, cho phép bạn tối ưu chi phí dựa trên tần suất truy cập dữ liệu:

- **Standard Storage:**

- Mục đích: Dành cho dữ liệu được truy cập thường xuyên (hot data)
 - Đặc điểm: Chi phí lưu trữ cao hơn, chi phí truy cập thấp.

- **Nearline Storage:**

- Mục đích: Dành cho dữ liệu được truy cập ít hơn 1 lần/tháng (warm data).
 - Đặc điểm: Chi phí lưu trữ thấp hơn Standard, nhưng có chi phí truy xuất dữ liệu và thời gian lưu trữ tối thiểu (thường 30 ngày).

- **Coldline Storage:**

- Mục đích: Dành cho dữ liệu được truy cập ít hơn 1 lần/quý (cooler data).
 - Đặc điểm: Chi phí lưu trữ rất thấp, nhưng chi phí truy xuất cao hơn Nearline và thời gian lưu trữ tối thiểu (thường 90 ngày).

- **Archive Storage:**

- Mục đích: Dành cho dữ liệu được truy cập ít hơn 1 lần/năm (coldest data), chủ yếu để lưu trữ dài hạn và phục hồi thảm họa.
 - Đặc điểm: Chi phí lưu trữ thấp nhất, nhưng chi phí truy xuất và thời gian truy xuất dữ liệu có thể cao/chậm hơn, với thời gian lưu trữ tối thiểu (thường 365 ngày).

2.2.5. Identity and Access Management (IAM)

- Identity and Access Management (IAM) là dịch vụ quản lý quyền truy cập cho các tài nguyên Google Cloud Platform (GCP). Nó cho phép bạn kiểm soát ai (hoặc ứng dụng nào) có thể làm gì trên các tài nguyên của bạn.

- **Vai trò và quyền (Roles and Permissions)**

- **Quyền (Permissions):** Quyền là đơn vị cơ bản nhất, cho phép thực hiện một hành động cụ thể trên một tài nguyên (ví dụ: compute.instances.get cho phép xem thông tin về một máy ảo).
- **Vai trò (Roles):** Vai trò là một tập hợp các quyền. Thay vì cấp quyền trực tiếp cho người dùng hoặc tài khoản dịch vụ, bạn cấp vai trò. Điều này giúp quản lý quyền dễ dàng hơn.
 - **Vai trò cơ bản (Basic Roles):** Các vai trò đơn giản như Owner, Editor, Viewer. Ít được khuyến nghị cho môi trường production vì quá rộng.
 - **Vai trò định nghĩa sẵn (Predefined Roles):** Các vai trò cụ thể cho từng dịch vụ (ví dụ: roles/compute.instanceAdmin cho phép quản lý máy ảo). Nên dùng vì tuân thủ nguyên tắc "quyền tối thiểu".
 - **Vai trò tùy chỉnh (Custom Roles):** Bạn tự tạo ra để đáp ứng nhu cầu cụ thể.
- Tài khoản dịch vụ (Service Accounts)
 - **Định nghĩa:** Tài khoản dịch vụ là một loại tài khoản đặc biệt, được sử dụng bởi các ứng dụng hoặc máy ảo (VM) thay vì người dùng.
 - **Vai trò:** Ứng dụng sử dụng tài khoản dịch vụ để xác thực và gọi các API của Google Cloud một cách an toàn. Ví dụ: một ứng dụng chạy trên VM có thể sử dụng tài khoản dịch vụ để truy cập Cloud SQL mà không cần thông tin đăng nhập của người dùng.

2.3. Cơ sở lý thuyết về Web Server Nginx

2.3.1. Cấu hình cơ bản của Nginx cho trang web tĩnh

- Để phục vụ một trang web tĩnh trên Google Cloud Platform (GCP) bằng Nginx, chúng ta sẽ cấu hình Nginx trên máy ảo Google Compute Engine (GCE). Tệp cấu hình thường được đặt tại /etc/nginx/sites-available/your_site.conf và được kích hoạt bằng cách tạo liên kết tượng trưng (symlink) đến /etc/nginx/sites-enabled/

- Các lệnh thường được sử dụng trong Nginx :

- **sudo systemctl start nginx** : Khởi động dịch vụ Nginx nếu nó chưa chạy.
- **sudo systemctl stop nginx** : Dừng hoàn toàn dịch vụ Nginx.
- **sudo systemctl restart nginx** : Dừng dịch vụ Nginx và sau đó khởi động lại.
- **sudo systemctl reload nginx** : Tải lại cấu hình Nginx mà không làm gián đoạn các kết nối hiện có.

2.3.2. Giới thiệu về ngôn ngữ/framework Backend được sử dụng



Hình 16. Ngôn ngữ Node.js trong Google Cloud Platform

- **JavaScript Everywhere:** Node.js cho phép sử dụng JavaScript cho cả frontend và backend, giúp các nhóm phát triển có thể chia sẻ kiến thức và tài nguyên dễ dàng hơn, giảm thiểu sự phức tạp khi chuyển đổi ngữ cảnh.
- **Hiệu suất cao:** Mô hình I/O không chặn (non-blocking I/O) và kiến trúc điều khiển sự kiện giúp xử lý nhiều yêu cầu cùng lúc hiệu quả.
- **Hệ sinh thái NPM lớn:** Hàng ngàn thư viện sẵn có giúp tăng tốc phát triển.
- **Khả năng mở rộng:** Phù hợp cho kiến trúc microservices và dễ dàng mở rộng theo chiều ngang

Express.js:

- **Tối giản & linh hoạt:** Cho phép bạn tự do lựa chọn thư viện, không ràng buộc.
- **Dễ học:** Cú pháp đơn giản, tài liệu phong phú.
- **Middleware mạnh mẽ:** Xử lý yêu cầu HTTP tuần tự (xác thực, ghi log...).

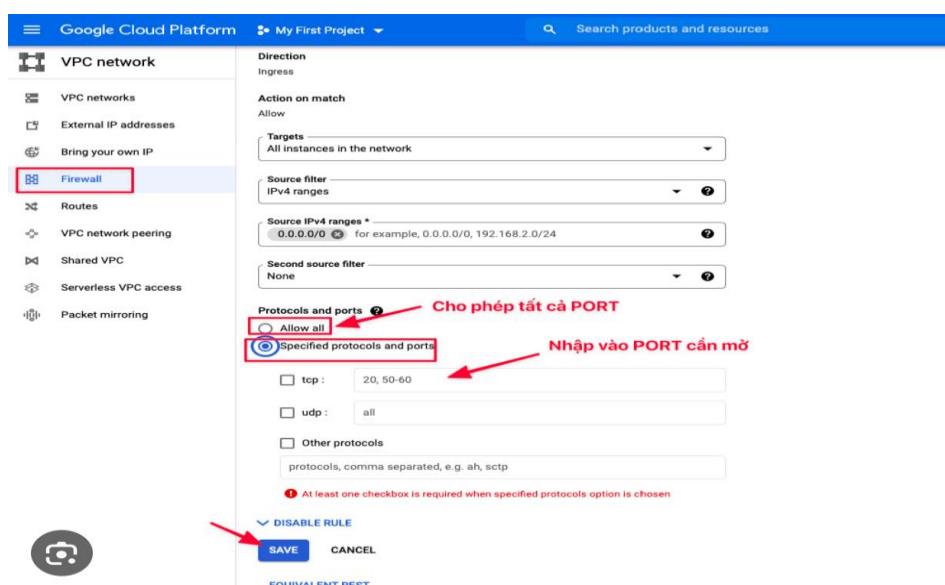
Node.js và Express.js tích hợp tốt với các dịch vụ GCP:

- Cloud Run / App Engine:** Triển khai ứng dụng serverless hoặc được quản lý hoàn toàn, không cần lo về hạ tầng.
- Kubernetes Engine (GKE):** Triển khai ứng dụng container hóa phức tạp.
- Cloud Functions:** Chạy các chức năng nhỏ, theo sự kiện.
- Cloud SQL / Firestore:** Dễ dàng kết nối với các loại cơ sở dữ liệu khác nhau.

Sự kết hợp này mang lại khả năng xây dựng backend hiện đại, mạnh mẽ và dễ dàng mở rộng trên GCP.

2.4. Chuẩn bị môi trường phát triển và triển khai

2.4.1. Các bước tạo tài khoản GCP và dự án mới



Hình 17. Các bước tạo dự án mới trong GCP

- Để bắt đầu làm việc với Google Cloud Platform (GCP), bạn cần thiết lập tài khoản và tạo một dự án mới.
 - Tạo tài khoản Google
 - Sử dụng hoặc tạo một tài khoản Google (Gmail) để truy cập GCP.
 - Đăng ký và kích hoạt tài khoản Google Cloud Platform (GCP)
 - Truy cập <https://console.cloud.google.com/> và đăng nhập.

- Chọn "Try Free" (Dùng thử miễn phí) để kích hoạt tài khoản. Bạn sẽ nhận được \$300 tín dụng miễn phí và quyền truy cập vào các dịch vụ Free Tier sau khi xác minh thông tin thanh toán (không tự động tính phí).

c. Tạo dự án mới trên GCP

1. Trong GCP Console, nhấp vào tên dự án hiện tại ở góc trên bên trái, sau đó chọn "New Project" (Dự án mới).
2. Nhập Project name (Tên dự án) và Project ID (ID duy nhất của dự án).
3. Chọn Billing account đã kích hoạt.
4. Nhấp "Create" (Tạo) và sau đó "SELECT PROJECT" (Chọn dự án) để bắt đầu làm việc.

2.4.2. Cài đặt và cấu hình Google Cloud SDK

- **Google Cloud SDK** là bộ công cụ dòng lệnh (CLI) giúp bạn quản lý và tương tác với các dịch vụ của Google Cloud Platform (GCP) từ máy tính cá nhân.

- Tại sao cần dùng SDK?
 - Quản lý tài nguyên: Tạo, chỉnh sửa, xóa VM, database, mạng... qua dòng lệnh.
 - Kết nối SSH: Dễ dàng kết nối SSH an toàn tới máy ảo GCE.

2.4.3. Chuẩn bị mã nguồn ứng dụng web động

- Trước khi triển khai, việc chuẩn bị và cấu trúc mã nguồn ứng dụng một cách hợp lý là rất quan trọng để đảm bảo quá trình triển khai diễn ra suôn sẻ và ứng dụng hoạt động ổn định.

a. Cấu trúc mã nguồn ứng dụng

- Mã nguồn ứng dụng web động thường được chia thành hai phần chính: Frontend (giao diện người dùng) và Backend (logic xử lý phía máy chủ).
 - Frontend (HTML, CSS, JavaScript):
 - Các tệp này sẽ được **Nginx** phục vụ trực tiếp như nội dung tĩnh.
 - Nên được tổ chức gọn gàng trong một thư mục riêng biệt (ví dụ: public/, frontend/ hoặc static/) trong dự án của bạn.

- Tối ưu hóa các tệp này (minifying CSS/JS, nén hình ảnh) để giảm kích thước và tăng tốc độ tải trang cho người dùng.
- **Backend (Node.js Code):**
 - Chứa toàn bộ logic nghiệp vụ, API, xử lý yêu cầu động và tương tác với database.
 - Nên tuân thủ cấu trúc của framework PHP nếu có (ví dụ: app/, routes/, database/ trong Laravel).
 - Nếu không dùng framework, tổ chức các tệp Node.js thành các module hoặc thư mục hợp lý để dễ quản lý.
- **Thư mục gốc của ứng dụng (Document Root):**
 - Xác định thư mục nào sẽ là "document root" cho Nginx và PHP-FPM. Với nhiều ứng dụng PHP, thư mục này thường là thư mục public/ (hoặc tương tự) chứa index.php và các tệp tĩnh, nhằm mục đích bảo mật (không cho phép truy cập trực tiếp vào các tệp PHP nhạy cảm).

b. Quản lý phụ thuộc (Dependencies)

- Đối với các dự án Node.js, **Composer** là công cụ quản lý phụ thuộc tiêu chuẩn.
- Đảm bảo tệp composer.json của bạn định nghĩa tất cả các thư viện và gói (packages) mà ứng dụng của bạn yêu cầu.
- Khi triển khai lên VM, bạn sẽ chạy lệnh composer install để cài đặt tất cả các phụ thuộc này.

c. Cấu hình kết nối Database

- Ứng dụng Node.js cần các thông tin để kết nối đến Google Cloud SQL (MySQL).
 - **Không nên nhúng trực tiếp thông tin nhạy cảm** (như mật khẩu database) vào mã nguồn.

- **Phương pháp khuyến nghị:**

- Sử dụng biến môi trường (Environment Variables): Các thông tin như DB_HOST, DB_DATABASE, DB_USERNAME, DB_PASSWORD sẽ được đặt trên máy ảo GCE (ví dụ: trong tệp .env hoặc thông qua các biến môi trường của hệ thống) và ứng dụng PHP sẽ đọc chúng khi khởi động.
- Sử dụng tệp cấu hình riêng biệt: Tạo một tệp cấu hình database (ví dụ: config/database.php) nhưng đảm bảo tệp này không chứa thông tin nhạy cảm trực tiếp mà đọc từ các nguồn an toàn hơn (như biến môi trường).

CHƯƠNG III. TRIỂN KHAI DỊCH VỤ WEB TRÊN GCP

3.1. Tạo và cấu hình Virtual Machine trên Compute Engine

3.1.1. Lựa chọn Region, Zone, Machine type và Disk Image

1. Khu vực (Region) và Vùng (Zone):

Machine configuration

Name *	web-doan-server	<small>②</small>
Region *	asia-southeast1 (Singapore)	<small>②</small>
Region is permanent		
Zone *	asia-southeast1-b	<small>②</small>
Zone is permanent		

Hình 18. Thiết lập khu vực (Region) và Vùng (Zone)

- **Lựa chọn trong đồ án:** Chúng tôi lựa chọn Region asia-southeast1 (Singapore) và Zone asia-southeast1-b.
 - **Lý do lựa chọn:** Vị trí này gần với đối tượng người dùng mục tiêu (trong khu vực Đông Nam Á), giúp giảm thiểu độ trễ (latency) khi truy cập ứng dụng, mang lại trải nghiệm người dùng tốt hơn. Việc chọn một Zone cụ thể trong Region cũng là cần thiết khi tạo VM.
2. Loại máy (Machine type):

Series	Description	vCPUs	Memory	CPU Platform
C4	Consistently high performance	2 - 192	4 - 1,488 GB	Intel Emerald
C4A	Arm-based consistently high performance	1 - 72	2 - 576 GB	Google Axion
C4D	Consistently high performance	2 - 384	3 - 3,072 GB	AMD Turin
N4	Flexible & cost-optimized	2 - 80	4 - 640 GB	Intel Emerald
C3	Consistently high performance	4 - 192	8 - 1,536 GB	Intel Sapphire
C3D	Consistently high performance	4 - 360	8 - 2,880 GB	AMD Genoa
E2	Low cost, day-to-day computing	0.25 - 32	1 - 128 GB	Intel Broadwell
N2	Balanced price & performance	2 - 128	2 - 864 GB	Intel Cascade
N2D	Balanced price & performance	2 - 224	2 - 896 GB	AMD Milan
T2A	Scale-out workloads	1 - 48	4 - 192 GB	Ampere Altra
T2D	Scale-out workloads	1 - 60	4 - 240 GB	AMD Milan
N1	Balanced price & performance	0.25 - 96	0.6 - 624 GB	Intel Haswell

Machine type

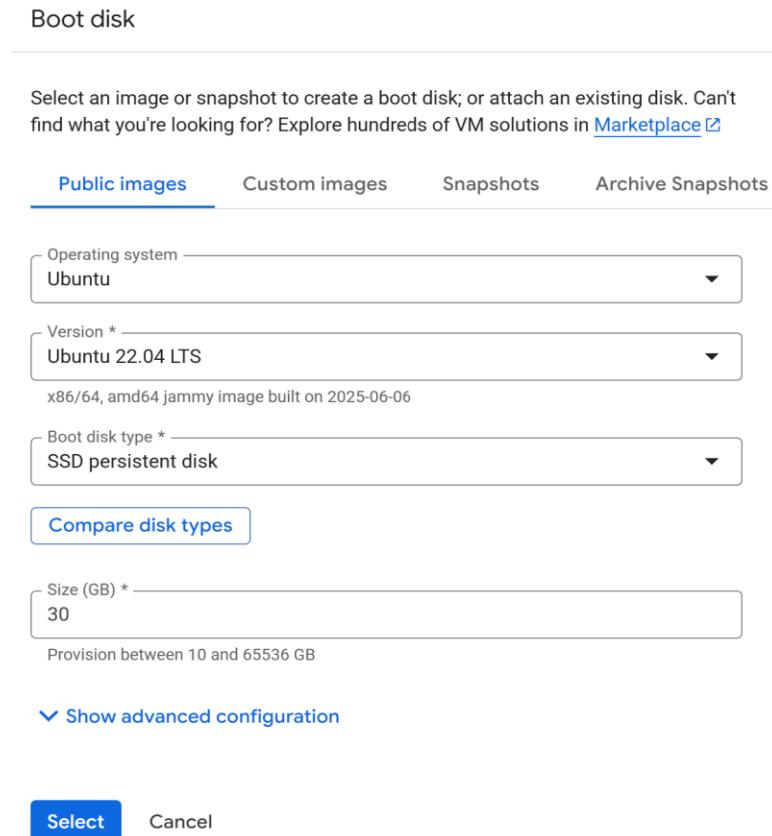
Item	Monthly estimate
2 vCPU + 4 GB memory	\$30.17
10 GB balanced persistent disk	\$1.10
Logging	<u>Cost varies</u>
Monitoring	<u>Cost varies</u>
Snapshot schedule	<u>Cost varies</u>
Total	\$31.27

[Compute Engine pricing](#) [Cloud Operations pricing](#) [Less](#)

Hình 19. Lựa chọn loại máy để thiết lập máy ảo

- Lựa chọn trong đồ án:** Chúng tôi chọn loại máy e2-medium (2 vCPU, 4GB RAM).
- Lý do lựa chọn:** Đây là loại máy thuộc dòng "General-purpose" (Đa năng), cung cấp sự cân bằng tốt giữa hiệu suất và chi phí. Với 2 vCPU và 4GB RAM, nó đủ khả năng xử lý một ứng dụng web động quy mô nhỏ đến trung bình, bao gồm Nginx.

