

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH

Đề tài: Khai triển và giám sát một hệ thống ứng dụng đơn giản trên nền tảng Kubernetes sử dụng AWS

Chuyên ngành: Mạng máy tính và truyền thông dữ liệu

Giảng viên hướng dẫn : ThS. Trần Hồng Nghi
Mã lớp học : NT114.MMCL.P21

Nhóm sinh viên thực hiện:

1. 21520202 Hồ Hải Dương
2. 21521839 Trần Ngọc Phương Anh

TP HỒ CHÍ MINH, THÁNG 7 NĂM 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Thông tin nhóm sinh viên thực hiện đề tài:

STT	Mã số SV	Họ & tên SV	Mã lớp học
1	21520202	Hồ Hải Dương (Chủ nhiệm đề tài)	NT114.P21.MMCL
2	21521839	Trần Ngọc Phương Anh (Thành viên)	

Tên đề tài: Khai triển và giám sát một hệ thống ứng dụng đơn giản trên nền tảng Kubernetes sử dụng AWS (Deployment and Monitoring of a Simple Application System on Kubernetes Platform using AWS).

Ghi chú về đề tài: This project aims to deploy and monitor a basic web application on a Kubernetes cluster using AWS infrastructure, with tools like Prometheus and Grafana for system observability.

Thời gian thực hiện đề tài: Học kỳ 2, Năm học 2024–2025

Đánh giá của giảng viên hướng dẫn:

Điểm: Bảng chữ:

TP Hồ Chí Minh, ngày tháng 07 năm 2025
Giảng viên hướng dẫn

ThS. Trần Hồng Nghi

LỜI CẢM ƠN

Trước hết, nhóm sinh viên xin chân thành cảm ơn Khoa Mạng máy tính và Truyền thông cùng Trường Đại học Công nghệ Thông tin – Đại học Quốc gia TP.HCM đã xây dựng chương trình học tập chất lượng và tạo điều kiện thuận lợi để chúng em có cơ hội thực hiện đồ án chuyên ngành. Đây là một bước đệm quan trọng để sinh viên vận dụng kiến thức lý thuyết vào thực tế, phát triển tư duy phân tích và khả năng giải quyết vấn đề một cách toàn diện.

Đặc biệt, nhóm xin gửi lời tri ân sâu sắc đến Cô ThS. Trần Hồng Nghi – giảng viên hướng dẫn đề tài, người đã luôn tận tâm đồng hành, theo sát tiến độ và kịp thời đưa ra những định hướng, góp ý quý báu để nhóm hoàn thiện sản phẩm đúng mục tiêu và đúng tiến độ. Sự hỗ trợ của cô là động lực và kim chỉ nam để chúng em vượt qua những khó khăn trong quá trình khai triển hệ thống thực tế trên nền tảng đám mây.

Đồ án không chỉ là một thử thách học thuật, mà còn là cơ hội để chúng em rèn luyện kỹ năng làm việc nhóm, kỹ năng tự học, và đặc biệt là khả năng vận hành hệ thống thực tế với các công nghệ hiện đại.

Chúng em hiểu rằng kiến thức là vô tận, trong khi khả năng tiếp thu của mỗi người là giới hạn. Mặc dù đã nỗ lực tìm tòi, nghiên cứu và hoàn thiện đồ án, nhưng chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được những góp ý để cải thiện và tiến bộ hơn trong tương lai.

Trân trọng,

TP Hồ Chí Minh, tháng 7 năm 2025

Nhóm sinh viên thực hiện:

- **21520202 – Hồ Hải Dương**
- **21521839 – Trần Ngọc Phương Anh**

MỤC LỤC NỘI DUNG

A. PHẦN MỞ ĐẦU	1
1. BỐI CẢNH CHỌN ĐỀ TÀI	1
2. MỤC TIÊU ĐỀ TÀI	1
3. PHẠM VI THỰC HIỆN	2
4. LINK SLIDE VÀ VIDEO DEMO	2
B. NỘI DUNG CHÍNH	3
1. ĐIỆN TOÁN ĐÁM MÂY	3
1.1. Tổng quan	3
1.2. Lợi ích của điện toán đám mây	3
1.3. Các mô hình dịch vụ điện toán đám mây	3
1.4. Kiến trúc tổng quát của điện toán đám mây	4
2. CÔNG NGHỆ SỬ DỤNG	5
2.1. Nhóm Cloud Infrastructure	5
2.1.1. AWS	5
2.1.1.1. Tổng quan	5
2.1.1.2. Tính năng nổi bật của AWS	6
2.1.2. Các dịch vụ AWS sử dụng trong đồ án	6
2.1.2.1. EC2	7
2.1.2.2. S3	7
2.1.2.3. IAM	7
2.1.2.4. Route 53	7
2.1.2.5. Elastic Load Balancer	7

2.1.2.6. Auto Scaling	8
2.2. Nhóm Orchestration	8
2.2.1. Kubernetes	8
2.2.1.1. Tổng quan	8
2.2.1.2. Kiến trúc của Kubernetes	9
2.2.1.3. Các thành phần chính của Kubernetes.....	10
2.2.1.4. Containers trong Kubernetes	10
2.2.1.5. Ưu điểm khi sử dụng	11
2.2.2. kOps	12
2.2.2.1. Tổng quan	12
2.2.2.2. Lợi ích khi sử dụng.....	12
2.3. Nhóm Monitoring.....	13
2.3.1. Prometheus.....	13
2.3.1.1. Tổng quan	13
2.3.1.2. Ưu điểm khi sử dụng	13
2.3.2. Grafana.....	14
2.3.2.1. Tổng quan	14
2.3.2.2. Ưu điểm khi sử dụng	14
2.4. Nhóm Web Application.....	15
2.4.1. WordPress	15
2.4.1.1. Tổng quan	15
2.4.1.2. Ưu điểm của WordPress	15
2.4.1.3. WordPress trong đè tài	16
2.4.2. MariaDB.....	16

2.4.2.1. Tổng quan	16
2.4.2.2. Tính năng nổi bật của MariaDB	17
2.4.2.3. MariaDB trong đê tài	17
2.5. Nhóm Client & Tooling	17
AWS CLI	18
kubectl	18
WSL2	19
2.6. Khác (Hostinger)	19
2.6.1. Tổng quan	19
2.6.2. Lợi ích khi sử dụng	20
3. KHAI TRIỀN HỆ THỐNG	21
3.1. Kiến trúc tổng thể	21
3.2. Các bước khai triển chi tiết	23
3.2.1. Chuẩn bị môi trường cài đặt	23
3.2.2. Tạo S3 và Route 53 cho kOps.....	26
3.2.2.1. Tạo S3 Bucket làm nơi lưu trữ trạng thái của kOps	26
3.2.2.2. Tạo Hosted Zone trên Route 53.....	27
3.2.2.3. Liên kết domain với Route 53	27
3.2.3. Tạo cụm Kubernetes bằng kOps	29
3.2.3.1. Tạo cụm Kubernetes với kOps	29
3.2.3.2. Kiểm tra cụm đã sẵn sàng.....	30
3.2.3.3. Tệp kubeconfig và truy cập kubectl	33
3.3. Khai triển dịch vụ WordPress và MariaDB	33
3.3.1. Khai triển MariaDB	33

3.3.2. Khai triển WordPress	34
3.3.3. Truy cập website từ domain tùy chỉnh.....	35
3.4. Khai triển hệ thống giám sát với Prometheus và Grafana.....	37
3.4.1. Tạo namespace và khai triển Prometheus.....	37
3.4.2. Khai triển Grafana.....	38
3.4.3. Tạo dashboard giám sát cơ bản.....	39
C. KẾT LUẬN	42
1. KẾT QUẢ ĐẠT ĐƯỢC	42
2. NHỮNG HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN	43
2.1. Khó khăn và hạn chế khi thực hiện	43
2.2. Đề xuất hướng phát triển	43
TÀI LIỆU THAM KHẢO.....	44

MỤC LỤC HÌNH ẢNH

Hình 1 – Mô hình tổng quát của điện toán đám mây	5
Hình 2 – Logo AWS	5
Hình 3 – Logo EC2	7
Hình 4 – Logo S3	7
Hình 5 – Logo IAM.....	7
Hình 6 – Logo Route 53.....	7
Hình 7 – Logo ELB.....	7
Hình 8 – Logo AS	8
Hình 9 – Logo Kubernetes	8
Hình 10 – Kiến trúc của Kubernetes.....	9
Hình 11 – Logo kOps	12
Hình 12 – Logo Prometheus	13
Hình 13 – Kiến trúc của Prometheus	13
Hình 14 – Logo Grafana	14
Hình 15 – Logo WordPress.....	15
Hình 16 – Logo MariaDB	16
Hình 17 – Logo Hostinger.....	19
Hình 18 – Mô hình hoạt động của hệ thống.....	21
Hình 19 – Thông tin các node đã khai triển	23
Hình 20 – IAM user đã tạo trên AWS Console	24
Hình 21 – Tạo thành công Access và Secret key trên IAM.....	25
Hình 22 – Các chính sách đã cài trên IAM	25
Hình 23 – AWS CLI đã cài thành công và hoạt động tốt trên WSL2.....	25
Hình 24 – Kết nối AWS từ WSL2 thành công	26
Hình 25 – Bucket đã tồn tại và sẵn sàng được kOps sử dụng.....	26
Hình 26 – Kiểm tra bucket trong S3	26
Hình 27 – Tạo thành công Hosted Zone cho domain gr2p11.site trên Route 53.....	27

Hình 28 – Giao diện website Hostinger sau khi thay đổi Nameservers.....	28
Hình 29 – Đã cập nhật lên hệ thống DNS toàn cầu	28
Hình 30 – Giao diện Route 53 trên AWS sau khi cập nhật tên miền	28
Hình 31 – Chi tiết các record của “gr2p11.site”	29
Hình 32 – Đã tạo thành công Kubernetes cluster trên AWS bằng kOps	30
Hình 33 – Kubernetes Cluster trên AWS đã hoạt động hoàn chỉnh và sẵn sàng khai triển ứng dụng	31
Hình 34 – Kiểm tra trên giao diện Instances EC2 của AWS.....	31
Hình 35 – Chi tiết thông tin của Control Plane trên AWS.....	31
Hình 36 – Chi tiết của Worker Node 1 trên AWS	32
Hình 37 – Chi tiết của Worker Node 2 trên AWS	32
Hình 38 – Cài đặt kubectl thành công và đang dùng phiên bản mới nhất	33
Hình 39 – Cài bản v1.32.0 và chạy kops version thành công	33
Hình 40 – MariaDB đã được deploy thành công vào Kubernetes cluster	34
Hình 41 – Kiểm tra pod đã hoạt động	34
Hình 42 – Đã khai triển WordPress frontend thành công trên Kubernetes cluster...34	34
Hình 43 – WordPress đã được public thành công ra internet thông qua AWS Load Balancer.....	35
Hình 44 – Cả 2 ứng dụng đều đã Running.....	35
Hình 45 – Địa chỉ Load Balancer mà AWS đã cấp để truy cập trình duyệt	35
Hình 46 – Deploy thành công WordPress lên Kubernetes chạy trên AWS.....	36
Hình 47 – Vào được dashboard “WordPress Admin Panel” cho thấy hệ thống đã hoạt động hoàn chỉnh	37
Hình 48 – Khai triển thành công Prometheus vào namespace monitoring.....	38
Hình 49 – Grafana đã được khai triển thành công với LoadBalancer – Có thể truy cập từ trình duyệt.....	39
Hình 50 – Giao diện Grafana	39
Hình 51 – Biểu đồ giám sát mức sử dụng CPU của Prometheus.....	40
Hình 52 – Biểu đồ giám sát số lượng HTTP request đến Prometheus	41

MỤC LỤC BẢNG

Bảng 1 – Các Service của AWS dùng trong đồ án	8
Bảng 2 – Các công cụ “Client & Tooling” dùng trong đồ án	19
Bảng 3 – Thông tin chi tiết của 3 node đã khai triển	22
Bảng 4 – Tổng kết các nội dung đã khai triển thành công.....	42

A. PHẦN MỞ ĐẦU

1. BỐI CẢNH CHỌN ĐỀ TÀI

Trong bối cảnh nhu cầu phát triển ứng dụng web và dịch vụ số ngày càng tăng cao, các tổ chức/doanh nghiệp đòi hỏi hạ tầng khai triển phải bảo đảm tính linh hoạt, khả năng mở rộng và dễ giám sát. Đồng thời, công nghệ điện toán đám mây và container hóa đang ngày càng phổ biến trong thực tế nhờ các ưu điểm về chi phí, tốc độ khai triển và khả năng tự động hóa.

Việc kết hợp giữa **Kubernetes (K8s)** – nền tảng orchestration container phổ biến nhất hiện nay – với các dịch vụ đám mây như **AWS** mang lại giải pháp khai triển ứng dụng web hiệu quả, chuyên nghiệp, và mang tính thực tế cao.

Đề tài này được thực hiện với mục tiêu áp dụng các kiến thức đã học vào khai triển một ứng dụng web thực tế – **WordPress** – trong môi trường Kubernetes, kết hợp với **Prometheus** và **Grafana** để giám sát tài nguyên hệ thống. Việc sử dụng công cụ **kOps** và các dịch vụ AWS như EC2, Route53, IAM, S3, ELB... giúp mô phỏng môi trường khai triển sản phẩm trong doanh nghiệp vừa và nhỏ.

2. MỤC TIÊU ĐỀ TÀI

Khai triển một hệ thống **WordPress** hoạt động ổn định trên cụm **Kubernetes** do công cụ **kOps** tạo ra trên nền tảng **AWS**.

Kết hợp **Prometheus** và **Grafana** để thu thập và trực quan hóa dữ liệu giám sát tài nguyên hệ thống, giúp theo dõi hiệu năng và trạng thái của cụm.

Cấu hình tên miền thân thiện (ví dụ: blog.gr2p11.site) trỏ đến dịch vụ WordPress thông qua **AWS Route 53** và **Elastic Load Balancer (ELB)**.

Làm quen với quy trình khai triển thực tế trên nền tảng cloud, thông qua việc sử dụng các dịch vụ như IAM, S3, EC2, Route53...

Thể hiện năng lực tổng hợp và áp dụng kiến thức từ nhiều môn học: Điện toán đám mây, Quản trị hệ thống, Thiết kế mạng, Mô hình hóa hệ thống phân tán...

3. PHẠM VI THỰC HIỆN

Đề tài tập trung vào việc khai triển và giám sát một ứng dụng web cụ thể là **WordPress**, không phát triển nội dung hoặc chức năng riêng cho hệ thống này.

Hệ thống được khai triển trên cụm **Kubernetes** được tạo bằng **kOps**, chạy trên hạ tầng **AWS EC2** với cấu hình nhỏ gọn nhằm tối ưu chi phí sử dụng.

Việc giám sát hệ thống chỉ giới hạn ở mức sử dụng **Prometheus** để thu thập số liệu và **Grafana** để hiển thị dashboard cơ bản (CPU, Memory, Pod status).

Tên miền có sẵn được sử dụng để cấu hình DNS thông qua **Route53**, nhằm mục đích truy cập hệ thống bằng đường dẫn rõ ràng và chuyên nghiệp.

Hệ thống không khai triển SSL, CI/CD pipeline, autoscaling hoặc backup tự động, nhưng có thể phát triển mở rộng trong tương lai.

4. LINK SLIDE VÀ VIDEO DEMO

https://drive.google.com/drive/folders/1mt4Ra6kn_kKg74bLNYDDLiRsMFzgf9FK?usp=sharing

B. NỘI DUNG CHÍNH

1. ĐIỆN TOÁN ĐÁM MÂY

1.1. Tổng quan

Điện toán đám mây (cloud computing) là việc truy cập theo yêu cầu các tài nguyên máy tính – bao gồm máy chủ vật lý hoặc ảo, lưu trữ dữ liệu, khả năng mạng, công cụ phát triển ứng dụng, phần mềm và các công cụ phân tích được hỗ trợ bởi trí tuệ nhân tạo thông qua internet với mô hình thanh toán dựa trên mức sử dụng.

1.2. Lợi ích của điện toán đám mây

Hiệu quả về chi phí: Điện toán đám mây cho phép doanh nghiệp giảm bớt chi phí và công sức liên quan đến việc mua sắm, cài đặt, cấu hình và quản lý cơ sở hạ tầng tại chỗ. Thay vào đó, doanh nghiệp chỉ trả phí cho các tài nguyên và dịch vụ đám mây mà họ sử dụng.

Tăng tốc độ và tính linh hoạt: Với điện toán đám mây, các ứng dụng doanh nghiệp có thể được khai triển trong vài phút, giúp các nhóm phát triển và DevOps tận dụng nhanh chóng các phần mềm và hạ tầng hỗ trợ trên đám mây.

Khả năng mở rộng không giới hạn: Điện toán đám mây cung cấp tính đàn hồi và khả năng tự phục vụ, cho phép doanh nghiệp điều chỉnh tài nguyên theo nhu cầu thực tế, tránh lãng phí và tối ưu hóa hiệu suất.

Giá trị chiến lược nâng cao: Bằng cách sử dụng điện toán đám mây, doanh nghiệp có thể tiếp cận các công nghệ tiên tiến nhất, như trí tuệ nhân tạo và phân tích dữ liệu lớn, để tạo lợi thế cạnh tranh và thúc đẩy đổi mới.

1.3. Các mô hình dịch vụ điện toán đám mây

IaaS (Infrastructure-as-a-Service): Cung cấp quyền truy cập theo yêu cầu vào các tài nguyên cơ bản như máy chủ, lưu trữ và mạng, cho phép doanh nghiệp linh hoạt trong việc quản lý và khai triển ứng dụng.

PaaS (Platform-as-a-Service): Cung cấp một nền tảng hoàn chỉnh bao gồm phần cứng, phần mềm và hạ tầng để phát triển, chạy và quản lý ứng dụng, giúp đơn giản hóa quá trình phát triển và khai triển.