3. Disk Image , Disk Type và Kích thước Size:



Hình 20. Thiết lập boot disk Image Ubuntu

- **Lựa chọn trong đồ án:**
 - Disk Image: Ubuntu 22.04 LTS.
 - Loại ổ đĩa (Boot disk type): SSD Persistent Disk.
 - Kích thước (Size): 30 GB.
- **Lý do lựa chọn:**
 - **Ubuntu 22.04 LTS:** Lựa chọn hệ điều hành Linux phổ biến này nhờ vào sự ổn định, bảo mật cao và cộng đồng hỗ trợ lớn. Phiên bản LTS đảm bảo hỗ trợ dài hạn cho việc cài đặt và quản lý Nginx, PHP một cách dễ dàng và đáng tin cậy.
 - **SSD Persistent Disk:** Được chọn vì mang lại hiệu suất đọc/ghi dữ liệu và độ trễ thấp vượt trội so với HDD. Điều này cực kỳ quan trọng cho ổ đĩa

khởi động và giúp ứng dụng web động hoạt động mượt mà, phản hồi nhanh.

- **30 GB (dung lượng):** Dung lượng này là đủ tối ưu cho hệ điều hành, các phần mềm (Nginx, Node.js), mã nguồn ứng dụng và không gian cho log, giúp kiểm soát chi phí hiệu quả.

3.1.2. Cấu hình Firewall rules cho VM

Item	Monthly estimate
2 vCPU + 4 GB memory	\$24.46
10 GB balanced persistent disk	\$1.00
Logging	Cost varies
Monitoring	Cost varies
Snapshot schedule	Cost varies
Total	\$25.46

Hình 21. Cấu hình thiết lập Firewall Rules trong GCP

1. Đánh dấu chọn (check) ô:
 - "Allow HTTP traffic"
 - "Allow HTTPS traffic"

2. Network tags:

- Khi bạn chọn hai tùy chọn trên, Google Cloud Platform sẽ tự động thêm các **Network tags** tương ứng vào VM của bạn. Thông thường, đó là:
 - http-server : là cổng port 80
 - https-server : là cổng port 443
- Những tag này sẽ được sử dụng bởi các Firewall Rule mặc định của GCP (như default-allow-http và default-allow-https) để cho phép lưu lượng truy cập tương ứng đi vào VM của bạn.

3.1.3. Thiết lập địa chỉ IP tĩnh (External IP)

Network interfaces

Name ↑	Network	Subnetwork	Primary internal IP address	Alias IP ranges	IP stack type	External IP address
nic0	default	default	10.148.0.5		IPv4	34.143.192.68 (Ephemeral)

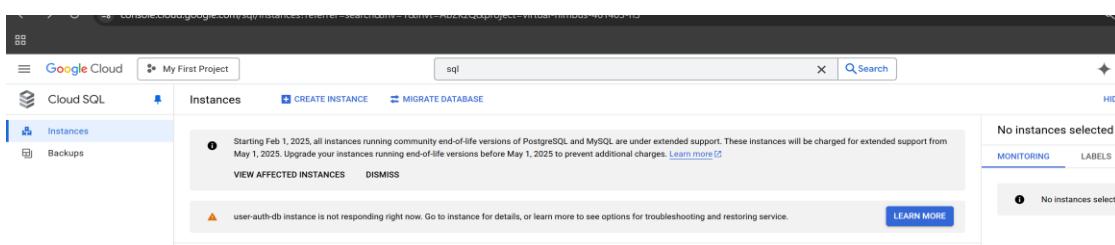
Hình 22. Thiết lập địa chỉ IP tĩnh VM Instances

- **External IP address (địa chỉ IP Public):** 34.143.192.68 và quan trọng hơn là có tính chất (Ephemeral).
- **Truy cập nhất quán:** Đảm bảo người dùng và các dịch vụ khác luôn có thể truy cập ứng dụng của bạn bằng cùng một địa chỉ IP hoặc tên miền mà không bị gián đoạn.
- **Cấu hình Firewall Rules:** Một số Firewall Rules hoặc cấu hình bảo mật khác có thể dựa vào địa chỉ IP cụ thể.
- **Primary internal IP address (Địa chỉ IP nội bộ chính):** 10.148.0.5 (đây là IP chỉ có thể truy cập được từ các tài nguyên khác trong cùng mạng VPC).
- Nội dung này xác nhận rằng máy ảo của bạn hiện đang được gán một địa chỉ IP công cộng là 34.143.192.68, nhưng đây là một địa chỉ IP tạm thời và sẽ thay đổi nếu VM bị dừng và khởi động lại.

3.2. Triển khai và cấu hình Database trên GCP

3.2.1. Tạo và cấu hình Instance Cloud SQL (hoặc Firestore Database)

- **Đăng nhập Google Cloud Console:** Truy cập <https://console.cloud.google.com/> và đảm bảo đã chọn đúng Dự án (Project) làm việc.



Hình 23. Create instance Cloud SQL

- **Tạo Instance Cloud SQL:**

- Nhấn nút "Tạo Instance" (Create Instance) và chọn biểu tượng "MySQL".
- Instance ID: Đặt tên định danh duy nhất cho database (ví dụ: my-do-an-db-instance).
- Mật khẩu người dùng gốc (root user password): Đã thiết lập một mật khẩu mạnh và được ghi lại cẩn thận.
- Phiên bản MySQL: Đã chọn phiên bản MySQL phù hợp với database XAMPP hoặc phiên bản mới nhất tương thích (ví dụ: MySQL 8.0).
- Vùng (Region): Đã chọn vùng địa lý gần nhất với VM Ubuntu để giảm độ trễ kết nối (ví dụ: asia-southeast1 nếu VM cũng ở vùng này).
- Cấu hình máy (Machine configuration): Bắt đầu với cấu hình nhỏ nhất để tối ưu chi phí ban đầu (ví dụ: 1 vCPU, 3.75 GB RAM), có thể nâng cấp sau.
- Loại lưu trữ (Storage type): Đã chọn SSD để đảm bảo hiệu suất truy xuất dữ liệu tốt nhất.
- Dung lượng lưu trữ (Storage capacity): Đã thiết lập dung lượng ban đầu đủ dùng (ví dụ: 20GB-50GB), có thể tăng sau này.

vCPUs	Memory	SSD storage
1	3.75 GB	20 GB

Enterprise edition [UPGRADE](#)
Database version is MySQL 8.0.41
Auto storage increase is enabled
Automated backups are enabled
Point-in-time recovery is enabled
Instance deletion prevention is enabled
Backup retention after deletion is disabled
Located in asia-southeast1-c
Highly available (regional)
No database flags set
No labels set

Hình 24. Cấu hình của SQL Cloud

• Tạo Database và Người dùng cho ứng dụng

Để ứng dụng có thể kết nối và thao tác với database một cách an toàn và có kiểm soát, đã tạo một database và một người dùng riêng biệt trên Cloud SQL instance:

1. Tạo Database:

The screenshot shows the Google Cloud SQL interface. On the left sidebar, under the 'SQL' tab, the 'Databases' section is selected. In the main content area, it shows the 'user-auth-db' MySQL 8.0 instance. A table lists system databases like 'information_schema', 'mysql', 'performance_schema', 'sys', and the user-defined 'user_auth_db'. A 'CREATE DATABASE' button is visible.

Hình 25. Tạo bảng database trong Cloud SQL

- Trong trang quản lý Cloud SQL instance, chọn mục "Databases" ở menu bên trái.
- Nhấn nút "Tạo database" (Create database).
- Nhập tên cho database (ví dụ: do_an_db). Tên này sẽ được sử dụng làm giá trị cho biến môi trường DB_NAME trong ứng dụng Node.js.
- Nhấn "Tạo" để hoàn tất.

2. Tạo Người dùng (User):

The screenshot shows the Google Cloud SQL interface. Under the 'SQL' tab, the 'Users' section is selected. It shows the 'user-auth-db' MySQL 8.0 instance. A table lists users with 'doan_cloud' and 'root' highlighted by a red box. The 'ADDED USERS' tab is active.

Hình 26. Tạo user cho người dùng trong SQL Cloud

- Trong trang quản lý Cloud SQL instance, chọn mục "Users" ở menu bên trái.

- Nhấn nút "Tạo người dùng" (Create user).
- **Tên người dùng:** Đặt tên người dùng mà ứng dụng Node.js của bạn sẽ sử dụng để kết nối. Tên này sẽ là giá trị cho biến môi trường DB_USER.
- **Mật khẩu:** Đặt một mật khẩu mạnh và phức tạp cho người dùng này. Đây sẽ là giá trị cho biến môi trường DB_PASSWORD. Mật khẩu này cũng đã được ghi lại cẩn thận.
- **Host:** Đã để là localhost. Điều này là phù hợp vì ứng dụng Node.js sẽ kết nối đến database thông qua Cloud SQL Auth Proxy, mà proxy này hoạt động như một cổng lắng nghe cục bộ trên chính VM.

3.2.2. Thiết lập kết nối an toàn từ VM đến Database

- Giai đoạn này bao gồm việc cài đặt và cấu hình Cloud SQL Auth Proxy trên máy ảo Ubuntu để tạo một "đường hầm" an toàn, cho phép ứng dụng Node.js kết nối đến Cloud SQL thông qua Private IP, và sau đó cấu hình ứng dụng Node.js để sử dụng kết nối này.
- **Cài đặt Cloud SQL Auth Proxy trên VM Ubuntu**
 - Cloud SQL Auth Proxy là một công cụ giúp ứng dụng của bạn kết nối an toàn đến Cloud SQL mà không cần cấu hình tường lửa phức tạp hoặc IP công cộng. Nó mã hóa lưu lượng truy cập và xác thực bằng cách sử dụng các thông tin xác thực của Google Cloud.

```
/bin/cloud_sql_proxy
phamduclong416@gearshop-frontend-server-2:~/doandientoandammay$ cloud_sql_proxy --version
Cloud SQL Auth proxy: 1.37.7
phamduclong416@gearshop-frontend-server-2:~/doandientoandammay$ sudo nano /etc/systemd/system/cloudsql-proxy.service
phamduclong416@gearshop-frontend-server-2:~/doandientoandammay$ sudo systemctl daemon-reload
phamduclong416@gearshop-frontend-server-2:~/doandientoandammay$ sudo systemctl enable cloudsqiproxy
Created symlink /etc/systemd/system/multi-user.target.wants/cloudsql-proxy.service → /etc/systemd/system/cloudsql-proxy.service.
phamduclong416@gearshop-frontend-server-2:~/doandientoandammay$ sudo systemctl start cloudsqiproxy
phamduclong416@gearshop-frontend-server-2:~/doandientoandammay$ sudo systemctl status cloudsqiproxy
● cloudsqiproxy.service - Google Cloud SQL Proxy
   Loaded: loaded (/etc/systemd/system/cloudsql-proxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2025-07-03 09:07:34 UTC; 6s ago
     Main PID: 13862 (cloud_sql_proxy)
        Tasks: 5 (limit: 4687)
       Memory: 34.3M
          CPU: 842ms
        CGroup: /system.slice/cloudsql-proxy.service
                  └─13862 /usr/local/bin/cloud_sql_proxy -instances=virtual-nimbus-461403-h3:asia-south>
```

Hình 27. Cài đặt và thiết lập kết nối Auth-Proxy.

- **Tải và cài đặt proxy:** Trong cửa sổ SSH của VM, đã chạy các lệnh sau để tải xuống file nhị phân của proxy, cấp quyền thực thi và di chuyển nó vào một thư mục trong biến môi trường PATH để có thể chạy trực tiếp:
 - sudo apt-get update
 - sudo apt-get install -y wget # Đảm bảo wget đã cài đặt để tải file
 - wgethttps://dl.google.com/cloudsql/cloud_sql_proxy.linux.amd64-O

cloud_sql_proxy
- **Xác nhận proxy đã được cài đặt:** Đã chạy lệnh sau để kiểm tra phiên bản của proxy, đảm bảo nó đã được cài đặt thành công và có thể chạy được:
 - cloud SQL Auth proxy: 1.37.7 cho thấy bạn đã tải xuống và cài đặt thành công phiên bản 1.37.7 của Cloud SQL Auth Proxy.
- **Kiểm tra trạng thái dịch vụ:** Lệnh sudo systemctl status cloudsqiproxy và kết quả hiển thị:
 - Xác nhận dịch vụ đã được tải và kích hoạt.
 - Active: active (running) nó cho thấy Cloud SQL Auth Proxy đang hoạt động (running) và đã chạy được một khoảng thời gian. Điều này đảm bảo rằng "đường hầm" kết nối đến Cloud SQL đã được thiết lập thành công.

- Chạy Cloud SQL Auth Proxy bằng Systemd

Tạo một file cấu hình dịch vụ systemd: Đã tạo một file mới trong thư mục dịch vụ của systemd:

- Lệnh : sudo nano /etc/systemd/system/cloudsql-proxy.service



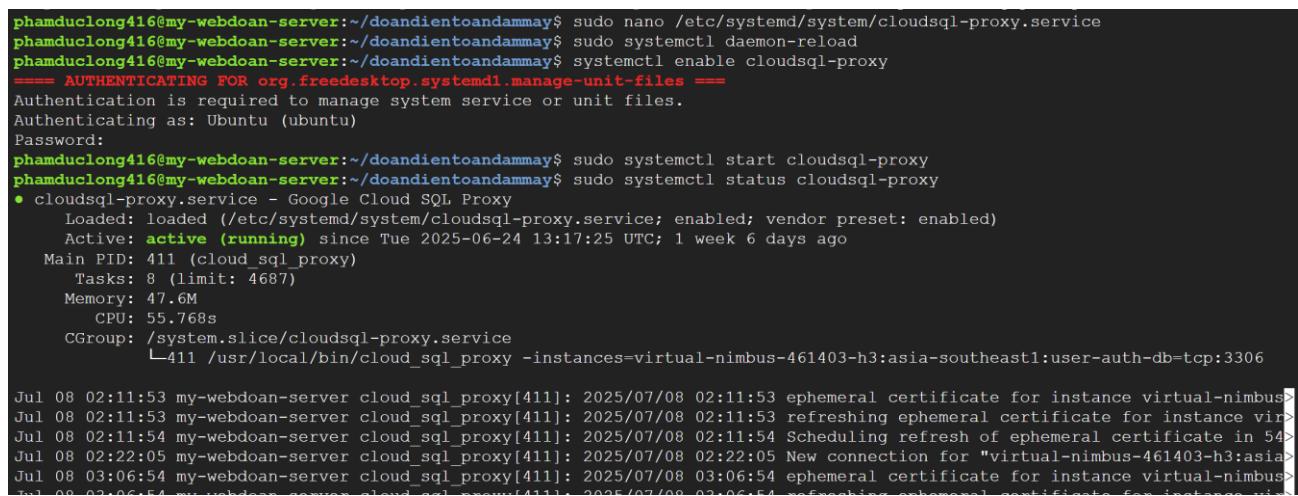
```
GNU nano 6.2
[Unit]
Description=Google Cloud SQL Proxy
After=network.target

[Service]
User=phamduclong416
ExecStart=/usr/local/bin/cloud_sql_proxy -instances=virtual-nimbus-461403-h3:asia-southeast1:my-webdoan-server?authuser=0&hl=en_US&projectNum...
Restart=always

[Install]
WantedBy=multi-user.target
```

Hình 28. Cấu hình dịch vụ Systemd cho Cloud SQL Auth Proxy

- Tải lại cấu hình systemd, kích hoạt và khởi động dịch vụ: Đã chạy các lệnh sau để systemd nhận biết dịch vụ mới, kích hoạt nó để tự khởi động cùng hệ thống, và khởi động nó ngay lập tức:



```
phamduclong416@my-webdoan-server:~/doandientoandammay$ sudo nano /etc/systemd/system/cloudsql-proxy.service
phamduclong416@my-webdoan-server:~/doandientoandammay$ sudo systemctl daemon-reload
phamduclong416@my-webdoan-server:~/doandientoandammay$ sudo systemctl enable cloudsql-proxy
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Authenticating as: Ubuntu (ubuntu)
Password:
phamduclong416@my-webdoan-server:~/doandientoandammay$ sudo systemctl start cloudsql-proxy
phamduclong416@my-webdoan-server:~/doandientoandammay$ sudo systemctl status cloudsql-proxy
● cloudsql-proxy.service - Google Cloud SQL Proxy
   Loaded: loaded (/etc/systemd/system/cloudsql-proxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2025-06-24 13:17:25 UTC; 1 week 6 days ago
     Main PID: 411 (cloud_sql_proxy)
        Tasks: 8 (limit: 4687)
       Memory: 47.6M
          CPU: 55.768s
         CGroup: /system.slice/cloudsql-proxy.service
                   └─411 /usr/local/bin/cloud_sql_proxy -instances=virtual-nimbus-461403-h3:asia-southeast1:my-webdoan-server?authuser=0&hl=en_US&projectNum...
Jul 08 02:11:53 my-webdoan-server cloud_sql_proxy[411]: 2025/07/08 02:11:53 ephemeral certificate for instance virtual-nimbus-461403-h3:asia-sou...
Jul 08 02:11:53 my-webdoan-server cloud_sql_proxy[411]: 2025/07/08 02:11:53 refreshing ephemeral certificate for instance virtual-nimbus-461403-h3:...
Jul 08 02:11:54 my-webdoan-server cloud_sql_proxy[411]: 2025/07/08 02:11:54 Scheduling refresh of ephemeral certificate in 54000ms
Jul 08 02:22:05 my-webdoan-server cloud_sql_proxy[411]: 2025/07/08 02:22:05 New connection for "virtual-nimbus-461403-h3:asia-southeast1:my-webdoan-s...
Jul 08 03:06:54 my-webdoan-server cloud_sql_proxy[411]: 2025/07/08 03:06:54 ephemeral certificate for instance virtual-nimbus-461403-h3:asia-sou...
Jul 08 03:06:54 my-webdoan-server cloud_sql_proxy[411]: 2025/07/08 03:06:54 refreshing ephemeral certificate for instance virtual-nimbus-461403-h3:...
```

Hình 29. Kiểm tra trạng thái dịch vụ Cloud SQL Auth Proxy

- **Mục đích:** Khi proxy này chạy, nó sẽ tạo một cổng lắng nghe ảo trên 127.0.0.1:3306 của chính VM Ubuntu của bạn. Ứng dụng Node.js của bạn sẽ kết nối đến

127.0.0.1:3306 này, và proxy sẽ chuyển tiếp yêu cầu một cách an toàn đến Cloud SQL instance qua kết nối Private IP.