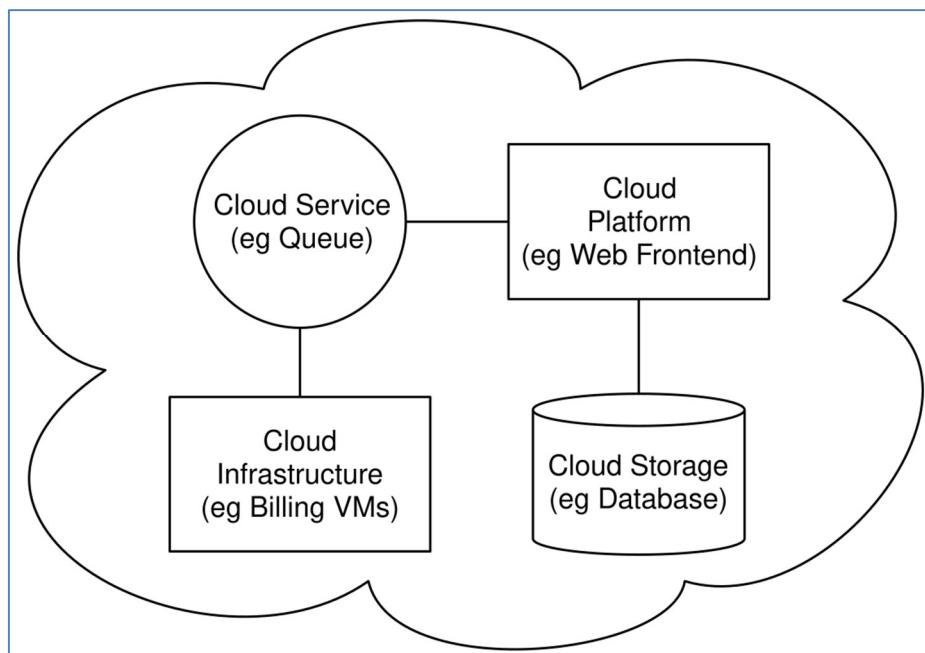
SaaS (Software-as-a-Service): Cung cấp các ứng dụng phần mềm sẵn sàng sử dụng thông qua internet, giúp doanh nghiệp tiết kiệm chi phí và thời gian khai triển. SaaS cho phép truy cập dễ dàng mà không cần cài đặt, giúp tiết kiệm chi phí và thời gian khai triển.

BaaS (Backend-as-a-Service): Cung cấp các dịch vụ backend như lưu trữ dữ liệu, quản lý người dùng, thông báo đẩy và tích hợp mạng xã hội thông qua API. BaaS giúp nhà phát triển tập trung vào giao diện người dùng, giảm bớt công việc liên quan đến quản lý backend.

FaaS (Function-as-a-Service): Cho phép phát triển, khai triển và quản lý các hàm ứng dụng nhỏ mà không cần xây dựng và duy trì hạ tầng. FaaS hoạt động dựa trên các sự kiện, tự động mở rộng tài nguyên khi cần thiết, giúp tối ưu hóa chi phí và tăng tính linh hoạt trong phát triển ứng dụng.

1.4. Kiến trúc tổng quát của điện toán đám mây

Kiến trúc điện toán đám mây bao gồm các thành phần và tiểu thành phần cần thiết cho điện toán đám mây. Các thành phần này thường bao gồm một nền tảng frontend (máy khách dày, máy khách mỏng, di động), các nền tảng backend (máy chủ, lưu trữ), một dịch vụ dựa trên đám mây và một mạng lưới (Internet, Intranet, Intercloud). Kết hợp lại, các thành phần này tạo nên kiến trúc điện toán đám mây, cung cấp dịch vụ linh hoạt và hiệu quả cho người dùng.



Hình 1 – Mô hình tổng quát của điện toán đám mây

2. CÔNG NGHỆ SỬ DỤNG

2.1. Nhóm Cloud Infrastructure

2.1.1. AWS

2.1.1.1. Tổng quan



Hình 2 – Logo AWS

Amazon Web Services (AWS) là nền tảng điện toán đám mây toàn cầu của Amazon, cung cấp hơn 200 dịch vụ tích hợp từ hạ tầng như điện toán, lưu trữ, cơ sở dữ liệu đến các công nghệ tiên tiến như học máy, trí tuệ nhân tạo (AI), và phân tích dữ liệu.

lớn. Với mô hình tính phí dựa trên mức sử dụng, AWS phù hợp cho cá nhân, doanh nghiệp, và tổ chức chính phủ khai triển và quản lý ứng dụng hiệu quả.

2.1.1.2. Tính năng nổi bật của AWS

Tính linh hoạt: AWS cung cấp các dịch vụ như EC2, S3, và Lambda, giúp khai triển ứng dụng trên mọi nền tảng với khả năng tùy chỉnh cao.

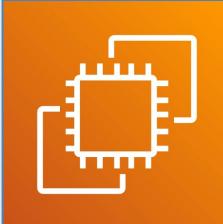
Tính sẵn sàng và bảo mật: Với mạng lưới trung tâm dữ liệu rộng khắp, AWS bảo đảm khả năng chịu lỗi, giảm thiểu rủi ro gián đoạn. Dịch vụ IAM tăng cường bảo mật và quản lý quyền truy cập.

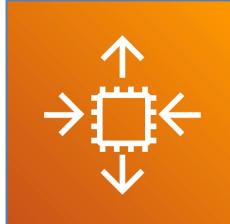
Khả năng mở rộng: AWS hỗ trợ mở rộng tài nguyên tự động (Auto Scaling) để đáp ứng nhu cầu tăng cao và giảm tài nguyên khi không cần thiết, tối ưu chi phí.

Trong lĩnh vực Hệ thống phân tán hoặc Điện toán đám mây, AWS cung cấp môi trường lý tưởng để thực hành khai triển các ứng dụng container hóa với Kubernetes (EKS), quản lý lưu lượng mạng (Route 53, Elastic Load Balancer), và tối ưu hóa hạ tầng với Auto Scaling. AWS cũng hỗ trợ học tập qua việc xây dựng các bài thực hành về quản lý cơ sở dữ liệu (RDS), phân tích dữ liệu (Redshift), và phát triển ứng dụng serverless (Lambda).

2.1.2. Các dịch vụ AWS sử dụng trong đồ án

Dịch vụ	Logo	Tổng quan
---------	------	-----------

2.1.2.1. EC2	 <i>Hình 3 – Logo EC2</i>	Amazon EC2 cung cấp máy chủ ảo trên đám mây với khả năng mở rộng linh hoạt, cho phép người dùng khởi tạo, cấu hình và quản lý các instance phù hợp với nhu cầu. EC2 hỗ trợ nhiều hệ điều hành và cấu hình tài nguyên (CPU, RAM, lưu trữ), giúp tối ưu hóa hiệu suất ứng dụng. Tính năng Auto Scaling kết hợp giúp tự động điều chỉnh số lượng instance dựa trên lưu lượng truy cập.
2.1.2.2. S3	 <i>Hình 4 – Logo S3</i>	Amazon S3 là dịch vụ lưu trữ đối tượng, cho phép lưu trữ và truy xuất dữ liệu ở mọi quy mô với độ bền cao (99.99999999%). S3 hỗ trợ quản lý phiên bản, tích hợp các chính sách quyền truy cập, và tính năng mã hóa dữ liệu nhằm bảo đảm bảo mật. S3 phù hợp cho lưu trữ dữ liệu tĩnh, sao lưu, và phân phối nội dung.
2.1.2.3. IAM	 <i>Hình 5 – Logo IAM</i>	IAM là dịch vụ quản lý quyền truy cập, giúp kiểm soát chi tiết tài nguyên AWS. Người dùng có thể tạo, quản lý danh tính và gán quyền truy cập theo vai trò hoặc nhóm. IAM hỗ trợ chính sách bảo mật mạnh mẽ, bao gồm xác thực đa yếu tố (MFA), bảo đảm việc sử dụng tài nguyên an toàn và hiệu quả.
2.1.2.4. Route 53	 <i>Hình 6 – Logo Route 53</i>	Route 53 là dịch vụ DNS (Domain Name System) giúp định tuyến lưu lượng đến các ứng dụng trên AWS với độ trễ thấp và tính khả dụng cao. Route 53 hỗ trợ quản lý tên miền, kiểm tra trạng thái ứng dụng và tích hợp cân bằng tải, giúp tối ưu hóa trải nghiệm người dùng.
2.1.2.5. Elastic Load Balancer	 <i>Hình 7 – Logo ELB</i>	ELB phân phối lưu lượng truy cập đến các instance EC2 hoặc container, bảo đảm tính sẵn sàng và khả năng chịu tải cao. ELB hỗ trợ cân bằng tải ở các lớp HTTP, HTTPS và TCP, đồng thời tích hợp tính năng giám sát sức khỏe để bảo đảm dịch vụ hoạt động ổn định.