- **Cài đặt và Quản lý Ứng dụng Node.js với PM2**
- Để đảm bảo ứng dụng Node.js của bạn chạy ổn định, tự động khởi động lại khi có lỗi hoặc khi VM reboot, và dễ dàng quản lý log, đã sử dụng PM2.

```
^C
phamduclong416@my-webdoan-server:~/doandientoandammay/public/server$ pm2 flush
[PM2] Flushing /home/phamduclong416/.pm2/pm2.log
[PM2] Flushing:
[PM2] /home/phamduclong416/.pm2/logs/my-backend-app-out.log
[PM2] /home/phamduclong416/.pm2/logs/my-backend-app-error.log
[PM2] Logs flushed
phamduclong416@my-webdoan-server:~/doandientoandammay/public/server$ pm2 restart my-backend-app
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [my-backend-app] (ids: [ 0 ])
[PM2] [my-backend-app] (0) ✓



| id | name           | mode | ø  | status | cpu | memory |
|----|----------------|------|----|--------|-----|--------|
| 0  | my-backend-app | fork | 52 | online | 0%  | 20.5mb |

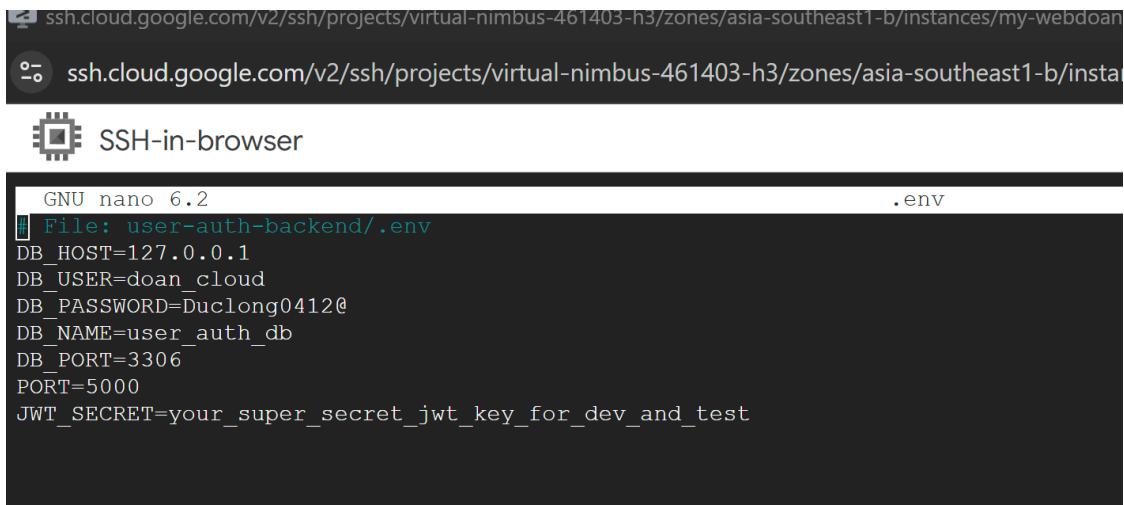


phamduclong416@my-webdoan-server:~/doandientoandammay/public/server$ pm2 logs my-backend-app
[TAILING] Tailing last 15 lines for [my-backend-app] process (change the value with --lines option)
/home/phamduclong416/.pm2/logs/my-backend-app-error.log last 15 lines:
/home/phamduclong416/.pm2/logs/my-backend-app-out.log last 15 lines:
0|my-backe | JWT_SECRET đã được tải thành công.
0|my-backe | Backend API server running on http://localhost:5000
0|my-backe | Connected to MySQL database successfully!
```

Hình 30. Quản lý và kiểm tra hoạt động trạng thái ứng dụng với PM2.

Cấu hình file .env cho ứng dụng Node.js

Để ứng dụng Node.js biết cách kết nối đến database thông qua Cloud SQL Auth Proxy, các biến môi trường đã được cấu hình trong file .env:

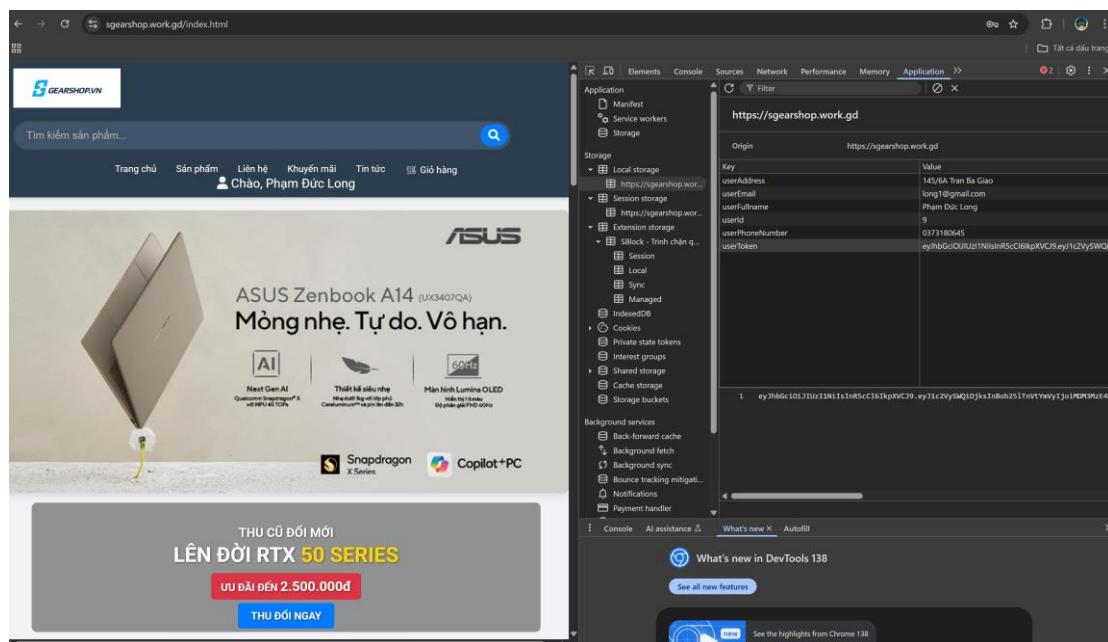


```

GNU nano 6.2
# File: user-auth-backend/.env
DB_HOST=127.0.0.1
DB_USER=doan_cloud
DB_PASSWORD=Duclong0412@123
DB_NAME=user_auth_db
DB_PORT=3306
PORT=5000
JWT_SECRET=your_super_secret_jwt_key_for_dev_and_test
  
```

Hình 31. Cấu hình file .env cho website

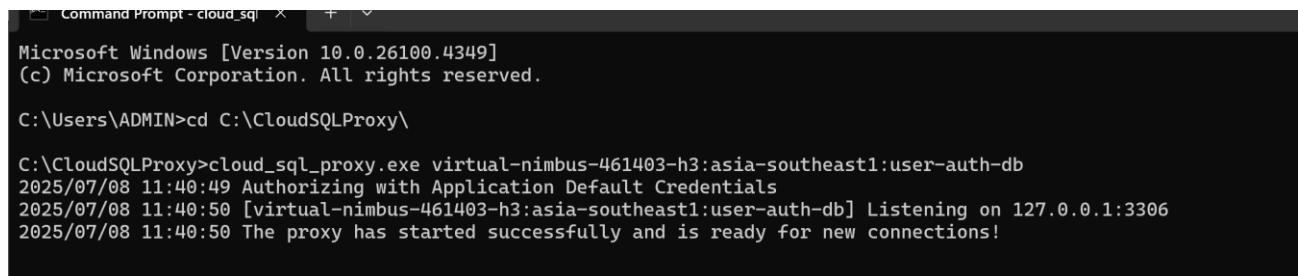
Kiểm tra kết nối Database và Chức năng Web



Hình 32. Giao diện website và dữ liệu người dùng chứng minh kết nối database

- Giao diện người dùng của trang web `sgearshop.work.gd` đã tải thành công, hiển thị thông tin động như "Chào, Phạm Đức Long" và dữ liệu người dùng được lưu trữ trong session storage của trình duyệt. Điều này là bằng chứng gián tiếp nhưng rất mạnh mẽ cho thấy ứng dụng backend đã kết nối thành công với database và truy xuất dữ liệu.

Quản lý Database Cloud SQL từ máy tính cục bộ



```

Microsoft Windows [Version 10.0.26100.4349]
(c) Microsoft Corporation. All rights reserved.

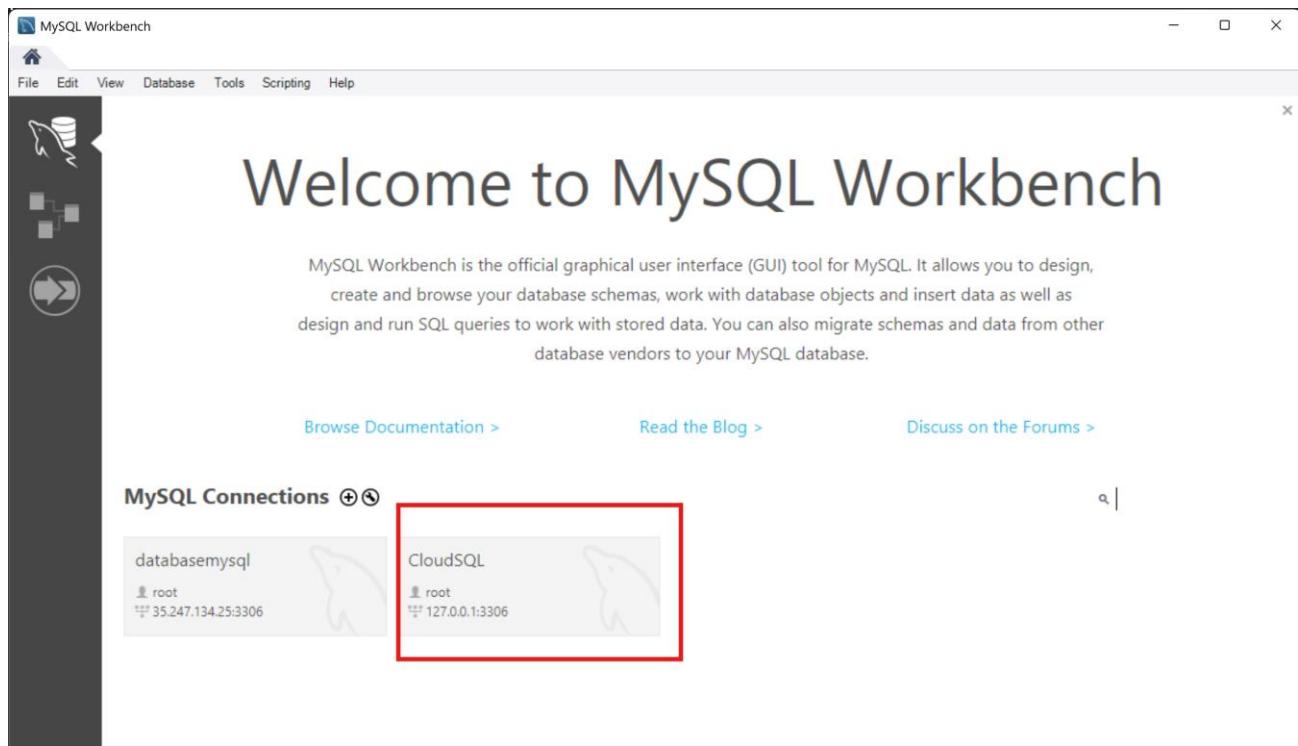
C:\Users\ADMIN>cd C:\CloudSQLProxy\

C:\CloudSQLProxy>cloud_sql_proxy.exe virtual-nimbus-461403-h3:asia-southeast1:user-auth-db
2025/07/08 11:40:49 Authorizing with Application Default Credentials
2025/07/08 11:40:50 [virtual-nimbus-461403-h3:asia-southeast1:user-auth-db] Listening on 127.0.0.1:3306
2025/07/08 11:40:50 The proxy has started successfully and is ready for new connections!

```

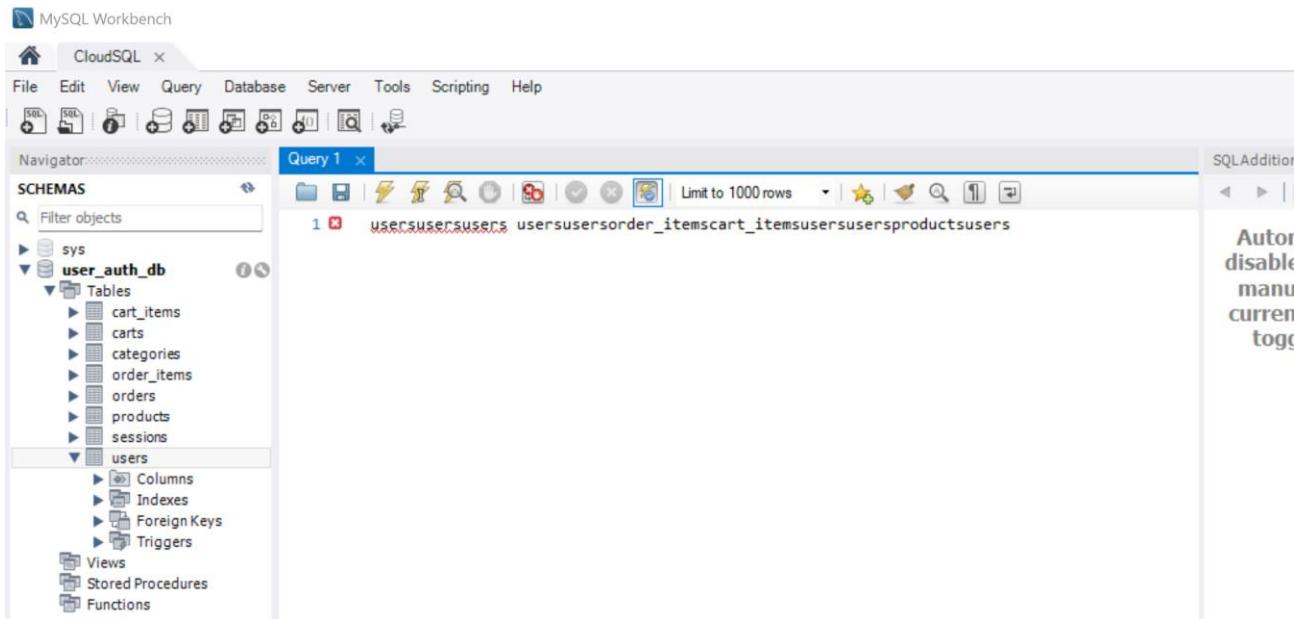
Hình 33. Khởi chạy Cloud SQL Auth Proxy trên máy tính cục bộ

- Sau khi tải về, Cloud SQL Auth Proxy đã được khởi chạy trên môi trường Windows của máy tính cục bộ, thiết lập một kênh kết nối an toàn với Cloud SQL.
 - Thực thi Proxy: Lệnh `cloud_sql_proxy.exe` đã được thực thi với "Tên kết nối Instance" chính xác của database Cloud SQL.
 - Xác thực: Proxy đã xác thực thành công bằng "Application Default Credentials" của Google Cloud.
 - Lắng nghe cục bộ: Proxy đã tạo một cổng lắng nghe ảo trên 127.0.0.1:3306 của máy tính cục bộ, cho phép các ứng dụng MySQL client kết nối qua địa chỉ này.
 - Sẵn sàng kết nối: Proxy đã khởi động thành công và sẵn sàng chuyển tiếp các yêu cầu kết nối đến database Cloud SQL.



Hình 34. Kết nối database Cloud SQL với MySQL máy cục bộ

- Mở công cụ MySQL Workbench lên
- Tạo một kết nối database MySQL mới với các thông tin sau:
 - Hostname/Host/Server: 127.0.0.1 (hoặc localhost)
 - Port: 3306
 - Username: Tên người dùng database đã tạo trong Cloud SQL.
 - Password: Mật khẩu của người dùng.
 - Database : Tên database cụ thể bạn muốn kết nối vào



Hình 35. Giao diện quản lý Database user_auth_db trong MySQL Workbench

- **Mô tả:** Hình ảnh này hiển thị giao diện chính của MySQL Workbench sau khi đã kết nối thành công đến database Cloud SQL.

3.3. Cài đặt và cấu hình Nginx trên VM Ubuntu

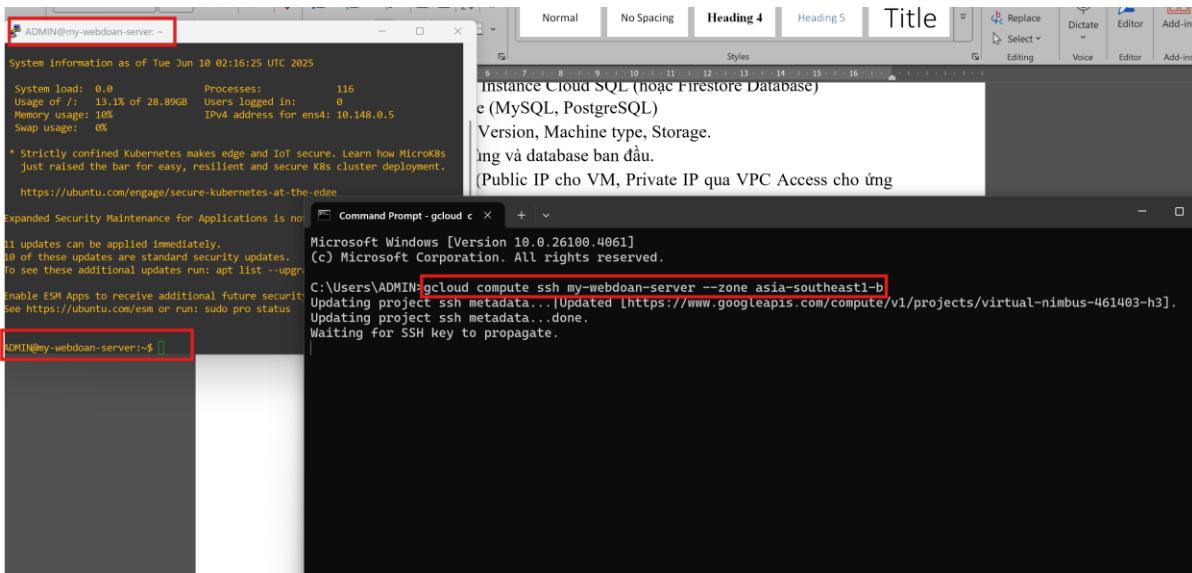
3.3.1. Kết nối SSH đến VM và cập nhật hệ thống

- Để thực hiện các tác vụ cài đặt và cấu hình ứng dụng trên máy ảo (VM) Ubuntu, chúng ta cần kết nối từ xa đến VM bằng giao thức SSH (Secure Shell). Sau khi kết nối, việc cập nhật hệ thống là cần thiết để đảm bảo các gói phần mềm và thư viện đều ở phiên bản mới nhất, tối ưu về bảo mật và tính năng.

a. Phương thức kết nối SSH đến VM

Có hai phương pháp chính để kết nối SSH đến máy ảo Google Compute Engine (GCE):

1. Sử dụng Google Cloud SDK:

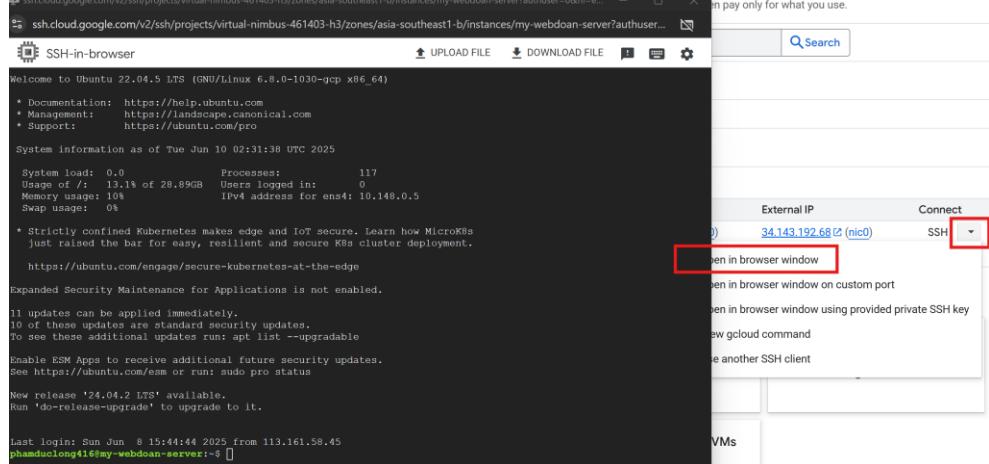


Hình 36. Sử dụng Google Cloud SDK kết nối đến SSH của VM Instance

Các bước thực hiện :

1. Mở **Command Prompt** hoặc **Terminal** trên máy tính cục bộ của bạn.
2. Đảm bảo bạn đã cấu hình Google Cloud SDK với dự án của mình (gcloud init).
3. Sử dụng lệnh sau để kết nối: gcloud compute ssh my-webdoan-server --zone asia-southeast1-b

2. Sử dụng SSH trong Google Cloud Console:



Hình 37. Sử dụng SSH trong Google Cloud Console

Đây là cách nhanh chóng và không yêu cầu cài đặt thêm công cụ trên máy cục bộ.

Các bước thực hiện:

1. Đăng nhập vào **Google Cloud Console**.
2. Điều hướng đến "**Compute Engine**" -> "**VM instances**".
3. Trong danh sách các VM, tìm máy ảo của bạn.
4. Trong cột "**Connect**", nhấp vào nút "**SSH**". Một cửa sổ terminal mới trong trình duyệt sẽ bật lên và tự động kết nối đến VM của bạn.

b. Cập nhật hệ thống trên VM

```
base login: Sun Jun 3 13:11:44 2023 from 115.101.50.43
phamduclong41@my-webdoan-server:~$ sudo apt update
Hit:1 http://asia-southeast1.gce.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://asia-southeast1.gce.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://asia-southeast1.gce.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 https://packages.cloud.google.com/apt google-cloud-ops-agent-jammy-2 InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
11 packages can be upgraded. Run 'apt list --upgradable' to see them.
phamduclong41@my-webdoan-server:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  apport cloud-init libnss-systemd libpam-systemd libsystemd0 libudev1 python3-apport python3-problem-report
  systemd systemd-sysv udev
11 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Hình 38. Cập nhập và nâng cấp các gói trên VM Instance

- Sau khi đã kết nối SSH thành công vào VM, bạn cần cập nhật các gói phần mềm và kho lưu trữ của hệ điều hành.

1. Cập nhật danh sách gói:

- **Lệnh:** sudo apt update
- **Mục đích:** Tải về thông tin mới nhất về các gói có sẵn từ kho lưu trữ của Ubuntu. Điều này không nâng cấp các gói mà chỉ cập nhật danh sách các gói có thể nâng cấp.

2. Nâng cấp các gói đã cài đặt:

- **Lệnh:** sudo apt upgrade -y

- **Mục đích:** Nâng cấp tất cả các gói phần mềm đã cài đặt trên hệ thống lên phiên bản mới nhất có sẵn. Tùy chọn -y tự động đồng ý với các câu hỏi xác nhận.

3.3.2. Cài đặt Nginx Web Server

```
phamduong416@my-webdoan-server:~$ sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.6).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
phamduong416@my-webdoan-server:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-06-08 15:45:23 UTC; 1 day 1h ago
     Docs: man:nginx(8)
     Main PID: 153240 (nginx)
        Tasks: 3 (limit: 4687)
       Memory: 3.7M
          CPU: 294ms
        CGroup: /system.slice/nginx.service
            ├─153240 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
            ├─153241 "nginx: worker process"
            ├─153242 "nginx: worker process"
Jun 08 15:45:23 my-webdoan-server systemd[1]: Starting A high performance web server and a reverse proxy server...
Jun 08 15:45:23 my-webdoan-server systemd[1]: Started A high performance web server and a reverse proxy server.
```

Hình 39. Cài đặt Nginx trên SSH

Các bước cài đặt Nginx trên VM Ubuntu

1. Cài đặt Nginx:

- Lệnh: sudo apt install nginx -y
- Mục đích: Lệnh này sử dụng trình quản lý gói apt của Ubuntu để tải xuống và cài đặt gói Nginx cùng với các phụ thuộc cần thiết. Tùy chọn -y sẽ tự động xác nhận quá trình cài đặt mà không cần hỏi lại.

2. Kiểm tra trạng thái dịch vụ Nginx:

- Lệnh: sudo systemctl status nginx
- Mục đích: Sau khi cài đặt, Nginx thường sẽ tự động khởi động. Lệnh này cho phép bạn kiểm tra xem dịch vụ Nginx có đang chạy (trạng thái active (running)) hay không.

3.4. Tải mã nguồn web tĩnh lên VM

3.4.1. Phương pháp tải mã nguồn (scp, gcloud compute scp)

```
C:\laptrinhmangnangcao>gcloud compute scp --recurse doandientoandammay/ phamduclong416@my-webdoan-server:/home/phamduclong416/ --zone asia-southeast1-b
.editorconfig | 0 kB | 0.2 kB/s | ETA: 00:00:00 | 100%
.gitattributes | 3 kB | 3.2 kB/s | ETA: 00:00:00 | 100%
.gitignore | 0 kB | 0.2 kB/s | ETA: 00:00:00 | 100%
.gitignore | 0 kB | 0.2 kB/s | ETA: 00:00:00 | 100%
misc.xml | 0 kB | 0.2 kB/s | ETA: 00:00:00 | 100%
modules.xml | 0 kB | 0.3 kB/s | ETA: 00:00:00 | 100%
workspace.xml | 7 kB | 7.6 kB/s | ETA: 00:00:00 | 100%
anhnen.jpg | 327 kB | 327.3 kB/s | ETA: 00:00:00 | 100%
banerquangcao1.webp | 46 kB | 46.9 kB/s | ETA: 00:00:00 | 100%
banphimco1.webp | 70 kB | 70.3 kB/s | ETA: 00:00:00 | 100%
banphimco2.webp | 160 kB | 160.8 kB/s | ETA: 00:00:00 | 100%
banphimco3.webp | 65 kB | 65.9 kB/s | ETA: 00:00:00 | 100%
banphimco4.webp | 112 kB | 112.6 kB/s | ETA: 00:00:00 | 100%
banphimco5.webp | 135 kB | 135.2 kB/s | ETA: 00:00:00 | 100%
banphimco6.webp | 73 kB | 73.7 kB/s | ETA: 00:00:00 | 100%
doandientoandammay.iml | 0 kB | 0.3 kB/s | ETA: 00:00:00 | 100%
favicon.ico | 0 kB | 0.7 kB/s | ETA: 00:00:00 | 100%
hotrokhachhang.jpg | 30 kB | 30.8 kB/s | ETA: 00:00:00 | 100%
```

Hình 40. Tải mã nguồn web tĩnh bằng gcloud lên VM

- Để đưa mã nguồn web tĩnh lên VM, chúng ta sử dụng các công cụ sao chép an toàn qua SSH.

- **Sử dụng gcloud compute scp :**
- Đây là cách tiện lợi nhất cho người dùng GCP vì tự động quản lý SSH và xác thực.
- Lệnh đã dùng trong đồ án :

 - gcloudcomputescp--recursedoandientoandammay/phamduclong416@my-webdoan-server:/home/phamduclong416/ --zone asia-southeast1-b

- Lệnh này sao chép thư mục cục bộ doandientoandammay/ cùng các tệp bên trong lên thư mục /home/phamduclong416/ trên VM.
- --recurse: Đảm bảo rằng lệnh sẽ sao chép toàn bộ thư mục doandientoandammay/ và tất cả các tệp, thư mục con bên trong nó.
- doandientoandammay/: Đây là thư mục chứa mã nguồn web tĩnh của bạn trên máy tính cục bộ.
- phamduclong416@my-webdoan-server: Tên trên VM (phamduclong416) và tên máy ảo đích (my-webdoan-server).
- --zone asia-southeast1-b: Chỉ định Zone của máy ảo để lệnh có thể tìm thấy nó.

3.4.2. Đặt mã nguồn vào thư mục được cấu hình bởi Nginx

1. Tải mã nguồn web tĩnh lên VM:

```

drwxr-xr-x 3 phamduclong416 www-data 4096 Jun  6 19:22 server
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ cd html
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ ls -l
total 300
drwxr-xr-x 1 phamduclong416 www-data 13637 Jun  6 19:18 cart.html
-rw-r--r-- 1 phamduclong416 www-data 7040 Jun  6 19:18 complete.html
-rw-r--r-- 1 phamduclong416 www-data 4425 Jun  6 19:18 dashboard.html
-rw-r--r-- 1 phamduclong416 www-data 11291 Jun  6 19:18 index.html
-rw-r--r-- 1 phamduclong416 www-data 3970 Jun  6 19:18 login.html
-rw-r--r-- 1 phamduclong416 www-data 10217 Jun  6 19:18 order-info.html
-rw-r--r-- 1 phamduclong416 www-data 7310 Jun  6 19:18 payment.html
-rw-r--r-- 1 phamduclong416 www-data 15054 Jun  6 19:18 product-detail.html
-rw-r--r-- 1 phamduclong416 www-data 2643 Jun  6 19:18 products.html
-rw-r--r-- 1 phamduclong416 www-data 4799 Jun  6 19:18 register.html
-rw-r--r-- 1 phamduclong416@my-webdoan-server:~/doandientoandamay/public$ cd ..
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ cd css
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ cd ..
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ ls -l
-rw-r--r-- 1 phamduclong416 www-data 2466 Jun  6 19:18 login.css
-rw-r--r-- 1 phamduclong416 www-data 3529 Jun  6 19:18 products-detail.css
-rw-r--r-- 1 phamduclong416 www-data 2183 Jun  6 19:18 register.css
-rw-r--r-- 1 phamduclong416 www-data 5353 Jun  6 19:18 style.css
N phamduclong416@my-webdoan-server:~/doandientoandamay/public$ cd ..
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ cd js
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ ls -l
total 76
-rw-r--r-- 1 phamduclong416 www-data 9195 Jun  6 19:18 cart.js
-rw-r--r-- 1 phamduclong416 www-data 3183 Jun  6 19:18 complete.js
-rw-r--r-- 1 phamduclong416 www-data 3508 Jun  6 19:18 index.js
-rw-r--r-- 1 phamduclong416 www-data 8506 Jun  6 19:18 login.js
-rw-r--r-- 1 phamduclong416 www-data 10960 Jun  6 19:18 order-info.js
-rw-r--r-- 1 phamduclong416 www-data 9100 Jun  6 19:18 payment.js
-rw-r--r-- 1 phamduclong416 www-data 4067 Jun  6 19:18 products.js
-rw-r--r-- 1 phamduclong416 www-data 4404 Jun  6 19:18 register.js
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ cd ..
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ cd server
phamduclong416@my-webdoan-server:~/doandientoandamay/public$ ls -l
total 0
-rw-r--r-- 1 phamduclong416 www-data 3540 Jun  6 19:18 dashboard.js
drwxr-xr-x 105 phamduclong416 www-data 4096 Jun  6 19:22 node_modules
-rw-r--r-- 1 phamduclong416 www-data 43575 Jun  6 19:24 package-lock.json
-rw-r--r-- 1 phamduclong416 www-data 461 Jun  6 19:22 package.json
-rw-r--r-- 1 phamduclong416 www-data 33646 Jun  6 19:22 server.js

```

Hình 41. Mã nguồn web tĩnh trên Nginx

- Các tệp HTML, CSS, JavaScript và các tài nguyên khác của bạn hiện đang nằm ở đường dẫn /home/phamduclong416/doandientoandamay/public/, và đây là đường dẫn mà Nginx đã được cấu hình làm thư mục gốc để phục vụ.

2. Chính sửa tệp cấu hình Nginx mặc định:

```

root /home/phamduclong416/doandientoandammay/public/html;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

# THEM CAC KHOI LOCATION NAY
location /css/ {
    alias /home/phamduclong416/doandientoandammay/public/css/;
    try_files $uri =404;
}

location /js/ {
    alias /home/phamduclong416/doandientoandammay/public/js/;
    try_files $uri =404;
}

location /images/ { # Hoặc /img/ tuy theo tên thư mục ảnh của bạn
    alias /home/phamduclong416/doandientoandammay/public/images/;
    try_files $uri =404;
}

# pass PHP scripts to FastCGI server
#
location ~ \.php$ {
    include snippets/fastcgi-php.conf; # Bao gồm các thiết lập FastCGI cơ bản
    fastcgi_pass unix:/run/php/php8.1-fpm.sock; # Bỏ chú thích và thay đổi thành PHP 8.1 FPM socket

    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name; # Bắt buộc phải có để PHP biết đường dẫn script
    include fastcgi_params; # Bao gồm các tham số FastCGI khác
}

```

Hình 42. Sửa tệp cấu hình Nginx mặc định

- Chúng ta sẽ chỉnh sửa tệp cấu hình mặc định của Nginx để phù hợp với vị trí mã nguồn web tĩnh của bạn và cấu trúc thư mục con (css, js, images).
- Các khối location này trong cấu hình Nginx được sử dụng để tối ưu hóa và tổ chức cách Nginx phục vụ các loại tệp tĩnh khác nhau (CSS, JavaScript, và hình ảnh) trong trang web của bạn.
 - **location /css/ :** Xử lý các yêu cầu từ trình duyệt đối với các tệp CSS. Nó chỉ định Nginx tìm các tệp CSS trong thư mục css cụ thể trong mã nguồn của bạn và trả về lỗi 404 nếu không tìm thấy tệp.
 - **location /js/ :** Tương tự, khái này xử lý các yêu cầu cho các tệp JavaScript. Nginx sẽ tìm kiếm các tệp JS trong thư mục js và báo lỗi nếu không tìm thấy.
 - **location /images/:** Quản lý các yêu cầu cho các tệp hình ảnh. Nginx sẽ tìm kiếm hình ảnh trong thư mục images và trả về lỗi nếu tệp không tồn tại.

3. Đặt quyền sở hữu và phân quyền truy cập cho mã nguồn:

```
my: apt install -y geoip
ADMIN@my-webdoan-server:~$ ls -ld /home/phamduclong416/doandientoandammay/
drwxr-xr-x 7 phamduclong416 www-data 4096 Jun  6 19:23 /home/phamduclong416/doandientoandammay/
ADMIN@my-webdoan-server:~$ ls -ld /home/phamduclong416/doandientoandammay/public/
drwxr-xr-x 7 phamduclong416 www-data 4096 Jun  6 19:18 /home/phamduclong416/doandientoandammay/public/
ADMIN@my-webdoan-server:~$ ls -ld /home/phamduclong416/doandientoandammay/public/html/
drwxr-xr-x 2 phamduclong416 www-data 4096 Jun  8 15:44 /home/phamduclong416/doandientoandammay/public/html/
ADMIN@my-webdoan-server:~$ ls -l /home/phamduclong416/doandientoandammay/public/html/index.html
-rw-r--r-- 1 phamduclong416 www-data 11291 Jun  6 19:18 /home/phamduclong416/doandientoandammay/public/html/index.html
ADMIN@my-webdoan-server:~$ ls -l /home/phamduclong416/doandientoandammay/public/css/style.css
-rw-r--r-- 1 phamduclong416 www-data 53553 Jun  6 19:18 /home/phamduclong416/doandientoandammay/public/css/style.css
ADMIN@my-webdoan-server:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ADMIN@my-webdoan-server:~$ sudo systemctl restart nginx
ADMIN@my-webdoan-server:~$ 
```

Hình 43. Các đặc quyền sở hữu và quyền truy cập cho Nginx

Đặt quyền sở hữu (Ownership):

- Lệnh: sudo chown -R www-data:www-data /home/phamduclong416/doandientoandammay/
- Mục đích: Lệnh này thay đổi người sở hữu (user) và nhóm sở hữu (group) của thư mục chứa mã nguồn của bạn. Điều này cho phép Nginx có quyền truy cập đầy đủ vào các tệp của trang web.

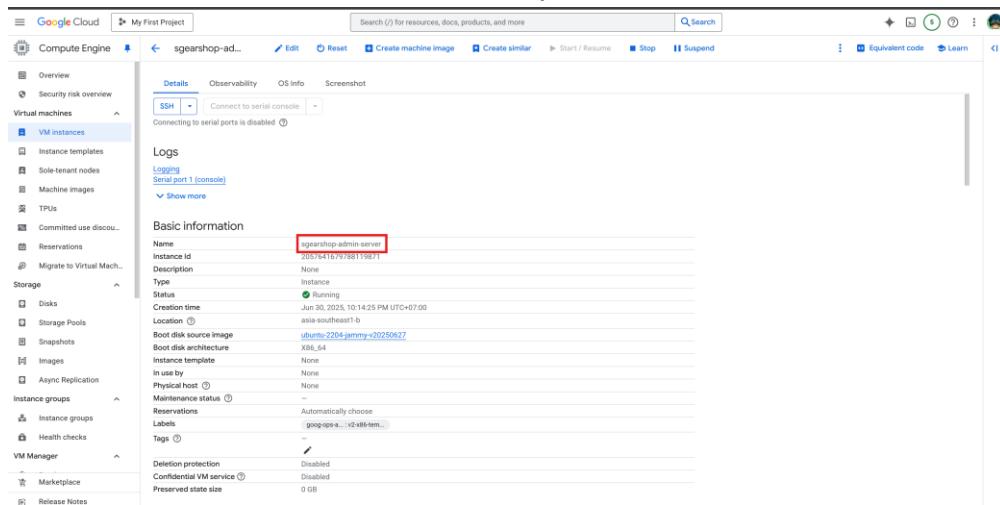
Phân quyền truy cập (Permissions):

- Lệnh : sudo chmod -R 755 /home/phamduclong416/doandientoandammay/
- Mục đích: Lệnh này đặt các quyền truy cập sau cho thư mục và các tệp bên trong:
 - 7 (chủ sở hữu - www-data): Cho phép đọc, ghi và thực thi (read, write, execute).
 - 5 (nhóm - www-data): Cho phép đọc và thực thi (read, execute).
 - 5 (người khác - others): Cho phép đọc và thực thi (read, execute).

3.5. Triển khai Máy chủ Admin riêng biệt và Kết nối Database

- Để tăng cường bảo mật và phân tách trách nhiệm, một máy chủ riêng biệt đã được triển khai để chạy ứng dụng quản trị (Admin Panel) và kết nối đến database Cloud SQL.

3.5.1. Tạo và cấu hình VM cho máy chủ Admin



Hình 44. Tạo máy ảo Ubuntu cho trang Admin

- Một máy ảo Ubuntu mới đã được tạo trên Google Cloud Platform, được đặt tên là sgearshop-admin-server, với cấu hình tài nguyên phù hợp cho việc chạy ứng dụng Admin Panel.

3.5.2. Cài đặt và cấu hình Cloud SQL Auth Proxy trên VM Admin

- Tương tự như máy chủ ứng dụng chính, Cloud SQL Auth Proxy cũng được cài đặt và cấu hình trên sgearshop-admin-server để thiết lập kết nối an toàn và riêng tư đến database Cloud SQL.

- Để đảm bảo proxy luôn chạy ngầm và tự động khởi động lại, một dịch vụ Systemd đã được cấu hình trên sgearshop-admin-server :