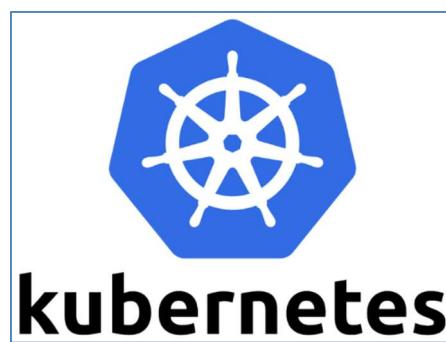
2.1.2.6. Auto Scaling		<p>Auto Scaling tự động điều chỉnh số lượng instance dựa trên nhu cầu thực tế, giúp tối ưu hóa chi phí và hiệu suất. Khi lưu lượng tăng, Auto Scaling mở rộng tài nguyên; khi lưu lượng giảm, tài nguyên sẽ được thu nhỏ, bảo đảm hệ thống luôn hoạt động ở trạng thái tối ưu.</p>
<i>Hình 8 – Logo AS</i>		

Bảng 1 – Các Service của AWS dùng trong đồ án

2.2. Nhóm Orchestration

2.2.1. Kubernetes

2.2.1.1. Tổng quan

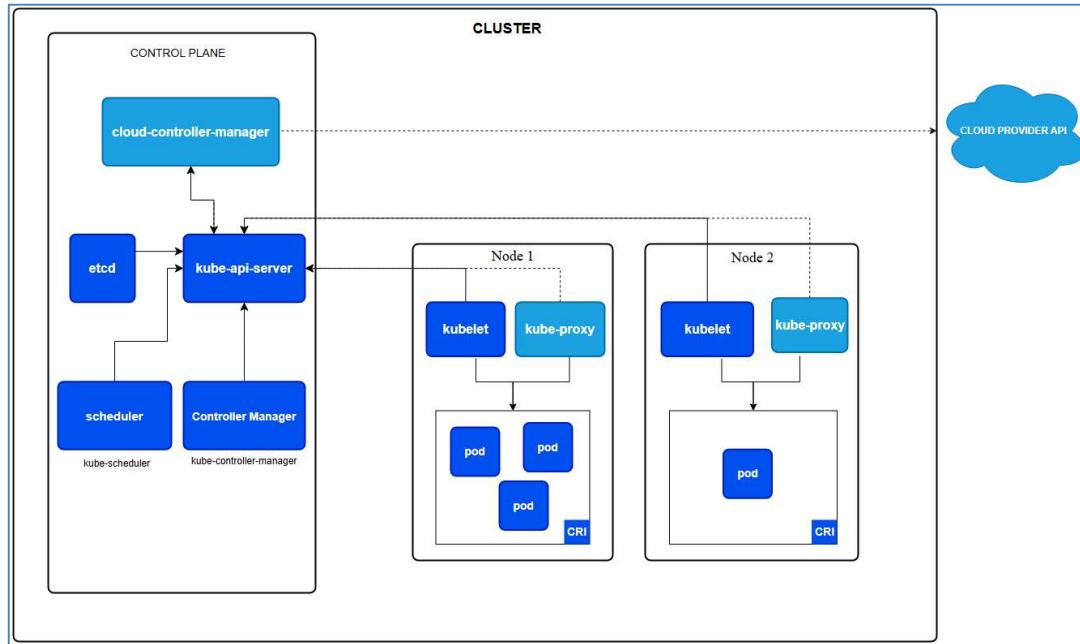


Hình 9 – Logo Kubernetes

Kubernetes, hay **K8s**, là hệ thống mã nguồn mở dùng để tự động hóa việc khai triển, mở rộng và quản lý các ứng dụng dưới dạng container. Kubernetes nhóm các container của một ứng dụng thành các đơn vị logic, giúp dễ quản lý và tìm kiếm. Hệ thống này được xây dựng dựa trên 15 năm kinh nghiệm vận hành khối lượng công việc sản xuất tại Google và các thực tiễn tốt nhất từ cộng đồng.

Kubernetes tạo ra nền tảng để chạy container trên máy chủ vật lý hoặc máy ảo, đồng thời đảm nhận việc phân bổ và quản lý tài nguyên cho container.

2.2.1.2. Kiến trúc của Kubernetes



Hình 10 – Kiến trúc của Kubernetes

Kiến trúc của Kubernetes được chia thành hai phần chính:

Control Plane:

- **kube-apiserver**: Thành phần trung tâm cung cấp API cho toàn bộ cluster, cho phép tương tác và quản lý các tài nguyên.
- **etcd**: Kho lưu trữ key-value phân tán, bảo đảm tính nhất quán và khả dụng cao cho dữ liệu cấu hình của cluster.
- **kube-scheduler**: Chịu trách nhiệm phân bổ các pod đến các node phù hợp dựa trên tài nguyên sẵn có và các ràng buộc đã định nghĩa.
- **kube-controller-manager**: Chạy các controller để giám sát trạng thái của cluster, bảo đảm rằng trạng thái thực tế phù hợp với trạng thái mong muốn.
- **cloud-controller-manager**: Tích hợp với các nhà cung cấp dịch vụ đám mây, quản lý các tài nguyên như máy ảo, lưu trữ và mạng.

Worker Nodes:

- **kubelet**: Chạy trên mỗi node, bảo đảm rằng các container trong pod hoạt động đúng cách và theo dõi trạng thái của chúng.
- **kube-proxy**: Duy trì các quy tắc mạng trên node, cho phép giao tiếp giữa các pod và cân bằng tải cho các dịch vụ.

- **Container Runtime:** Phần mềm chịu trách nhiệm chạy các container, như Docker hoặc containerd.

2.2.1.3. Các thành phần chính của Kubernetes

Pod: Đơn vị khai triển nhỏ nhất trong Kubernetes, chứa một hoặc nhiều container chia sẻ cùng một không gian mạng và lưu trữ.

Service: Định nghĩa một tập hợp các pod và cách truy cập chúng, cung cấp một điểm truy cập ổn định cho các ứng dụng.

Namespace: Cung cấp không gian tên logic để phân tách và quản lý các tài nguyên trong cluster.

Deployment: Quản lý việc khai triển và cập nhật các pod, bảo đảm số lượng bản sao mong muốn luôn hoạt động.

2.2.1.4. Containers trong Kubernetes

Containers, hay còn gọi là vùng chứa, là một phương pháp ảo hóa ở cấp độ hệ điều hành, cho phép đóng gói ứng dụng cùng với tất cả các phụ thuộc của nó – bao gồm thư viện, tệp cấu hình và các thành phần cần thiết khác – thành một đơn vị duy nhất. Điều này bảo đảm rằng ứng dụng có thể chạy nhất quán trên nhiều môi trường khác nhau, từ máy tính cá nhân đến hệ thống đám mây.

Trong Kubernetes, containers được khai triển và quản lý thông qua các Pods – đơn vị khai triển nhỏ nhất trong hệ thống. Mỗi Pod có thể chứa một hoặc nhiều containers, chia sẻ cùng một không gian mạng và lưu trữ, cho phép các containers trong cùng một Pod giao tiếp với nhau một cách hiệu quả.

Kubernetes cung cấp các cơ chế để quản lý vòng đời của containers, bao gồm việc khởi động, giám sát, và tự động khởi động lại khi cần thiết. Điều này giúp bảo đảm rằng ứng dụng luôn ở trạng thái sẵn sàng và hoạt động ổn định.

Khi một container chạy trong Kubernetes, nó được cung cấp một số tài nguyên quan trọng như:

- Hệ thống tệp tin: Kết hợp giữa image của container và các volumes được gắn kết.
- Thông tin về container: Bao gồm hostname (thường là tên của Pod chứa container) và các biến môi trường được định nghĩa trong định nghĩa của Pod.

Thông tin về cluster: Danh sách các services đang chạy khi container được tạo, được cung cấp dưới dạng các biến môi trường hoặc thông qua DNS nếu dịch vụ DNS được kích hoạt.

2.2.1.5. Ưu điểm khi sử dụng

Quản lý ứng dụng:

- Tự động khai triển, khôi phục và tự phục hồi khi có lỗi
- Khám phá dịch vụ, cân bằng tải và gán IP cho Pods
- Quản lý cấu hình và bí mật an toàn

Tài nguyên và lưu trữ:

- Tự động phân bổ tài nguyên tối ưu
- Tích hợp đa dạng hệ thống lưu trữ (NFS, iSCSI)
- Mở rộng linh hoạt theo nhu cầu

Khai triển và phát triển:

- Hỗ trợ CI/CD và quản lý khối lượng công việc
- Phát triển nhanh hơn máy ảo
- Tính nhất quán từ môi trường dev đến production

Kiến trúc và tương thích:

- Hỗ trợ microservices và đa nền tảng
- Dual-stack IPv4/IPv6

- Khả năng mở rộng và tùy chỉnh cao
- Cách ly tài nguyên hiệu quả

Trong đề tài này, Kubernetes được sử dụng để khai triển toàn bộ hệ thống bao gồm cả WordPress (ứng dụng mẫu) và các công cụ giám sát như Prometheus và Grafana.

2.2.2. kOps

2.2.2.1. Tổng quan



Hình 11 – Logo kOps

kOps (Kubernetes Operations) là một công cụ dòng lệnh mã nguồn mở, miễn phí, được thiết kế để cấu hình, duy trì và cung cấp hạ tầng đám mây cần thiết cho việc khai triển các cluster Kubernetes. Với kOps, các nhóm phát triển có thể tự động hóa việc quản lý cluster Kubernetes, bao gồm tạo mới, áp dụng và cập nhật cấu hình cluster một cách hiệu quả.

2.2.2.2. Lợi ích khi sử dụng

Tự động khai triển: Tạo và cấu hình cluster Kubernetes trên AWS, GCP, Azure

Quản lý cấu hình: Quản lý cluster dưới dạng mã (Infrastructure as Code)

Cập nhật an toàn: Nâng cấp cluster với thời gian gián đoạn tối thiểu

Tích hợp DevOps: Kết hợp với CI/CD để tự động hóa toàn bộ quy trình

Trong đề tài này, kOps được sử dụng để khai triển cụm Kubernetes trên AWS, giúp sinh viên không cần cấu hình thủ công từng node, đồng thời bảo đảm cấu trúc hệ thống ổn định và dễ quản lý.

2.3. Nhóm Monitoring

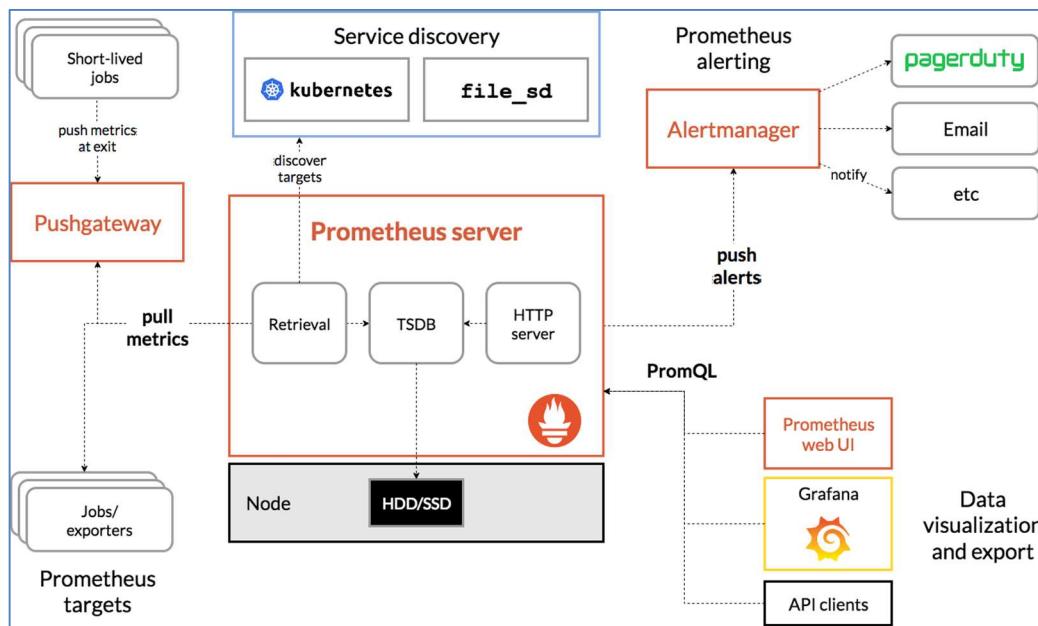
2.3.1. Prometheus

2.3.1.1. Tổng quan



Hình 12 – Logo Prometheus

Prometheus là một hệ thống giám sát và cảnh báo mã nguồn mở, ban đầu được phát triển bởi SoundCloud. Nó được thiết kế để thu thập các chỉ số (metrics) từ các dịch vụ và lưu trữ dữ liệu dạng chuỗi thời gian (time-series). Prometheus được cộng đồng Cloud Native Computing Foundation (CNCF) hỗ trợ mạnh mẽ và là lựa chọn phổ biến trong việc giám sát các hệ thống container hóa như Kubernetes.



Hình 13 – Kiến trúc của Prometheus

2.3.1.2. Ưu điểm khi sử dụng

- Thu thập dữ liệu theo chu kỳ từ các endpoint HTTP có định dạng tiêu chuẩn.

- Hệ cơ sở dữ liệu lưu trữ riêng, không phụ thuộc hệ thống ngoài.
- Ngôn ngữ truy vấn PromQL mạnh mẽ, cho phép phân tích dữ liệu theo thời gian.
- Hỗ trợ cảnh báo thông qua Alertmanager hoặc tích hợp với hệ thống khác (email, Slack...).
- Tích hợp tốt với Kubernetes để theo dõi trạng thái pod, container, service, node, v.v.

Việc tích hợp Prometheus giúp để tài vượt ra ngoài phạm vi khai triển website cơ bản, tiến tới việc theo dõi và quản trị cụm một cách chuyên nghiệp như trong thực tế.

2.3.2. Grafana

2.3.2.1. Tổng quan



Hình 14 – Logo Grafana

Grafana là một nền tảng phân tích và trực quan hóa dữ liệu mã nguồn mở, thường được dùng để hiển thị dữ liệu từ Prometheus và nhiều nguồn khác (MySQL, Elasticsearch, InfluxDB...). Với Grafana, người dùng có thể xây dựng dashboard trực quan để theo dõi hiệu suất hệ thống theo thời gian thực.

2.3.2.2. Ưu điểm khi sử dụng

- Hiển thị dữ liệu theo thời gian (time-series) dưới dạng biểu đồ, đồng hồ, bảng biểu...
- Cho phép tùy biến dashboard linh hoạt phù hợp từng nhu cầu.
- Hỗ trợ chia sẻ dashboard hoặc gắn vào các hệ thống giám sát hiện có.

- Có thể truy vấn dữ liệu từ Prometheus bằng giao diện trực quan hoặc ngôn ngữ truy vấn PromQL.

2.4. Nhóm Web Application

2.4.1. WordPress

2.4.1.1. Tổng quan



Hình 15 – Logo WordPress

Là một hệ quản trị nội dung (CMS – Content Management System) mã nguồn mở phổ biến nhất thế giới, được viết bằng ngôn ngữ lập trình PHP và sử dụng hệ quản trị cơ sở dữ liệu MySQL hoặc MariaDB. WordPress được sử dụng rộng rãi để xây dựng các trang blog cá nhân, website doanh nghiệp, cửa hàng trực tuyến, và cả các hệ thống quản lý nội dung chuyên nghiệp.

2.4.1.2. Ưu điểm của WordPress

- **Dễ sử dụng:** Giao diện quản trị trực quan, phù hợp cả với người không chuyên kỹ thuật.
- **Cộng đồng lớn:** Hàng nghìn plugin, giao diện (themes) và tài liệu hỗ trợ.
- **Linh hoạt và mở rộng:** Hỗ trợ viết plugin tùy chỉnh, dễ tích hợp với các dịch vụ bên ngoài.
- **Mã nguồn mở và miễn phí:** Cho phép sinh viên và doanh nghiệp nhỏ khai triển nhanh với chi phí thấp.

2.4.1.3. WordPress trong đê tài

Trong đê tài này, WordPress được khai triển dưới dạng container trên cụm Kubernetes với các thành phần chính:

- **Pod WordPress:** Chạy ứng dụng chính, được truy cập qua dịch vụ LoadBalancer.
- **Pod MariaDB:** Lưu trữ cơ sở dữ liệu cho WordPress.
- **Persistent Volume:** Bảo đảm dữ liệu được lưu giữ ổn định, không mất khi pod bị xoá hoặc khởi động lại.
- **Service:** Kết nối WordPress với MariaDB và công khai địa chỉ truy cập trang web cho người dùng.

Việc khai triển WordPress trên Kubernetes giúp sinh viên làm quen với các mô hình thực tế của việc **đóng gói ứng dụng, khai triển có thể mở rộng, và kết hợp nhiều thành phần trong một hệ thống phân tán**. Đồng thời, việc giám sát hệ thống WordPress thông qua Prometheus và Grafana còn mang lại góc nhìn tổng thể về **hiệu suất và hoạt động nội tại** của ứng dụng.

2.4.2. MariaDB

2.4.2.1. Tổng quan



Hình 16 – Logo MariaDB

Là một hệ quản trị cơ sở dữ liệu mã nguồn mở, được phát triển như một bản fork của MySQL bởi chính những nhà phát triển ban đầu của MySQL, sau khi MySQL được Oracle mua lại. MariaDB hoàn toàn tương thích với MySQL ở cấp độ cú pháp và chức năng, nên có thể thay thế MySQL trong hầu hết các ứng dụng mà không cần thay đổi mã nguồn.

2.4.2.2. Tính năng nổi bật của MariaDB

- **Hiệu suất cao** và hỗ trợ các công cụ lưu trữ (storage engines) linh hoạt.
- **Tính bảo mật mạnh mẽ**, hỗ trợ SSL, xác thực theo plugin, và mã hóa dữ liệu.
- **Tương thích với MySQL**: Các lệnh SQL, API, cấu trúc dữ liệu, và giao tiếp client đều giữ nguyên.
- **Tốc độ cập nhật nhanh**, do cộng đồng mã nguồn mở tích cực phát triển.

2.4.2.3. MariaDB trong đề tài

Trong đồ án này, MariaDB được sử dụng làm **hệ quản trị cơ sở dữ liệu** cho WordPress. Cụ thể:

- Được khai triển dưới dạng một **Pod riêng biệt trên Kubernetes**.
- Sử dụng **PersistentVolumeClaim** để bảo đảm dữ liệu được lưu trữ ổn định.
- Giao tiếp với pod WordPress thông qua một **service nội bộ** (mariadb), bảo đảm tính cô lập và bảo mật.

MariaDB lưu trữ toàn bộ dữ liệu của website WordPress, bao gồm: nội dung bài viết, thông tin người dùng, cấu hình, plugin, theme, v.v. Việc tách riêng MariaDB giúp hệ thống dễ bảo trì, nâng cấp, và có thể dễ dàng mở rộng (scale) trong tương lai nếu cần.