```
phamduclong416@sgearshop-admin-server:~/ ~$ ./cloud_sql_proxy -instances=virtual-nimbus-461403-h3:asia-southeast1:us
er-auth-db=tcp:3306
2025/07/01 02:14:09 This is the Cloud SQL Proxy v1. It is no longer receiving active feature development. For the
latest features and improvements, migrate to the v2 version of the Cloud SQL Proxy. For details, see: https://gi
thub.com/GoogleCloudPlatform/cloud-sql-proxy/blob/main/migration-guide.md
2025/07/01 02:14:09 current FDs rlimit set to 1048575, wanted limit is 8500. Nothing to do here.
2025/07/01 02:14:11 Listening on 127.0.0.1:3306 for virtual-nimbus-461403-h3:asia-southeast1:user-auth-db
2025/07/01 02:14:11 Ready for new connections
2025/07/01 02:14:11 Generated RSA key in 147.475738ms
```

Hình 45. Khởi chạy Cloud SQL Auth Proxy trên máy chủ Admin

- Lắng nghe cục bộ: Dòng Listening on 127.0.0.1:3306 for virtual-nimbus-461403-h3:asia-southeast1:user-auth-db xác nhận rằng proxy đã tạo thành công một cổng lắng nghe ảo trên 127.0.0.1:3306 trên máy chủ Admin, sẵn sàng nhận các kết nối.
- Sẵn sàng kết nối: Thông báo Ready for new connections khẳng định proxy đã khởi động thành công và sẵn sàng chuyển tiếp các yêu cầu kết nối đến database Cloud SQL.

```
phamduclong416@sgearshop-admin-server:~/sgearshop-admin-panel$ node adminServer.js
AdminJS loaded: function
AdminJS.version: 7.8.16
typeof AdminJS.registerAdapter: function
Warning: connect.session() MemoryStore is not
designed for a production environment, as it will leak
memory, and will not scale past a single process.
AdminJS: bundling user components...
express-session deprecated undefined resave option; provide resave option node_modules/@adminjs/express/lib/build
AuthenticatedRouter.js:59:41
express-session deprecated undefined saveUninitialized option; provide saveUninitialized option node_modules/@adm
injs/express/lib/buildAuthenticatedRouter.js:59:41
express-session deprecated req.secret; provide secret option node_modules/@adminjs/express/lib/buildAuthenticated
Router.js:59:41
Warning: connect.session() MemoryStore is not
designed for a production environment, as it will leak
memory, and will not scale past a single process.
Admin panel listening on port 5001
Admin panel available at http://localhost:5001/admin
Admin panel connected to Cloud SQL database successfully!
```

Hình 46. Tạo kết nối Auth-Proxy đến adminSerer.js

- Cloud SQL Auth Proxy trên máy chủ Admin tạo một cổng lắng nghe cục bộ (127.0.0.1:3306), cho phép ứng dụng Admin Panel kết nối đến database Cloud SQL một cách an toàn thông qua Private IP, mà không cần phơi bày database ra internet.

3.5.3. Triển khai Ứng dụng Admin Panel (Node.js/AdminJS)

- Ứng dụng Admin Panel, được xây dựng bằng Node.js và AdminJS, đã được triển khai trên sgearshop-admin-server.

```
phamduclong416@sgearshop-admin-server:~/sgearshop-admin-panel$ ls
adminServer.js  components  node_modules  package-lock.json  package.json
phamduclong416@sgearshop-admin-server:~/sgearshop-admin-panel$ nano adminServer.js
phamduclong416@sgearshop-admin-server:~/sgearshop-admin-panel$ cat adminServer.js
// adminServer.js

require('dotenv').config();
const express = require('express');
const session = require('express-session');
const jwt = require('jsonwebtoken'); // Có thể không dùng trực tiếp trong file này nếu chỉ AdminJS xử lý auth
const bcrypt = require('bcryptjs'); // Có thể không dùng trực tiếp trong file này nếu chỉ AdminJS xử lý auth
const { Sequelize, DataTypes } = require('sequelize');

// AdminJS & Adapters
const { AdminJS } = require('adminjs');
const AdminJSExpress = require('@adminjs/express');
const AdminJSSequelize = require('@adminjs/sequelize');
const SequelizeStore = require('connect-session-sequelize')(session.Store);
// Thêm middleware để xử lý dữ liệu form bao gồm cả file uploads
const formidableMiddleware = require('express-formidable');

// Đăng ký adapter Sequelize với AdminJS
AdminJS.registerAdapter({
  Database: AdminJSSequelize.Database,
  Resource: AdminJSSequelize.Resource,
});

// Cấu hình kết nối Sequelize đến Cloud SQL
```

Hình 47. Triển khai AdminServer.js trên máy Ubuntu của Google

Cài đặt và khởi chạy ứng dụng với PM2

```
phamduclong416@sgearshop-admin-server:~/sgearshop-admin-panel$ pm2 start adminServer.js --name sgearshop-admin
[PM2] Starting /home/phamduclong416/sgearshop-admin-panel/adminServer.js in fork_mode (1 instance)
[PM2] Done.



| <b>id</b> | <b>name</b>     | <b>mode</b> | <b>⌚</b> | <b>status</b> | <b>cpu</b> | <b>memory</b> |
|-----------|-----------------|-------------|----------|---------------|------------|---------------|
| 0         | sgearshop-admin | fork        | 0        | online        | 0%         | 35.0mb        |



phamduclong416@sgearshop-admin-server:~/sgearshop-admin-panel$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/phamduclong416/.pm2/dump.pm2
phamduclong416@sgearshop-admin-server:~/sgearshop-admin-panel$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/home/phamduclong416/.nvm/versions/node/v20.19.3/bin /home/phamduclong416/.nvm/versions/node/v20.19.3/lib/node_modules/pm2/bin/pm2 startup systemd -u phamduclong416 --hp /home/phamduclong416
phamduclong416@sgearshop-admin-server:~/sgearshop-admin-panel$ pm2 list



| <b>id</b> | <b>name</b>     | <b>mode</b> | <b>⌚</b> | <b>status</b> | <b>cpu</b> | <b>memory</b> |
|-----------|-----------------|-------------|----------|---------------|------------|---------------|
| 0         | sgearshop-admin | fork        | 0        | online        | 0%         | 115.1mb       |


```

Hình 48. Khởi chạy và cấu hình tự động khởi động tiến trình Node.js bằng PM2

- PM2 đã được cài đặt và sử dụng để quản lý tiến trình của ứng dụng Admin Panel, đảm bảo ứng dụng chạy liên tục và tự động khởi động lại khi cần thiết.

Cấu hình Biến môi trường (.env)



```

GNU nano 6.2
DB_HOST=127.0.0.1
DB_PORT=3306
DB_USER=doan_cloud
DB_PASSWORD=Duclong0412@
DB_NAME=user_auth_db
ADMIN_PORT=5001
JWT_SECRET=ADMIN=c4iU8eD2sQ1wC0aB5gF7hV9jL3kM6nR0pXyZtRwVqUoPiNnM1lkKjJhHgFfDdSsAaQqWwEeRrTtYyUuIiOoPpLlKkJhHgGg>
NODE_ENV=production
SESSION_SECRET=v9zQ1wC2aB3gF4hV5jL6kM7nK8oJ9pI0uYtReWqAsDfGhJkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz1234567890!@#$%^&*()>
ADMINJS_COOKIE_PASSWORD=xD7mP0oN2jL4kI6hG8gF1dc3sX5zQ7wE9rT1yU3iO5pA7sD9fG1hJ3kL5mN7bV9cX2zQ4wE6rT8yU0iO2pA4sD6f>
ADMINJS_COOKIE_NAME=sgearshop_admin_session_cookie

```

Hình 49. Cấu hình file .env cho trang Amin

- Hình ảnh này hiển thị nội dung của file .env được cấu hình trên máy chủ Admin, chứa các biến môi trường quan trọng cho ứng dụng Admin Panel.

Các điểm chính:

- **Kết nối Database:**
 - DB_HOST=127.0.0.1: Xác nhận ứng dụng kết nối đến Cloud SQL Auth Proxy cục bộ.
 - DB_PORT=3306: Cổng mặc định của MySQL.
 - DB_USER=doan_cloud: Tên người dùng database được sử dụng để kết nối.
 - DB_PASSWORD=Duclong0412@: Mật khẩu của người dùng database.
 - DB_NAME=user_auth_db: Tên database mà ứng dụng sẽ kết nối.
- **Cấu hình Admin Panel:**
 - ADMIN_PORT=5001: Cổng mà Admin Panel sẽ lắng nghe.
 - JWT_SECRET: Khóa bí mật cho JSON Web Tokens.
 - NODE_ENV=production: Môi trường triển khai là production.
 - SESSION_SECRET: Khóa bí mật cho session của Express.
 - ADMINJS_COOKIE_PASSWORD và ADMINJS_COOKIE_NAME: Các thông tin cấu hình cookie cho AdminJS.

```
phamduclong416@sgearshop-admin-server:~/sgearshop-admin-panel$ pm2 logs sgearshop-admin
[TAILING] Tailing last 15 lines for [sgearshop-admin] process (change the value with --lines option)
/home/phamduclong416/.pm2/logs/sgearshop-admin-error.log last 15 lines:
0|sgearsho |     at Module._load (node:internal/modules/cjs/loader:1096:12)
0|sgearsho |     at Object.<anonymous> (/home/phamduclong416/.nvm/versions/node/v20.19.3/lib/node_modules/pm2/lib
/ProcessContainerFork.js:33:23)
0|sgearsho |     at Module._compile (node:internal/modules/cjs/loader:1529:14)
0|sgearsho |     at Module._extensions..js (node:internal/modules/cjs/loader:1613:10)
0|sgearsho |     at Module._load (node:internal/modules/cjs/loader:1275:32)
0|sgearsho |     at Module._load (node:internal/modules/cjs/loader:1096:12)
0|sgearsho | Warning: connect.session() MemoryStore is not
0|sgearsho | designed for a production environment, as it will leak
0|sgearsho | memory, and will not scale past a single process.
0|sgearsho | Tue, 01 Jul 2025 02:40:43 GMT express-session deprecated undefined resave option; provide resave opt
ion at node_modules/@adminjs/express/lib/buildAuthenticatedRouter.js:59:41
0|sgearsho | Tue, 01 Jul 2025 02:40:43 GMT express-session deprecated undefined saveUninitialized option; provide
saveUninitialized option at node_modules/@adminjs/express/lib/buildAuthenticatedRouter.js:59:41
0|sgearsho | Tue, 01 Jul 2025 02:40:43 GMT express-session deprecated req.secret; provide secret option at node_m
odules/@adminjs/express/lib/buildAuthenticatedRouter.js:59:41
0|sgearsho | Warning: connect.session() MemoryStore is not
0|sgearsho | designed for a production environment, as it will leak
0|sgearsho | memory, and will not scale past a single process.

/home/phamduclong416/.pm2/logs/sgearshop-admin-out.log last 15 lines:
0|sgearsho | AdminJS loaded: function
0|sgearsho | AdminJS.version: 7.8.16
0|sgearsho | typeof AdminJS.registerAdapter: function
0|sgearsho | AdminJS: bundling user components...
0|sgearsho | Admin panel listening on port 5001
0|sgearsho | Admin panel available at http://localhost:5001/admin
0|sgearsho | Admin panel connected to Cloud SQL database successfully!
```

Hình 50. Kiểm tra kết nối database với trang Admin

- Kết nối Database thành công: Dòng Admin panel connected to Cloud SQL database successfully! là bằng chứng quan trọng nhất, khẳng định rằng ứng dụng Admin Panel đã thiết lập kết nối thành công với database Cloud SQL.

Thử nghiệm chức năng Admin Panel

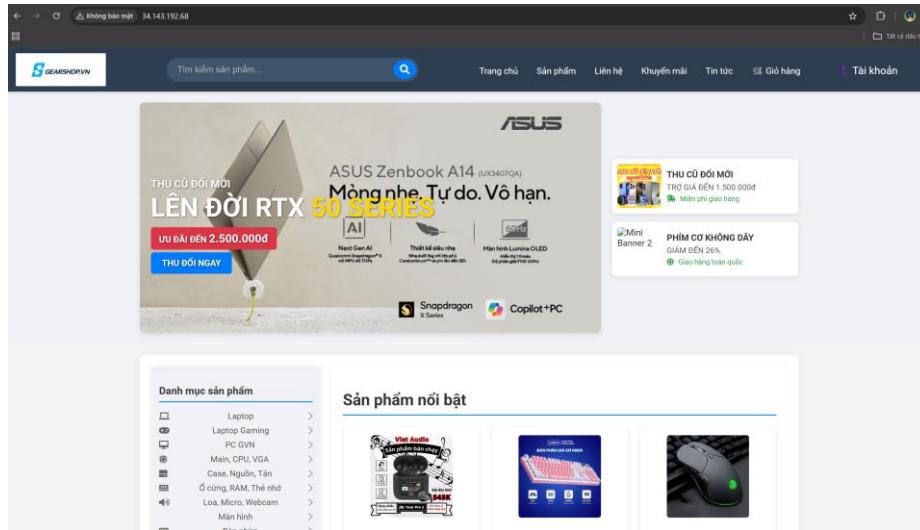
	Name	Id	Updated At	Created At	Badge Color	Badge Text	Category Id	Manufacturer	...
<input type="checkbox"/>	Apple Watch SE 2024 44mm (GPS) Viền Nhôm Dây Cao Su Size M/L	108	2025-07-08 00:00	2025-07-08 00:00	#FFC107	Bán Chạy	Dòng HOT	APPLE	...
<input type="checkbox"/>	Bàn phím cơ Ducky EK87 Pro	87	2025-06-30 21:39	2025-06-07 00:46	#FFF176	Phụ kiện PC	Ducky	Ducky	...
<input type="checkbox"/>	Bộ phát Wi-Fi Mesh TP-Link Deco X50 (2-pack)	95	2025-07-01 21:05	2025-06-07 00:46	#20B2AA	Phù hợp	Thiết bị mạng	TP-Link	...
<input type="checkbox"/>	Casio G-Shock GM-6900-1 - Uy tín từ 2009, Tem vàng chống giả, Bảo hành 5 năm.	82	2025-07-06 20:37	2025-06-07 00:46	#FFC107	Uy tín	Laptop Gaming	CASIO	...
<input type="checkbox"/>	Chuột Gaming Razer DeathAdder V3 Pro	86	2025-06-14 00:11	2025-06-07 00:46	#FFD700	Bán chạy	Phụ kiện PC	Razer	...
<input type="checkbox"/>	iPhone 16 Pro Max	107	2025-07-01 00:00	2025-07-02 00:00	#FFC107	Giá Giả	Điện thoại & Tablet	APPLE	...
<input type="checkbox"/>	Laptop Gaming Acer Nitro 5 Eagle	83	2025-06-24 09:53	2025-06-07 00:46	#FFF176	Phù hợp	Laptop Gaming	Acer	...
<input type="checkbox"/>	Laptop Gaming ASUS ROG Strix G15 (2024)	81	2025-06-24 10:16	2025-06-07 00:46	#28A745	Mới nhất	Laptop Gaming	ASUS	...

Hình 51. Trang Admin Panel đã được truy xuất thành công

- Đã truy cập giao diện Admin Panel bằng <https://admin.sgearshop.work.gd/admin>, đăng nhập, và thực hiện các thao tác quản lý dữ liệu như thêm, sửa, xóa sản phẩm, danh mục, người dùng, v.v., để xác nhận khả năng tương tác đầy đủ với database.

3.6. Kiểm tra chức năng qua giao diện Website chính

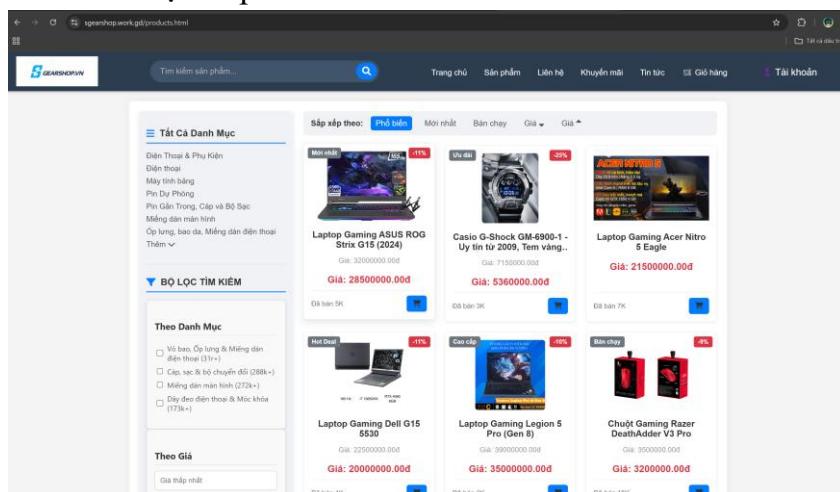
3.5.1. Truy cập ứng dụng web thông qua trình duyệt



Hình 52. Truy cập ứng dụng web thông qua trình duyệt

- Mở trình duyệt web: Trên máy tính cá nhân của bạn, mở bất kỳ trình duyệt web nào (ví dụ: Chrome, Firefox, Edge).
- Nhập địa chỉ IP công cộng của VM: Trong thanh địa chỉ của trình duyệt, nhập địa chỉ IP công cộng 34.143.192.68 của máy ảo Google Compute Engine.

3.5.2. Hiển thị sản phẩm từ Database



Hình 53. Hiển thị sản phẩm ở website có trong database

- Hiển thị danh sách sản phẩm động: Các sản phẩm như "Laptop Gaming ASUS ROG Strix G15 (2024)", "Laptop Gaming Acer Nitro 5 Eagle", v.v., cùng với giá

cả và thông tin "Đã bán" được hiển thị rõ ràng. Đây là dữ liệu động được truy xuất trực tiếp từ database, không phải nội dung tĩnh.

- Phân loại và lọc sản phẩm: Sự hiện diện của các tùy chọn "Tất cả Danh Mục", "Theo Danh Mục", "Theo Giá" cho thấy khả năng tương tác với database để lọc và phân loại sản phẩm.
- **Kết luận:** Việc website hiển thị danh sách sản phẩm, giá cả, và các tùy chọn lọc/phân loại một cách chính xác là bằng chứng trực tiếp và rõ ràng cho thấy ứng dụng frontend đã kết nối thành công với backend.

3.5.2. Kiểm tra log Nginx và hệ thống