2.5. Nhóm Client & Tooling

Công cụ	Chức năng chính	Mục đích dùng trong đồ án
---------	-----------------	---------------------------

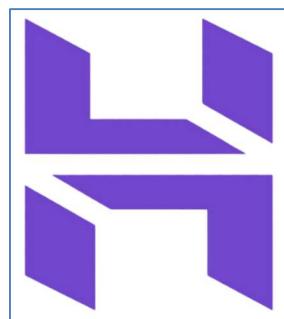
AWS CLI (Amazon Web Services Command Line Interface)	Là công cụ chính thức do AWS cung cấp để tương tác với tất cả các dịch vụ đám mây AWS qua giao diện dòng lệnh. Cho phép người dùng tạo, quản lý, và theo dõi tài nguyên mà không cần sử dụng giao diện đồ họa (Console).	<ul style="list-style-type: none"> - Dùng để tạo S3 Bucket lưu trữ trạng thái cụm Kubernetes cho kOps (kops-state-store). - Tạo và quản lý Hosted Zone trên Route 53 để gán tên miền riêng (gr2p11.site) cho cụm Kubernetes. - Lấy thông tin tài khoản, xác thực và định danh IAM (aws sts get-caller-identity). - Thêm bản ghi DNS (ví dụ: blog.gr2p11.site) để truy cập website WordPress dễ nhớ hơn. - Tự động hóa và kiểm soát tốt hơn toàn bộ quá trình khai triển hạ tầng.
kubectl (Kubernetes Command Line Tool)	Là công cụ dòng lệnh chính thức của Kubernetes, dùng để tương tác với API server của cụm Kubernetes. Cho phép người dùng khai triển, giám sát, quản lý tài nguyên như Pod, Service, Deployment, ...	<ul style="list-style-type: none"> - Khai triển và quản lý WordPress, MariaDB, Prometheus, Grafana dưới dạng YAML thông qua lệnh kubectl apply. - Kiểm tra tình trạng của các tài nguyên (kubectl get pods, kubectl describe, kubectl logs). - Mở rộng (scale) ứng dụng khi cần thiết (kubectl scale deployment). - Giao tiếp và xử lý lỗi nhanh chóng qua dòng lệnh mà không cần dashboard.

WSL2 (Windows Subsystem for Linux version 2)	Là môi trường chạy Linux tích hợp trong Windows 11, giúp người dùng trải nghiệm hệ điều hành Linux một cách đầy đủ mà không cần cài máy ảo riêng biệt.	<ul style="list-style-type: none"> - Cài đặt và chạy được các công cụ như AWS CLI, kubectl, kOps,... trên nền hệ điều hành Ubuntu. - Giúp quá trình khai triển cụm Kubernetes diễn ra ổn định, nhẹ nhàng và tiết kiệm tài nguyên. - Là nền tảng trung gian để kết nối từ máy tính cá nhân chạy Windows đến AWS. - Hỗ trợ tốt thao tác qua terminal cho các công việc DevOps hoặc học thuật.
--	--	---

Bảng 2 – Các công cụ “Client & Tooling” dùng trong đồ án

2.6. Khác (Hostinger)

2.6.1. Tổng quan



Hình 17 – Logo Hostinger

Hostinger là nhà cung cấp hosting chuyên nghiệp cung cấp dịch vụ hosting miễn phí và trả phí. Từ hosting web đẳng cấp thế giới đến trình tạo website AI thế hệ mới, Hostinger cung cấp giải pháp một cửa để giúp bạn thành công trực tuyến

Các dịch vụ chính: Dịch vụ hosting web của Hostinger được chia thành 5 loại: shared hosting, cloud hosting, VPS hosting, managed WordPress hosting và email hosting. Từ Shared Hosting và Domains đến VPS và Cloud plans

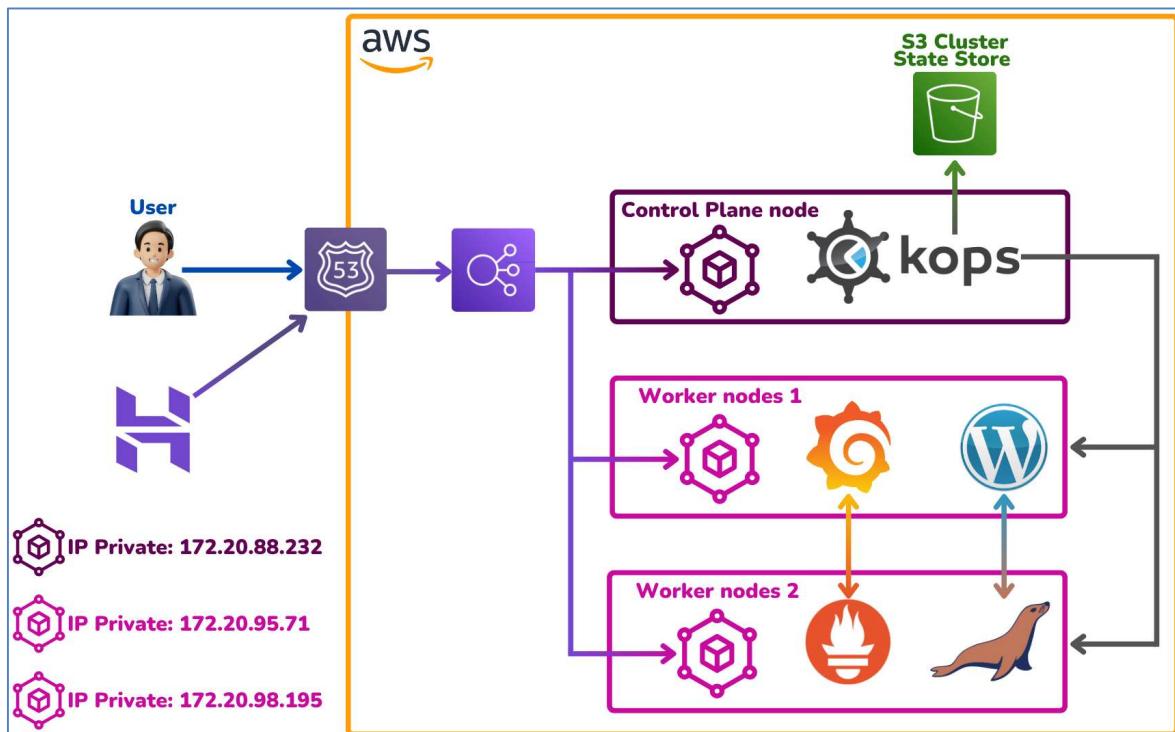
2.6.2. Lợi ích khi sử dụng

- Công nghệ tiên tiến: Hostinger sử dụng các công nghệ tiên tiến như LiteSpeed WebServer và giải pháp caching tích hợp để bảo đảm website tải nhanh.
- Tính linh hoạt: Các dịch vụ shared, cloud và managed WordPress hosting được hỗ trợ bởi máy chủ LiteSpeed và hệ điều hành Linux.
- Hỗ trợ khách hàng: Bảo đảm uptime 99.9% với hỗ trợ 24/7.
- Giá cả phải chăng với nhiều gói hosting đa dạng
- Bảo đảm hoàn tiền 30 ngày, migration website miễn phí, tích hợp với nhiều CMS khác nhau.
- Migration website miễn phí và dễ dàng.
- Gói hosting giá rẻ, nhiều tính năng với cách tiếp cận thân thiện cho cả người mới bắt đầu và người dùng nâng cao.

Hostinger được sử dụng trong đê tài với vai trò là nơi quản lý tên miền gr2p11.site. Thay vì sử dụng dịch vụ DNS mặc định của Hostinger, nhóm thực hiện đã chuyển sang sử dụng dịch vụ Route 53 của AWS bằng cách trả nameservers về phía Amazon. Tuy vậy, tài khoản Hostinger vẫn giữ vai trò quan trọng trong việc sở hữu tên miền chính thức, giúp truy cập vào ứng dụng (như WordPress) thông qua tên miền đẹp và dễ nhớ, thay vì dùng địa chỉ IP hoặc đường dẫn mặc định từ AWS ELB.

3. KHAI TRIỂN HỆ THỐNG

3.1. Kiến trúc tổng thể



Hình 18 – Mô hình hoạt động của hệ thống

Hệ thống được khai triển trên nền tảng đám mây AWS sử dụng công cụ kOps để tạo và quản lý một cụm Kubernetes. Cụm bao gồm 1 node điều khiển (control plane) và 2 node công việc (worker nodes). Mỗi node là một phiên bản EC2 được tự động cấu hình và liên kết trong cùng một vùng (availability zone) là ap-southeast-1a.

Các thành phần chính trong hệ thống bao gồm:

- WordPress:** được khai triển dưới dạng một Deployment với Service kiểu LoadBalancer để cung cấp giao diện web cho người dùng cuối.
- MariaDB:** được khai triển như một Deployment nội bộ để làm cơ sở dữ liệu cho WordPress.
- Prometheus và Grafana:** được khai triển trong namespace riêng (monitoring) để giám sát hoạt động của các thành phần trong cụm.

- **AWS Route 53:** dịch vụ DNS giúp ánh xạ tên miền phụ blog.gr2p11.site đến địa chỉ ELB của WordPress.
- **AWS S3:** nơi lưu trữ trạng thái (state store) của cụm Kubernetes, giúp kOps quản lý trạng thái hệ thống.
- **IAM (Identity and Access Management):** kiểm soát quyền truy cập từ CLI đến các dịch vụ AWS liên quan trong quá trình khai triển.

Hệ thống được thiết kế theo hướng module hóa, dễ dàng mở rộng, giám sát và tái sử dụng cho các mục đích học tập hoặc thực hành khai triển thực tế.

EC2 Instance	i-	i-	i-
ID	0abfa1c3a618c1b9c	0a037e8fd6f085e9b	0c976fe4378d22fb2
IP Private	172.20.88.232	172.20.95.71	172.20.98.195
IP Public (lúc kiểm tra)	18.141.10.12	13.250.123.45	54.251.210.76
Vai trò trong K8s	Control Plane	Worker Node #1	Worker Node #2
Chức năng trong cluster	Điều phối, quản lý cluster (kube-apiserver, scheduler, etcd, kOps controller...)	WordPress, Grafana, CoreDNS, Cilium	MariaDB, Prometheus, CoreDNS, Cilium
Pod	Kube-*, etcd, dns-controller	Wordpress, grafana	Mariadb, prometheus

Bảng 3 – Thông tin chi tiết của 3 node đã khai triển

NAMESPACE	NAME	NOMINATED NODE	READINESS GATES	READY	STATUS	RESTARTS	AGE	IP
default	mariadb-fd5788ff9-fpq98	i-0c976fe4378d22fb2	<none>	1/1	Running	0	15h	100.96.2.123
default	wordpress-69c74d7f78-4fwtg	i-0a037e8fd6f085e9b	<none>	1/1	Running	0	15h	100.96.1.143
kube-system	aws-cloud-controller-manager-9kvz5	i-0abfalc3a618c1b9c	<none>	1/1	Running	0	16h	172.20.88.232
kube-system	aws-node-termination-handler-654cfcf988-c7wv8	i-0abfalc3a618c1b9c	<none>	1/1	Running	0	16h	172.20.88.232
kube-system	cilium-62mhd	i-0a037e8fd6f085e9b	<none>	1/1	Running	0	16h	172.20.95.71
kube-system	cilium-bg8xv	i-0c976fe4378d22fb2	<none>	1/1	Running	0	16h	172.20.98.195
kube-system	cilium-dhfd4	i-0abfalc3a618c1b9c	<none>	1/1	Running	0	16h	172.20.88.232
kube-system	cilium-operator-6b6dd79fffb-hr6sz	i-0abfalc3a618c1b9c	<none>	1/1	Running	0	16h	172.20.88.232
kube-system	coredns-587b47655d-74n8t	i-0a037e8fd6f085e9b	<none>	1/1	Running	0	16h	100.96.1.69
kube-system	coredns-587b47655d-k528h	i-0c976fe4378d22fb2	<none>	1/1	Running	0	16h	100.96.2.70
kube-system	coredns-autoscaler-5f67cf4649-d6c7c	i-0a037e8fd6f085e9b	<none>	1/1	Running	0	16h	100.96.1.254
kube-system	dns-controller-5f99855995-jp298	i-0abfalc3a618c1b9c	<none>	1/1	Running	0	16h	172.20.88.232
kube-system	ebs-csi-controller-647b8c8c5d-q5txj	i-0abfalc3a618c1b9c	<none>	6/6	Running	0	16h	172.20.88.232
kube-system	ebs-csi-node-8dnw4	i-0c976fe4378d22fb2	<none>	3/3	Running	0	16h	100.96.2.136
kube-system	ebs-csi-node-kswsx	i-0a037e8fd6f085e9b	<none>	3/3	Running	0	16h	100.96.1.19
kube-system	ebs-csi-node-zdcdw	i-0abfalc3a618c1b9c	<none>	3/3	Running	0	16h	100.96.0.209
kube-system	etcd-manager-events-i-0abfalc3a618c1b9c	i-0abfalc3a618c1b9c	<none>	1/1	Running	0	16h	172.20.88.232
kube-system	etcd-manager-main-i-0abfalc3a618c1b9c	i-0abfalc3a618c1b9c	<none>	1/1	Running	0	16h	172.20.88.232
kube-system	kops-controller-p2hcc	i-0abfalc3a618c1b9c	<none>	1/1	Running	0	16h	172.20.88.232
kube-system	kube-apiserver-i-0abfalc3a618c1b9c	i-0abfalc3a618c1b9c	<none>	2/2	Running	2 (16h ago)	16h	172.20.88.232
kube-system	kube-controller-manager-i-0abfalc3a618c1b9c	i-0abfalc3a618c1b9c	<none>	1/1	Running	3 (16h ago)	16h	172.20.88.232
kube-system	kube-scheduler-i-0abfalc3a618c1b9c	i-0abfalc3a618c1b9c	<none>	1/1	Running	0	16h	172.20.88.232
monitoring	grafana-59b479666f-hk2lv	i-0a037e8fd6f085e9b	<none>	1/1	Running	0	15h	100.96.1.7
monitoring	prometheus-668455c869-v726m	i-0c976fe4378d22fb2	<none>	1/1	Running	0	15h	100.96.2.62

Hình 19 – Thông tin các node đã khai triển

3.2. Các bước khai triển chi tiết

3.2.1. Chuẩn bị môi trường cài đặt

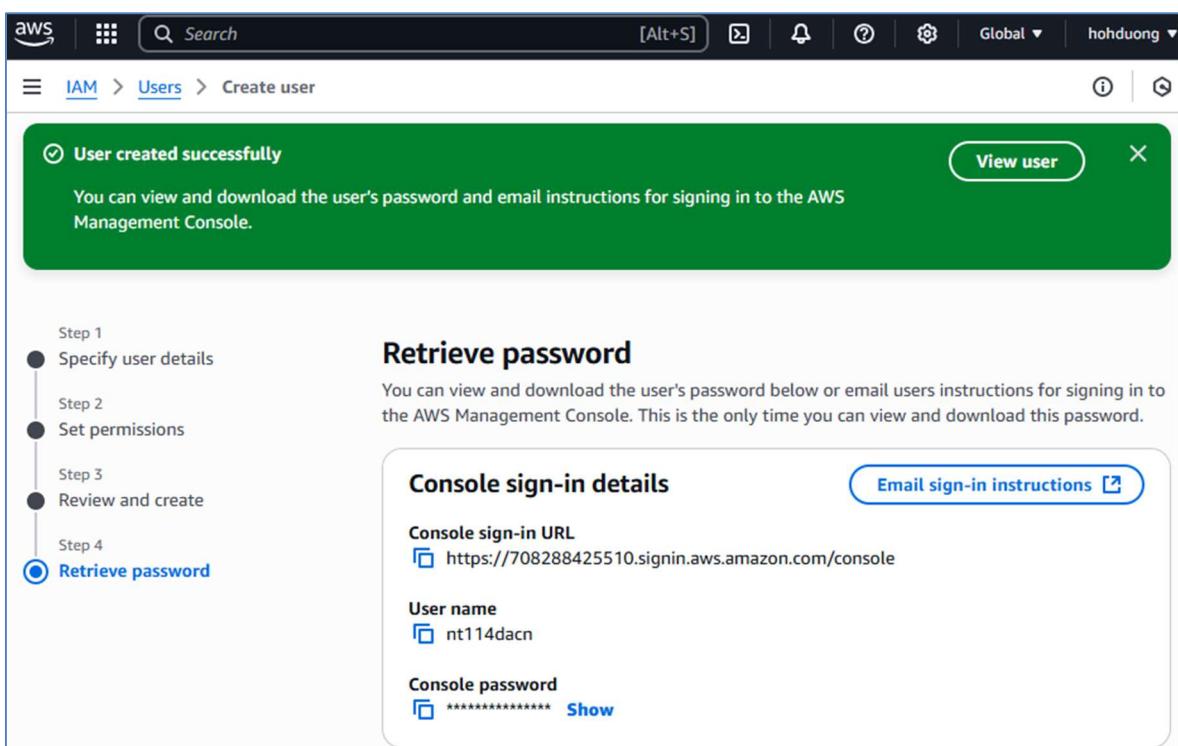
Để bảo đảm việc khai triển hệ thống diễn ra thuận lợi và đồng bộ, nhóm tiến hành chuẩn bị môi trường làm việc trên máy tính cá nhân chạy hệ điều hành Windows 11 với các bước sau:

- WSL2 (Ubuntu): Cho phép sử dụng môi trường Linux thật bên trong Windows, hỗ trợ chạy các lệnh CLI hiệu quả.

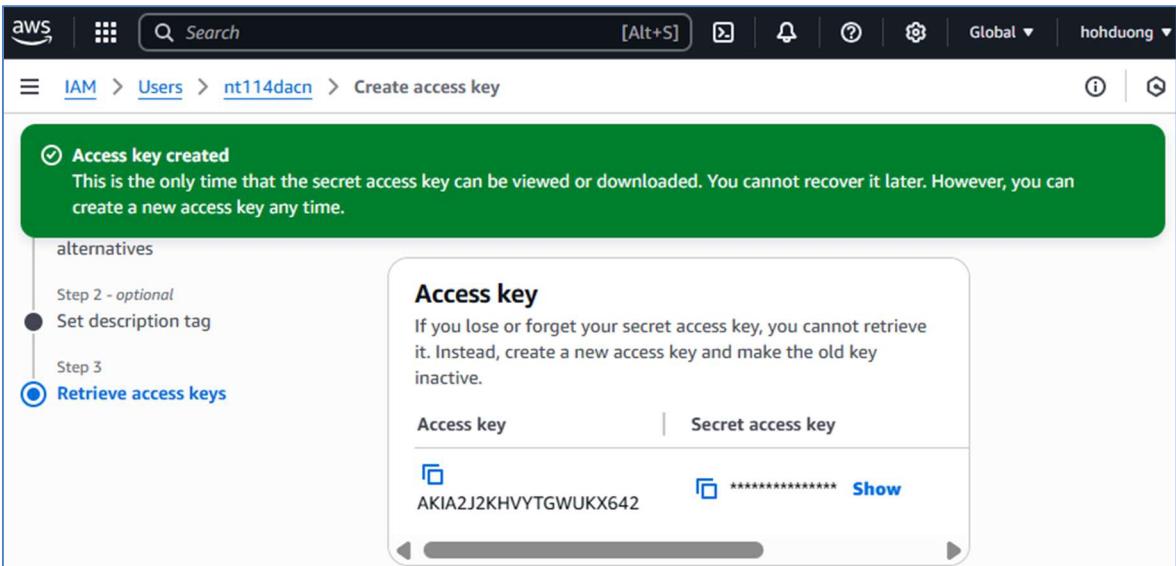
- AWS CLI: Giao tiếp với các dịch vụ của AWS từ dòng lệnh như tạo bucket, quản lý IAM, Route 53...
- kubectl: Công cụ điều khiển và quản trị cụm Kubernetes.
- kOps: Dùng để tạo và quản lý cụm Kubernetes trên AWS một cách tự động và linh hoạt.