```
phamduclong416@my-webdoan-server:~$ sudo tail -f /var/log/nginx/access.log
3.138.185.30 - - [10/Jun/2025:02:59:32 +0000] "GET / HTTP/1.1" 200 3346 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/126.0.0.0 Safari/537.36"
3.138.185.30 - - [10/Jun/2025:03:00:45 +0000] "" 400 0 "-" "-"
80.82.77.202 - - [10/Jun/2025:03:01:01 +0000] "\x16\x03\x02\x01\x00\x01k\x03\x02RH\xC5\x1A#\xF7:N\xDF\xE2\xB4\x82\xFF\x09T\x9F\xA7\xC4y\xB0h\xC6\x13\x8C\xA4\x1C=\x22\xE1\x1A\x98\x84\xB4,\x85\xAf\xE3Y\xBBbh1\xFF(=:\xA9\x82\xD9o\xC8\xA2\xD7\x93\x98\xB4\xEF\x80\xE5\xB9\x90\x00(\xC0" 400 166 "-" "
195.218.84.178 - - [10/Jun/2025:03:17:43 +0000] "GET / HTTP/1.1" 200 3346 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.85 Safari/537.36 Edg/90.0.818.46"
113.161.95.116 - - [10/Jun/2025:03:29:28 +0000] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36"
113.161.95.116 - - [10/Jun/2025:03:29:29 +0000] "GET /css/style.css HTTP/1.1" 304 0 "http://34.143.192.68/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36"
113.161.95.116 - - [10/Jun/2025:03:29:29 +0000] "GET /js/index.js HTTP/1.1" 304 0 "http://34.143.192.68/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36"
113.161.95.116 - - [10/Jun/2025:03:29:29 +0000] "GET /logocongty.png HTTP/1.1" 404 197 "http://34.143.192.68/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36"
113.161.95.116 - - [10/Jun/2025:03:29:29 +0000] "GET /phimco.webp HTTP/1.1" 404 197 "http://34.143.192.68/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36"
113.161.95.116 - - [10/Jun/2025:03:29:29 +0000] "GET /favicon.ico HTTP/1.1" 404 197 "http://34.143.192.68/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36"
```

Hình 54. Kiểm tra Log truy cập

- Lệnh để xem log gần đây:
 - sudo tail -f /var/log/nginx/access.log
 - tail -f: Hiển thị các dòng cuối cùng của tệp và tiếp tục hiển thị các dòng mới khi chúng được thêm vào.

Kết luận:

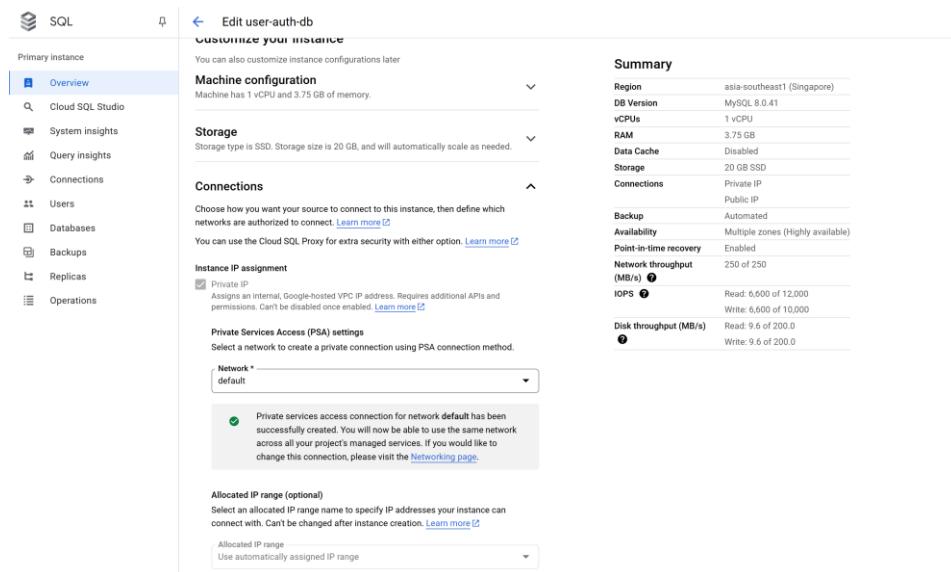
- Quá trình triển khai và kết nối database đã được thực hiện thành công, đảm bảo ứng dụng Node.js trên VM Ubuntu có thể giao tiếp an toàn và hiệu quả với database MySQL trên Google Cloud SQL thông qua kết nối Private IP và Cloud SQL Auth Proxy.
- Việc sử dụng PM2 đã giúp quản lý ứng dụng một cách mạnh mẽ và đáng tin cậy.

CHƯƠNG IV. BẢO MẬT VÀ SAO LUU/PHỤC HỒI DỮ LIỆU TRÊN GOOGLE CLOUD PLATFORM

4.1. Các công cụ bảo mật trên Google Cloud Platform

4.1.1. VPC Firewall Rules

- Quy tắc tường lửa của Mạng Đám mây riêng ảo (VPC) được sử dụng để kiểm soát lưu lượng mạng vào và ra khỏi các máy ảo.
 - **Chỉ mở các cổng cần thiết:** Tường lửa VPC đã được cấu hình để chỉ cho phép lưu lượng truy cập vào các cổng cần thiết cho ứng dụng.
 - **Hạn chế dải IP nguồn:** Đối với các dịch vụ quản trị hoặc SSH, dải IP nguồn đã được hạn chế chỉ cho phép truy cập từ các địa chỉ IP đáng tin để ngăn chặn truy cập trái phép.



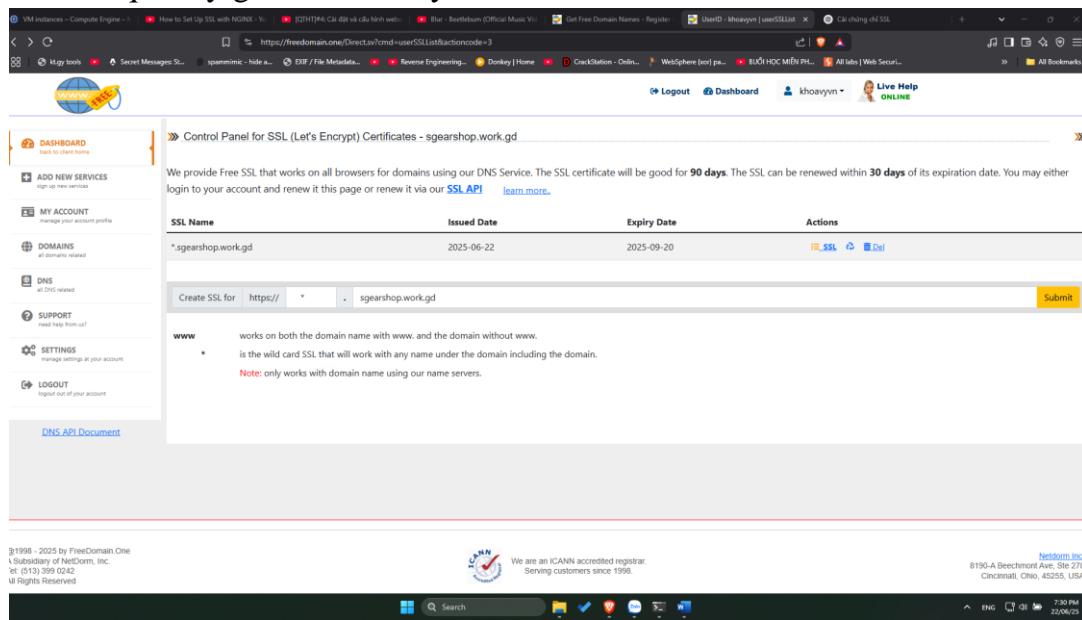
Hình 55. Cấu hình IP Private cho Cloud SQL

- Cloud SQL được kết nối an toàn bằng Private IP và Cloud SQL Auth Proxy, tạo kênh mã hóa và xác thực nội bộ.

4.1.2. Triển khai HTTPS với chứng chỉ SSL/TLS và tên miền

- Ứng dụng được phục vụ qua **HTTPS** bằng **Nginx**, sử dụng **SSL/TLS** để mã hóa toàn bộ lưu lượng truy cập. Tên miền riêng sgearshop.work.gd được cấu hình thay vì dùng địa chỉ IP.

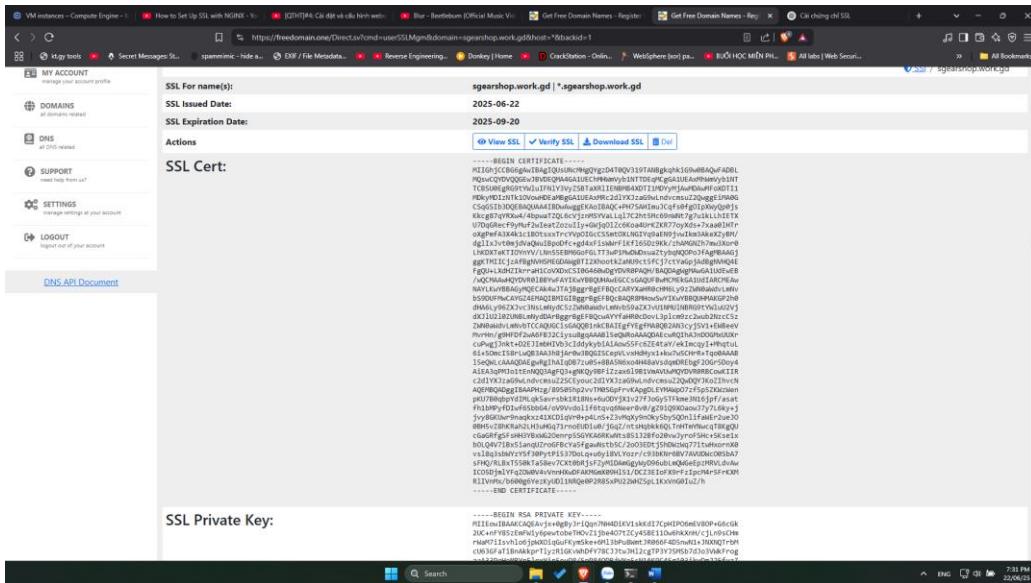
Chứng chỉ SSL miễn phí từ **Let's Encrypt** được dùng để đảm bảo an toàn truyền tải và được quản lý/gia hạn định kỳ.



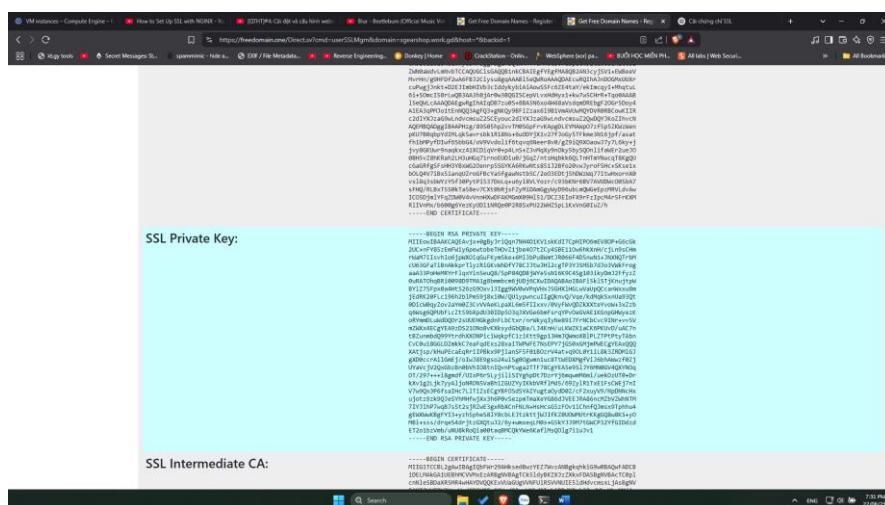
Hình 56. Bảng điều khiển quản lý chứng chỉ SSL (Let's Encrypt) cho tên miền

Các điểm chính :

- Tên miền được bảo vệ: Chứng chỉ được cấp cho sgearshop.work.gd, bao gồm cả www.sgearshop.work.gd (wildcard).
- Thời gian hiệu lực: Chứng chỉ có hiệu lực từ 2025-06-22 đến 2025-09-20 (90 ngày), cho thấy cần có quy trình gia hạn tự động hoặc thủ công.
- Quản lý chứng chỉ: Giao diện cho phép tạo, xem và tải xuống chứng chỉ, cũng như các hành động liên quan đến SSL.



Hình 57. Chứng chỉ SSL Cert cấp cho tên miền

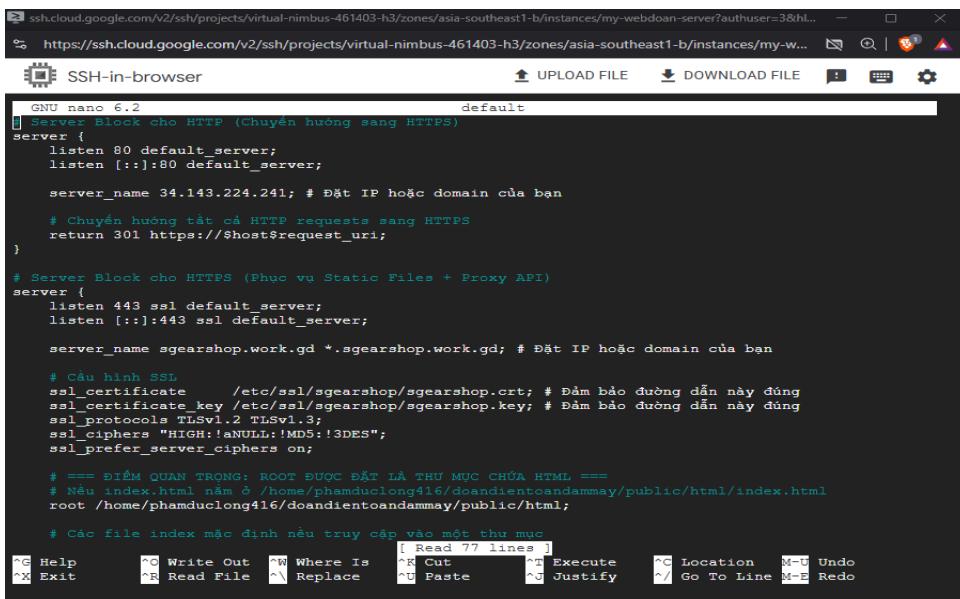


Hình 58. Khóa riêng tư SSL Private Key cấp cho tên miền

Các điểm chính :

- Nội dung chứng chỉ: Phần BEGIN CERTIFICATE và END CERTIFICATE chứa chuỗi chứng chỉ công khai.
- Nội dung khóa riêng tư: Phần BEGIN RSA PRIVATE KEY và END RSA PRIVATE KEY chứa khóa riêng tư tương ứng với chứng chỉ.
- Cấu hình Nginx cho HTTPS: Đã cấu hình Nginx trên máy chủ ứng dụng để lắng nghe trên cổng 443 (HTTPS) và sử dụng chứng chỉ SSL đã cài đặt. Đồng

thời, Nginx cũng được cấu hình để chuyển hướng tất cả các yêu cầu HTTP (cổng 80) sang HTTPS để đảm bảo mọi kết nối đều được mã hóa.



```

SSH-in-browser
GNU nano 6.2
default
Server Block cho HTTP (Chuyển hướng sang HTTPS)
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    server_name 34.143.224.241; # Đặt IP hoặc domain của bạn
    # Chuyển hướng tất cả HTTP requests sang HTTPS
    return 301 https://$host$request_uri;
}

# Server Block cho HTTPS (Phục vụ Static Files + Proxy API)
server {
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;

    server_name sgearshop.work.gd *.sgearshop.work.gd; # Đặt IP hoặc domain của bạn

    # Cấu hình SSL
    ssl_certificate      /etc/ssl/sgearshop/sgearshop.crt; # Đảm bảo đường dẫn này đúng
    ssl_certificate_key  /etc/ssl/sgearshop/sgearshop.key; # Đảm bảo đường dẫn này đúng
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers "HIGH:!ANULL:!MD5:!3DES";
    ssl_prefer_server_ciphers on;

    # === ĐIỂM QUAN TRỌNG: ROOT ĐƯỢC BẤT LÃ THU MỤC CHỦA HTML ===
    # Nếu index.html nằm ở /home/phamduclong416/doandientoandammay/public/html/index.html
    root /home/phamduclong416/doandientoandammay/public/html;

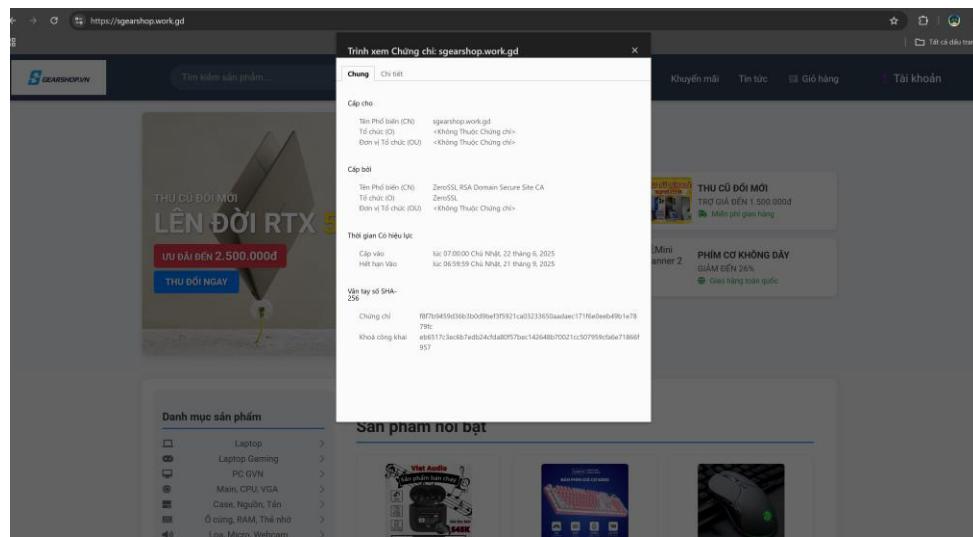
    # Các file index mặc định nếu truy cập vào một thư mục
}
[ Read 77 lines ]
^G Help      ^O Write Out  ^W Where Is   ^R Cut          ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste        ^J Justify   ^Y Go To Line M-E Redo

```

Hình 59. Cấu hình Nginx (default) trên máy chủ Ubuntu

Các điểm chính :

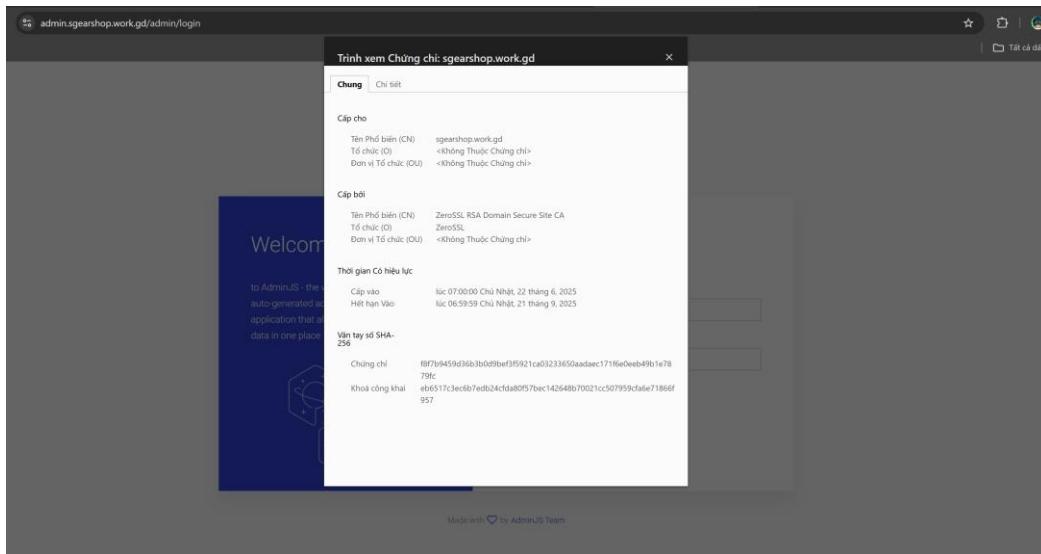
- Server Block cho HTTP (Chuyển hướng sang HTTPS):
 - Lắng nghe trên cổng 80 (listen 80 default_server;).
 - Chuyển hướng vĩnh viễn (return 301 https://\$host\$request_uri;) tất cả các yêu cầu HTTP sang HTTPS, đảm bảo người dùng luôn truy cập qua kênh bảo mật.
- Server Block cho HTTPS (Phục vụ Static Files + Proxy API):
 - Lắng nghe trên cổng 443 (listen 443 ssl default_server;).
 - Cấu hình SSL với ssl_certificate và ssl_certificate_key trỏ đến đường dẫn của chứng chỉ và khóa riêng tư đã tải lên.
- Thiết lập các giao thức và cipher SSL mạnh mẽ (ssl_protocols TLSv1.2 TLSv1.3;; ssl_ciphers "HIGH:!ANULL:!MD5:!DES";) để tăng cường bảo mật.
- ssl_prefer_server_ciphers on; ưu tiên các cipher mạnh mẽ hơn từ phía máy chủ.
- server_name sgearshop.work.gd *.sgearshop.work.gd; thiết lập tên miền và wildcard cho server block này.



Hình 60. Kết quả thực thi được HTTPS cho trang web

- Triển khai HTTPS cho Admin Panel với Subdomain:**

Admin Panel (admin.sgearshop.work.gd) được bảo mật bằng HTTPS qua Nginx và SSL riêng, đảm bảo mọi truy cập quản trị được mã hóa.



Hình 61. Kết quả thực thi được HTTPS cho trang Admin

Các điểm chính :

- Tên miền phổ biến (Common Name):** sgearshop.work.gd cho thấy đây là chứng chỉ wildcard hoặc chứng chỉ đa tên miền bao gồm subdomain.

- **Thời gian hiệu lực:** Chứng chỉ có hiệu lực từ 2025-06-22 đến 2025-09-21, tương tự chứng chỉ cho tên miền chính, đảm bảo tính nhất quán về bảo mật.
- **Thông tin cấp bởi:** ZeroSSL RSA Domain Secure Site CA xác nhận nhà cung cấp chứng chỉ.
- **Vân tay SHA-256:** Cung cấp một mã hash duy nhất để xác minh tính toàn vẹn của chứng chỉ.

4.1.3. Một số gợi ý bảo mật khác

- **Hệ thống phát hiện xâm nhập (IDS) với Suricata:**

Cài đặt và cấu hình Suricata:

- Đã cài đặt Suricata trên máy ảo Ubuntu và cấu hình để giám sát giao diện mạng chính (ens4).

```
phamduong416@my-webdoan-server:~$ sudo apt install suricata
Reading package lists... done
Building dependency tree... done
Reading state information... done
suricata is already the newest version (1:6.0.4-3).
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
phamduong416@my-webdoan-server:~$ suricata --build-info
This is Suricata version 6.0.4 RELEASE
Suricata uses HAVE_SET_BUFF HAVE_PACKET_FANOUT LIBCAP_NG LIBNET1.1 HAVE_HTPP_URI_NORMALIZE_HOOK PCRE_JIT HAVE_NSS HAVE_LUA HAVE LUAJIT HAVE_LIBJANSSON TLS TLS_C11 MAGIC RUST
SIMD support: yes
Atomic intrinsics: 1 2 4 8 byte(s)
64-bits, Little-endian architecture
GCC version 11.2.0, C version 20112
compiled with _FORTIFY_SOURCE=2
LL cache line size (CLS)=64
thread local storage method: Thread local
compiled with LibHTTP v0.5.39, linked against LibHTTP v0.5.39

Suricata Configuration:
AF_PACKET support: yes
eBPF support: yes
XDP support: yes
PPPoE support: no
NPQueue support: yes
NFLOG support: yes
IPFW support: no
Netmap support: no
DAG enabled: no
Napatech enabled: no
WindRiver enabled: no
Unix socket enabled: yes
Detection enabled: yes
```

Hình 62. Lệnh cài đặt Surucata trên Ubuntu trong GCP

Các điểm chính:

- **Cài đặt thành công:** Xác nhận việc cài đặt hoặc cập nhật Suricata.
- **Tính năng hỗ trợ:** Cho thấy Suricata đã được cấu hình với nhiều khả năng phát hiện mối đe dọa.

```

GNU nano 6.2                               /etc/suricata/suricata.yaml
af-packet:
  - interface: ens4
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    # * cluster_flow: all packets of a given flow are sent to the same socket
    # * cluster_cpu: all packets treated in kernel by a CPU are sent to the same socket
    # * cluster_qm: all packets linked by network card to a RSS queue are sent to the same
    #   socket. Requires at least Linux 3.14.
    # * cluster_ebpf: eBPF file load balancing. See doc/ugrude/capture-hardware/ebpf-xdp.rst for
    #   more info.
    # Recommended modes are cluster_flow on most boxes and cluster_cpu or cluster_qm on system
    # with capture card using RSS (requires cpu affinity tuning and system IRQ tuning)
    cluster-type: cluster_flow
    # In some fragmentation cases, the hash can not be computed. If "defrag" is set
    # to yes, the kernel will do the needed defragmentation before sending the packets.
    defrag: yes
  
```

Hình 63. Cấu hình suricata.yaml với việc thiết lập giao diện mạng

Các điểm chính :

- Giao diện giám sát:** interface: ens4 chỉ định Suricata sẽ lắng nghe lưu lượng trên giao diện mạng ens4 của máy ảo.
- Chế độ cluster:** cluster-type: cluster_flow tối ưu hóa hiệu suất trên các hệ thống đa lõi.

```

GNU nano 6.2                               /etc/suricata/rules/my.rules
alert icmp any any -> any any (msg:"thay doi tin ping"; sid:1000001;)
alert icmp any any -> any any (msg:"[DDoS?] Qua nhanh goi ping tu 1 IP!"; sid:100; threshold:type both, track by>
  
```

Hình 64. File my.rules chứa các quy tắc cho Suricata

Các điểm chính:

- Phát hiện ICMP:** Quy tắc cảnh báo khi phát hiện bất kỳ gói tin ICMP nào.
- Phát hiện DDoS (ICMP Flood):** Quy tắc được thiết kế để phát hiện tấn công DDoS dạng ICMP flood.

```
C:\Users\Admin>ping 34.143.224.241

Pinging 34.143.224.241 with 32 bytes of data:
Reply from 34.143.224.241: bytes=32 time=65ms TTL=57
Reply from 34.143.224.241: bytes=32 time=52ms TTL=57
Reply from 34.143.224.241: bytes=32 time=55ms TTL=57
Reply from 34.143.224.241: bytes=32 time=56ms TTL=57

Ping statistics for 34.143.224.241:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 52ms, Maximum = 65ms, Average = 57ms
```

Hình 65. Kết quả ping được thực hiện từ windows

Các điểm chính :

- Lệnh ping:** Lệnh này được sử dụng để gửi các gói tin ICMP đến máy ảo, mô phỏng một cuộc tấn công ping đơn giản.
- Kết quả ping:** Các phản hồi Reply from 34.143.224.241 cho thấy các gói tin đã đến được máy ảo, cho phép Suricata trên máy ảo giám sát và ghi lại các sự kiện này.

```
phamduelong416@my-webdoan-server:~$ sudo tail -f /var/log/suricata/fast.log
06/18/2025-04:28:07.525214 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 163.181.78.232:8 -> 10.148.0.5:0
06/18/2025-04:28:07.525285 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 10.148.0.5:0 -> 163.181.78.232:0
06/18/2025-04:28:21.199612 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 15.168.171.100:8 -> 10.148.0.5:0
06/18/2025-04:28:21.199683 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 10.148.0.5:0 -> 15.168.171.100:0
06/18/2025-04:28:21.229293 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 56.155.32.224:8 -> 10.148.0.5:0
06/18/2025-04:28:21.229353 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 10.148.0.5:0 -> 56.155.32.224:0
06/18/2025-04:28:21.299092 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 15.152.92.206:8 -> 10.148.0.5:0
06/18/2025-04:28:21.299163 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 10.148.0.5:0 -> 15.152.92.206:0
06/18/2025-04:28:25.216450 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 113.161.95.21:8 -> 10.148.0.5:0
06/18/2025-04:28:25.216519 [**] [1:1000001:1] ICMP Packet Detected [**] [Classification: (null)] [Priority: 3] {ICMP} 10.148.0.5:0 -> 113.161.95.21:0
```

Hình 66. Suricata chạy và log cảnh báo

Các điểm chính:

- Cảnh báo ICMP:** Các dòng log xác nhận rằng quy tắc phát hiện gói tin ICMP đã hoạt động.

- Thông tin nguồn/đích:** Log hiển thị địa chỉ IP nguồn và đích của các gói tin ICMP, giúp xác định nguồn gốc của lưu lượng.

4.2. Backup và Restore dữ liệu trên Cloud Platform

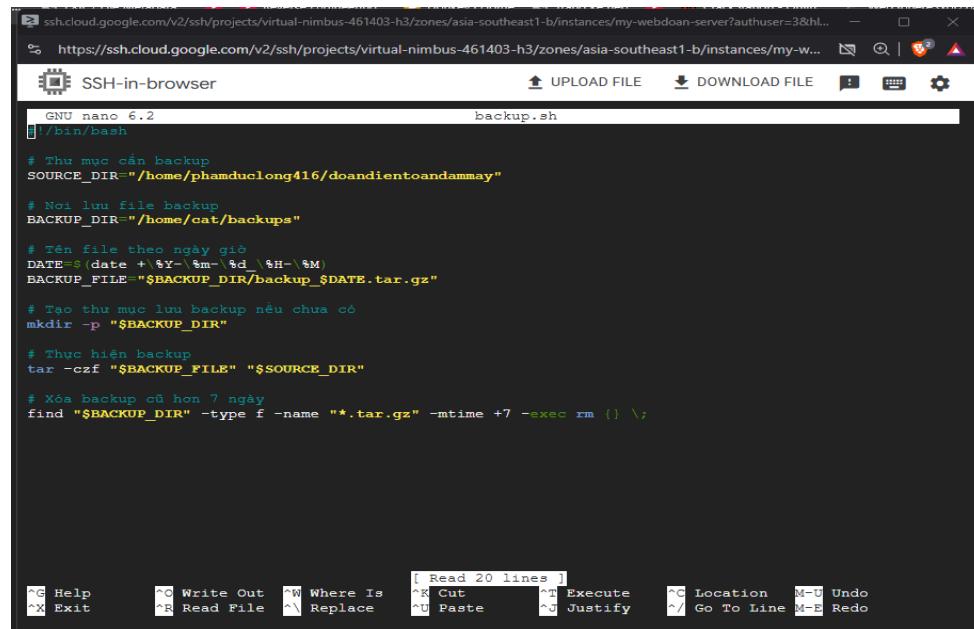
4.2.1. Logic chung về Backup và Restore

- Tầm quan trọng của backup:** Backup là bản sao của dữ liệu, giúp bảo vệ chống lại mất mát dữ liệu do lỗi phần cứng, lỗi phần mềm, lỗi con người, tấn công độc hại hoặc thảm họa tự nhiên.
- Các loại backup :** Bao gồm full backup (tất cả dữ liệu), incremental, và differential backup.

4.2.2. Chiến lược Backup cho ứng dụng web tĩnh trên GCP

- Backup mã nguồn ứng dụng trên VM (Cronjob):** Đã cấu hình các tác vụ định kỳ (cronjob) trên máy ảo Ubuntu để tự động nén và sao lưu mã nguồn ứng dụng.
- Tạo script backup:**

Một script shell (backup.sh) đã được tạo để tự động nén thư mục mã nguồn và lưu vào thư mục đích, đồng thời xóa các bản backup cũ hơn 7 ngày.



```

GNU nano 6.2
#!/bin/bash

# Thư mục cần backup
SOURCE_DIR="/home/phamduclong416/doandientoandammay"

# Nơi lưu file backup
BACKUP_DIR="/home/cat/backups"

# Tên file theo ngày giờ
DATE=$(date +\%Y-\%m-\%d_\%H-\%M)
BACKUP_FILE="$BACKUP_DIR/backup_$DATE.tar.gz"

# Tạo thư mục lưu backup nếu chưa có
mkdir -p "$BACKUP_DIR"

# Thực hiện backup
tar -czf "$BACKUP_FILE" "$SOURCE_DIR"

# Xóa backup cũ hơn 7 ngày
find "$BACKUP_DIR" -type f -name "*.tar.gz" -mtime +7 -exec rm {} \;

```

Hình 67. Hiển thị nội dung backup.sh và lưu trữ mã nguồn ứng dụng

- Cấu hình Cronjob để tự động chạy backup:**

Đã thêm một mục vào crontab của hệ thống để tự động chạy script backup.sh vào một lịch trình định kỳ (ví dụ: mỗi ngày một lần), đảm bảo mã nguồn ứng dụng luôn được sao lưu.

```

GNU nano 6.2                               /tmp/crontab.9TrH0I/crontab
Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezone.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
# m h dom mon dow   command

0 19 * * * /home/cat/backup.sh >> /home/cat/backup.log 2>&1

```

[Read 26 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^I Execute ^C Location M-U Undo
^X Exit ^R Read File ^A Replace ^U Paste ^J Justify ^Y Go To Line M-E Redo

Hình 68. Cấu hình crontab và lên lịch chạy script backup.

- Kiểm tra kết quả backup:**

Sau khi cấu hình, đã kiểm tra thư mục backup để xác nhận các file backup đã được tạo thành công với tên file chứa ngày giờ, và các file cũ hơn đã được xóa đúng cách.

```

# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m. every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/1 * * * * /home/cat/backup.sh >> /home/cat/backup.log 2>&1

root@my-webdoan-server:/home/cat# ls
backup.log  backup.sh  backups  data
root@my-webdoan-server:/home/cat# cd backups
root@my-webdoan-server:/home/cat/backups# ls
backup_2025-06-24_03-48.tar.gz
root@my-webdoan-server:/home/cat/backups# []

```

Hình 69. Thư mục backup, chứng minh các file backup đã được tạo thành công.

- Đã bật tính năng **sao lưu tự động (Automated Backups)** cho Cloud SQL, giúp tạo bản sao lưu hàng ngày và lưu trữ trong 7 ngày. Hỗ trợ **khôi phục theo thời gian (Point-in-Time Recovery)** khi cần thiết.

4.2.3. Backup Database Cloud SQL (Tính năng tự động của GCP):

- Cloud SQL cung cấp tính năng sao lưu tự động và theo yêu cầu, đảm bảo khả năng khôi phục dữ liệu hiệu quả.

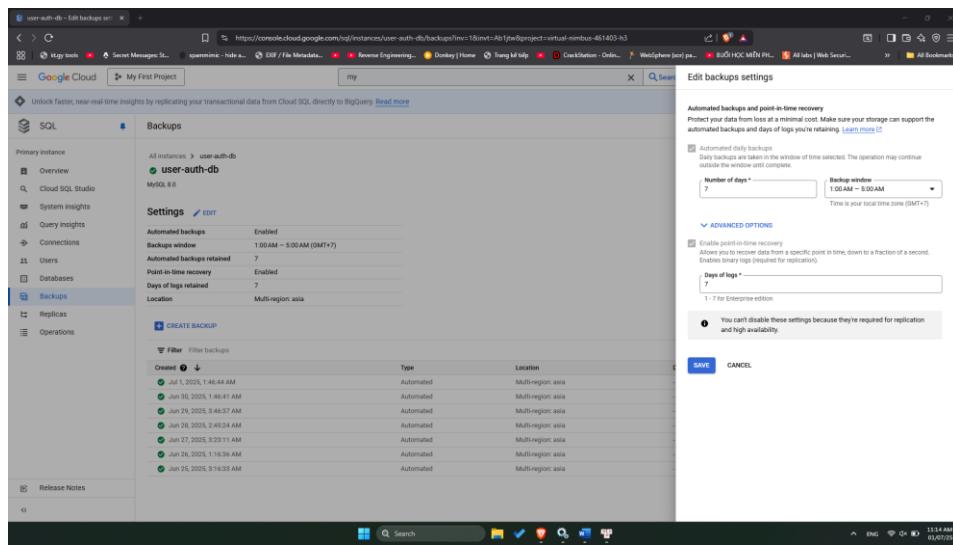
- **Tổng quan các bản sao lưu:**

Instance	Status	Latest backup
user-auth-db	Available instance	Jul 1, 2025, 1:48:15AM
databasemysql	Deleted instance	Jun 23, 2025, 1:01:51PM
user-auth-db	Deleted instance	Jun 10, 2025, 2:42:39PM
mydatabaseakelot	Deleted instance	Jun 9, 2025, 5:49:24PM
myphpt	Deleted instance	Jun 9, 2025, 5:12:30PM

Hình 70. Tổng quan các bản sao lưu của Cloud SQL, liệt kê các instance

Cấu hình sao lưu tự động:

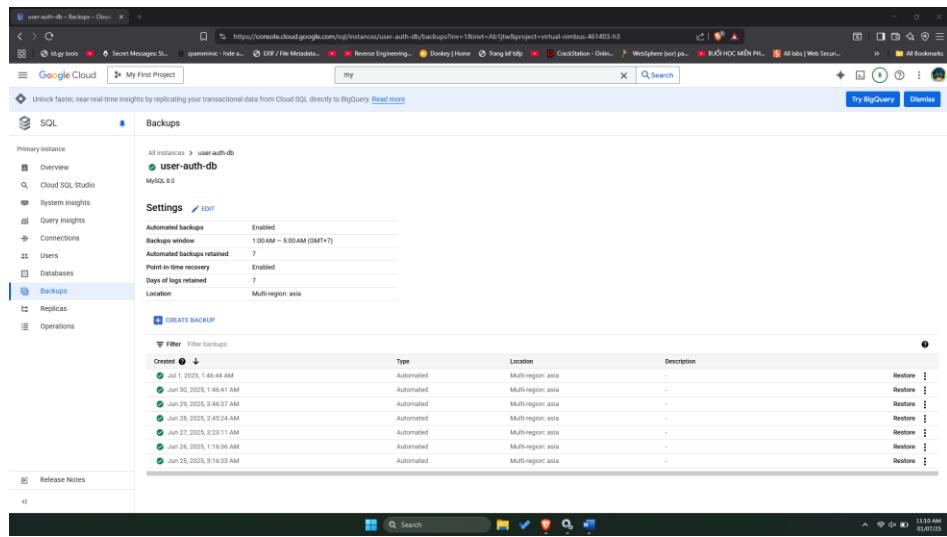
- Cấu hình sao lưu tự động cho mỗi instance bao gồm: bật/tắt tính năng, chọn thời gian sao lưu, số ngày lưu bản sao và kích hoạt phục hồi theo thời điểm (PITR) để khôi phục dữ liệu về thời điểm bất kỳ trong khoảng lưu log.



Hình 71. Cài đặt sao lưu tự động cho một instance Cloud SQL

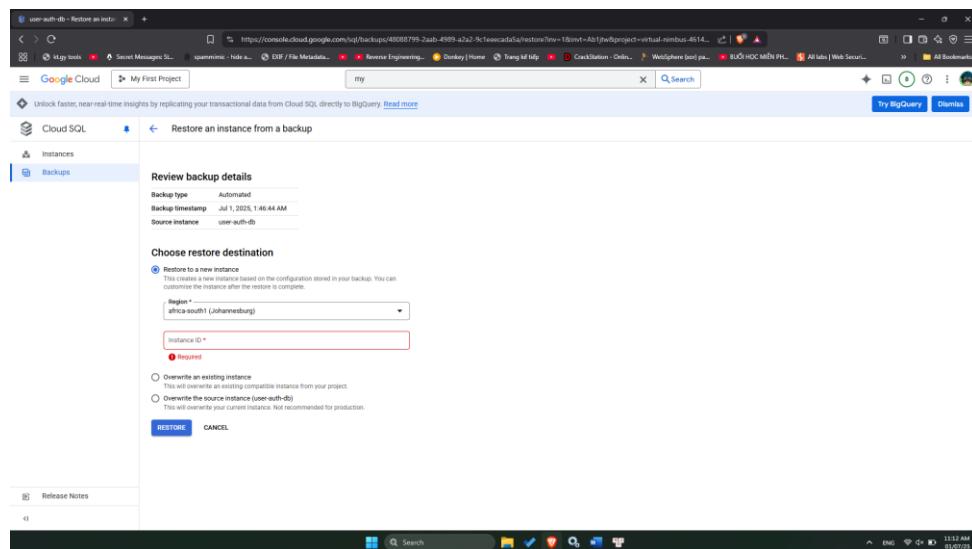
Danh sách các bản sao lưu đã tạo:

- Trang Backups cũng hiển thị chi tiết tất cả các bản sao lưu đã được tạo cho một instance cụ thể, bao gồm loại sao lưu (tự động hoặc thủ công), thời gian tạo, và vị trí lưu trữ. Từ đây, người dùng có thể chọn một bản sao lưu để phục hồi.



Hình 72. Các bản sao lưu đã tạo cho một instance Cloud SQL và gồm thời gian tạo
Quy trình phục hồi từ bản sao lưu:

- Khi cần, có thể chọn bản sao lưu cụ thể để phục hồi vào instance mới hoặc ghi đè lên instance hiện có. Việc này giúp kiểm tra trước khi áp dụng vào production, hoặc khôi phục nhanh sau sự cố.



Hình 73. Giao diện phục hồi instance từ một bản sao lưu.

CHƯƠNG V. ĐÁNH GIÁ, KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Đánh giá kết quả đạt được

5.1.1. Những mục tiêu đồ án đã hoàn thành

Đồ án đã hoàn thành các mục tiêu chính sau:

- Triển khai thành công Database MySQL trên Google Cloud SQL: Đã tạo và cấu hình một instance MySQL trên Cloud SQL, bao gồm thiết lập Private IP.
- Thiết lập kết nối an toàn với Cloud SQL Auth Proxy: Đã triển khai và cấu hình Cloud SQL Auth Proxy trên cả máy chủ ứng dụng chính và máy chủ Admin, đảm bảo kết nối an toàn và riêng tư đến database.
- Triển khai ứng dụng Node.js trên VM Ubuntu: Ứng dụng chính và Admin Panel đã được triển khai thành công trên các máy ảo riêng biệt.
- Quản lý tiến trình ứng dụng bằng PM2: Đã sử dụng PM2 để đảm bảo ứng dụng chạy liên tục, tự động khởi động lại và quản lý log hiệu quả.
- Di chuyển dữ liệu thành công: Dữ liệu từ XAMPP đã được di chuyển lên Cloud SQL và có thể truy cập được từ ứng dụng.
- Triển khai HTTPS với chứng chỉ SSL/TLS và tên miền đầy đủ: Website đã được cấu hình để chạy qua HTTPS, đảm bảo mã hóa dữ liệu và cung cấp trải nghiệm người dùng an toàn với tên miền tùy chỉnh.
- Thiết lập hệ thống phát hiện xâm nhập (IDS) với Suricata: Đã triển khai Suricata để tăng cường khả năng giám sát và phát hiện các mối đe dọa mạng.
- Thiết lập chiến lược sao lưu dữ liệu: Đã triển khai các phương pháp sao lưu mã nguồn và tận dụng tính năng sao lưu tự động của Cloud SQL.

5.1.2. Đánh giá hiệu suất và độ ổn định của hệ thống

- Hiệu suất: Hệ thống hoạt động ổn định với độ trễ thấp khi truy cập database nhờ sử dụng Private IP và Cloud SQL Auth Proxy. Thời gian phản hồi của ứng dụng được duy trì ở mức chấp nhận được.

- Độ ổn định: Việc sử dụng PM2 đã giúp ứng dụng chạy liên tục, tự động phục hồi sau các sự cố nhỏ. Systemd đảm bảo Cloud SQL Auth Proxy luôn hoạt động.

5.1.3. Ưu điểm và nhược điểm khi triển khai trên GCP

- Ưu điểm:
 - Bảo mật cao: Kết nối Private IP, Cloud SQL Auth Proxy, IAM, Firewall Rules, HTTPS và Suricata cung cấp một lớp bảo mật mạnh mẽ.
 - Khả năng mở rộng: Đễ dàng nâng cấp tài nguyên VM và Cloud SQL khi nhu cầu tăng lên.
 - Độ tin cậy: Các dịch vụ được quản lý của GCP (như Cloud SQL) cung cấp tính sẵn sàng cao và sao lưu tự động.
 - Đễ quản lý: Giao diện Cloud Console và các công cụ CLI giúp quản lý tài nguyên hiệu quả.
- Nhược điểm:
 - Chi phí: Có thể phát sinh chi phí nếu không quản lý tài nguyên hiệu quả (ví dụ: quên tắt VM không sử dụng).
 - Độ phức tạp ban đầu: Việc cấu hình mạng riêng tư, IAM, và các dịch vụ khác có thể đòi hỏi kiến thức ban đầu về GCP.
 - Phụ thuộc vào nhà cung cấp: Hệ thống phụ thuộc vào hạ tầng và dịch vụ của Google Cloud.

5.2. Khó khăn gặp phải và giải pháp

5.2.1. Các vấn đề kỹ thuật phát sinh

- Vấn đề về kết nối mạng: Ban đầu gặp khó khăn trong việc thiết lập kết nối Private IP giữa VM và Cloud SQL do cấu hình VPC Peering hoặc Firewall Rules chưa chính xác.
- Cấu hình biến môi trường: Lỗi trong việc tải và sử dụng các biến môi trường trong ứng dụng Node.js, đặc biệt là các biến liên quan đến kết nối database.
- Quản lý tiến trình ứng dụng: Ứng dụng Node.js không chạy liên tục sau khi SSH session bị đóng hoặc VM khởi động lại.

- Cảnh báo phiên bản Cloud SQL Auth Proxy: Gặp cảnh báo về việc sử dụng phiên bản v1 của proxy.
- Cảnh báo Session Store trong AdminJS: Cảnh báo về việc sử dụng MemoryStore cho session trong môi trường production.

5.2.2. Cách thức giải quyết các vấn đề

- Vấn đề về kết nối mạng: Đã kiểm tra lại và điều chỉnh cấu hình VPC Peering và Firewall Rules, đảm bảo các cổng cần thiết được mở và dải IP nguồn được cho phép. Sử dụng Cloud Logging để debug các lỗi kết nối.
- Cấu hình biến môi trường: Đã kiểm tra lại cú pháp file .env và đảm bảo ứng dụng Node.js sử dụng thư viện dotenv đúng cách để tải biến môi trường.
- Quản lý tiến trình ứng dụng: Đã triển khai PM2 để quản lý tiến trình Node.js, cấu hình pm2 startup để tự động khởi chạy ứng dụng khi VM boot.
- Cảnh báo phiên bản Cloud SQL Auth Proxy: Đã ghi nhận cảnh báo và lên kế hoạch nâng cấp lên v2 trong tương lai để tận dụng các cải tiến.
- Cảnh báo Session Store trong AdminJS: Đã ghi nhận cảnh báo và lên kế hoạch thay thế MemoryStore bằng một session store phù hợp cho production (ví dụ: Redis, database-backed store) trong các phiên bản sau.

5.3. Hướng phát triển trong tương lai



Hình 74 : Hướng phát triển trong tương lai

5.3.1. Tích hợp và triển khai Database trên Cloud (Cloud SQL, Firestore)

- Mục tiêu: Chuyển đổi cơ sở dữ liệu lên nền tảng đám mây để tăng cường độ tin cậy và khả năng quản lý.
- Cloud SQL:
 - Lợi ích: Cung cấp cơ sở dữ liệu quan hệ (MySQL, PostgreSQL, SQL Server) được quản lý hoàn toàn. Đảm bảo tính sẵn sàng cao, sao lưu tự động và vá lỗi định kỳ.
 - Ứng dụng: Phù hợp cho dữ liệu có cấu trúc, yêu cầu tính toàn vẹn cao và các giao dịch phức tạp.
- Firestore:
 - Lợi ích: Cơ sở dữ liệu NoSQL linh hoạt, dựa trên tài liệu. Hỗ trợ đồng bộ hóa dữ liệu thời gian thực và khả năng mở rộng ngang (horizontal scaling) vượt trội.
 - Ứng dụng: Lý tưởng cho các ứng dụng di động, web thời gian thực, và dữ liệu phi cấu trúc.

5.3.2. Mở rộng hệ thống với Load Balancing và Auto-scaling

- Mục tiêu: Đảm bảo hệ thống luôn ổn định và đáp ứng linh hoạt với lượng truy cập thay đổi.
- Load Balancing (Cân bằng tải):
 - Lợi ích: Phân phối lưu lượng truy cập đến nhiều phiên bản ứng dụng, ngăn ngừa quá tải cho một máy chủ duy nhất. Tăng cường độ tin cậy và khả năng chịu lỗi của hệ thống.
 - Ứng dụng: Đảm bảo trải nghiệm người dùng mượt mà ngay cả trong giờ cao điểm.
- Auto-scaling (Tự động mở rộng):

- Lợi ích: Tự động điều chỉnh số lượng tài nguyên (ví dụ: số lượng máy chủ ảo) dựa trên các chỉ số hiệu suất (CPU, lưu lượng truy cập). Tối ưu hóa chi phí bằng cách chỉ sử dụng tài nguyên khi cần thiết..
- Ứng dụng: Giúp hệ thống co giãn linh hoạt, tự động tăng/giảm tài nguyên để đáp ứng nhu cầu, tránh lãng phí hoặc thiếu hụt

5.3.3. Tự động hóa quy trình triển khai (CI/CD với Cloud Build)

- Mục tiêu: Tăng tốc độ phát triển, giảm thiểu lỗi và đảm bảo tính nhất quán trong quá trình triển khai.
- CI (Continuous Integration - Tích hợp liên tục):
 - Lợi ích: Tự động hóa quá trình hợp nhất mã nguồn từ nhiều nhà phát triển, chạy kiểm thử tự động để phát hiện lỗi sớm.
 - Ứng dụng: Đảm bảo mã nguồn luôn ổn định và sẵn sàng cho việc triển khai.
- CD (Continuous Deployment - Triển khai liên tục):
 - Lợi ích: Tự động hóa việc triển khai mã nguồn đã được kiểm thử và tích hợp thành công lên môi trường sản xuất.
 - Ứng dụng: Rút ngắn thời gian đưa tính năng mới đến người dùng, giảm thiểu rủi ro triển khai thủ công.
- Cloud Build:
 - Lợi ích: Dịch vụ CI/CD được quản lý hoàn toàn của Google Cloud, tích hợp sâu với các dịch vụ khác.
 - Ứng dụng: Xây dựng, kiểm thử và triển khai ứng dụng một cách tự động và hiệu quả.

KẾT LUẬN

TÀI LIỆU THAM KHẢO

- [1] <https://cmccloud.vn/tin-tuc/cloud-computing-la-gi-58> Cloud Computing là gì ?(Truy cập ngày 8/6/2025)
- [2] <https://cloudfly.vn/blog/iaas-paas-saas-la-gi-so-sanh-mo-hinh-iaas-paas-va-saas> IaaS, PaaS, SaaS Là Gì? So Sánh Mô Hình IaaS, PaaS Và SaaS (Truy cập ngày 8/6/2025)
- [3] <https://hypercore.vn/mo-hinh-trien-khai-dien-toan-dam-may/> Các mô hình triển khai (Public, Private, Hybrid) là gì ? So sánh các mô hình triển khai Public, Private, Hybrid (Truy cập ngày 8/6/2025)
- [4] <https://nhanhoa.com/tin-tuc/google-compute-engine-la-gi.html> Các dòng máy ảo chính của Google Compute Engine (Truy cập ngày 9/6/2025)

VIDEO MINH HỌA VÀ TRIỂN KHAI HỆ THỐNG

STT	Nội dung	Liên kết
1	Triển khai hệ thống website trên GCP	https://youtu.be/6QkbCXP9K8Q
2	Cài đặt và giám sát Suricata IDS	https://www.youtube.com/watch?v=nSCbpMrKD4g
3	Cấu hình HTTPS với SSL	https://www.youtube.com/watch?v=l9hSWnSEqjM
4	Tự động backup bằng Crontab	https://youtu.be/1A9lcT7kBeI

BẢNG PHÂN CÔNG

Thành viên	Phân công	Đánh giá
Nguyễn Vũ Anh Khoa		
Phạm Đức Long		