Tạo IAM user và cấu hình AWS CLI

- Tạo một IAM user mới với quyền AdministratorAccess.
- Kích hoạt Access Key ID và Secret Access Key, sau đó cấu hình bằng lệnh aws configure.



Hình 20 – IAM user đã tạo trên AWS Console



Hình 21 – Tạo thành công Access và Secret key trên IAM

The screenshot shows the AWS IAM 'User details' page for 'nt114dacn'. The left sidebar shows 'Identity and Access Management (IAM)' and 'Access management' sections. The main area shows the user's summary: ARN (arn:aws:iam::708288425510:user/nt114dacn), Console access (Enabled without MFA), Created (July 04, 2025, 12:42 (UTC+07:00)), Last console sign-in (Never), and two Access keys. The 'Permissions' tab is selected, showing a list of attached policies: 'AdministratorAccess', 'AmazonEC2FullAccess', 'AmazonRoute53FullAccess', 'AmazonS3FullAccess', 'IAMFullAccess', and 'IAMUserChangePassword'. Each policy is listed with its name, type (AWS managed - job function or AWS managed), and attachment status (Directly).

Hình 22 – Các chính sách đã cài trên IAM

```
hoeduong@PhanHD:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
hoeduong@PhanHD:~$ aws --version
aws-cli/2.27.49 Python/3.13.4 Linux/6.6.87.2-microsoft-standard-WSL2 exe/x86_64.ubuntu.24
```

Hình 23 – AWS CLI đã cài thành công và hoạt động tốt trên WSL2

```
hohduong@hpHHD:/mnt/c/Users/HP$ aws sts get-caller-identity
{
    "UserId": "AIDA2J2KHVYTNS6MAK550",
    "Account": "708288425510",
    "Arn": "arn:aws:iam::708288425510:user/nt114dacn"
}
```

Hình 24 – Kết nối AWS từ WSL2 thành công

3.2.2. Tạo S3 và Route 53 cho kOps

Để kOps có thể lưu trữ trạng thái (state store) và cấu hình DNS cho cụm Kubernetes, nhóm thực hiện hai bước quan trọng: tạo S3 bucket và cấu hình Hosted Zone trên dịch vụ Route 53 của AWS.

3.2.2.1. Tạo S3 Bucket làm nơi lưu trữ trạng thái của kOps

kOps yêu cầu một S3 bucket riêng biệt để lưu trữ toàn bộ thông tin cấu hình và trạng thái của cụm Kubernetes. Nhóm đã tạo bucket với tên:

```
aws s3api create-bucket \
--bucket kops-state-store-gr2p11 \
--region ap-southeast-1 \
--create-bucket-configuration LocationConstraint=ap-southeast-1
```

```
hohduong@hpHHD:~$ aws s3api create-bucket \
> --bucket kops-state-store-gr2p11 \
>   --region ap-southeast-1 \
>   --region ap-southeast-1 \
>   --create-bucket-configuration LocationConstraint=ap-southeast-1
{
    "Location": "http://kops-state-store-gr2p11.s3.amazonaws.com/"
}
```

Hình 25 – Bucket đã tồn tại và sẵn sàng được kOps sử dụng

```
hohduong@hpHHD:/mnt/c/Users/HP$ aws s3 ls
2025-07-04 11:49:03 kops-state-store-gr2p11
```

Hình 26 – Kiểm tra bucket trong S3

3.2.2.2. Tạo Hosted Zone trên Route 53

Để kOps có thể tạo và quản lý DNS cho cụm, cần một public hosted zone trùng với domain đã đăng ký (gr2p11.site). Nhóm sử dụng Route 53 để tạo Hosted Zone bằng lệnh:

```
aws route53 create-hosted-zone \
--name gr2p11.site \
--caller-reference $(date +%s) \
--hosted-zone-config Comment="Hosted zone for kOps cluster",PrivateZone=false
```

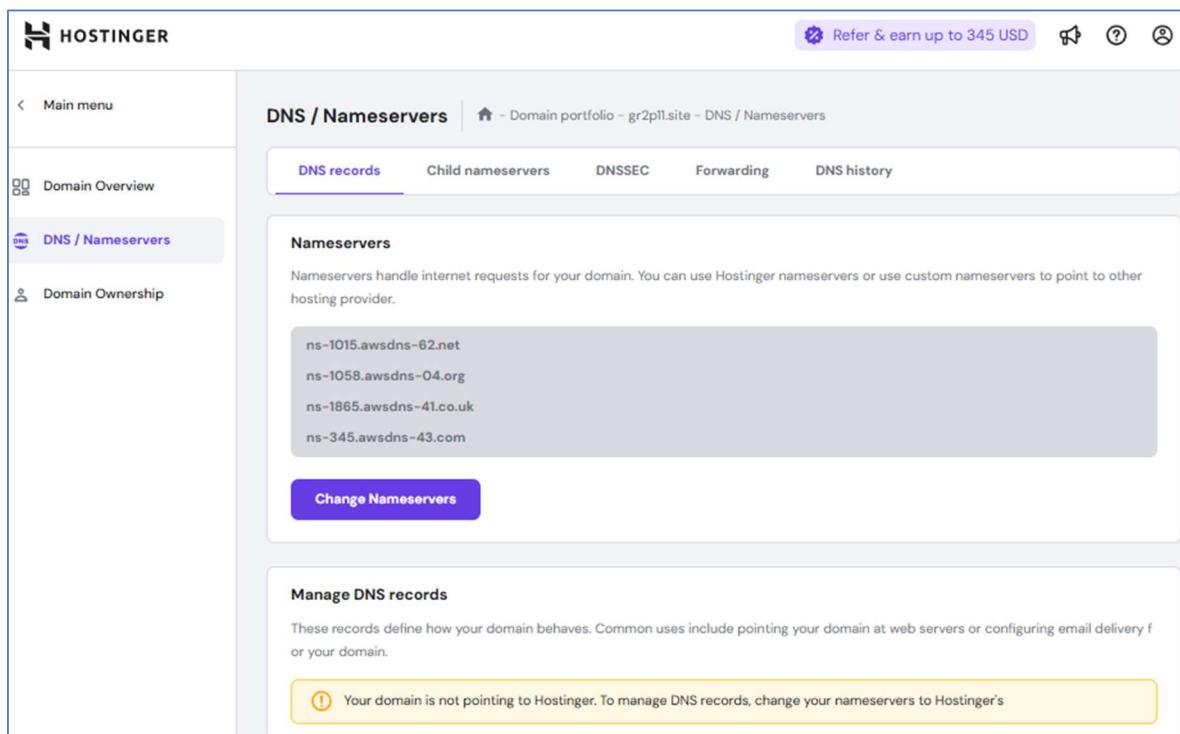
```
hohduong@hpHHD:~$ aws route53 create-hosted-zone \
> --name gr2p11.site \
> --caller-reference $(date +%s) \
> --hosted-zone-config Comment="Hosted zone for kOps cluster",PrivateZone=false
{
    "Location": "https://route53.amazonaws.com/2013-04-01/hostedzone/Z09203012SJ6UHVUK2FZD",
    "HostedZone": {
        "Id": "/hostedzone/Z09203012SJ6UHVUK2FZD",
        "Name": "gr2p11.site.",
        "CallerReference": "1751629999",
        "Config": {
            "Comment": "Hosted zone for kOps cluster",
            "PrivateZone": false
        },
        "ResourceRecordSetCount": 2
    },
    "ChangeInfo": {
        "Id": "/change/C100042330CSJ7N15II45",
        "Status": "PENDING",
        "SubmittedAt": "2025-07-04T11:53:21.525000+00:00"
    },
    "DelegationSet": {
        "NameServers": [
            "ns-1015.awsdns-62.net",
            "ns-345.awsdns-43.com",
            "ns-1865.awsdns-41.co.uk",
            "ns-1058.awsdns-04.org"
        ]
    }
}
```

Hình 27 – Tạo thành công Hosted Zone cho domain gr2p11.site trên Route 53

3.2.2.3. Liên kết domain với Route 53

Sau khi có danh sách NameServers từ AWS Route 53, nhóm đã đăng nhập vào tài khoản **Hostinger** và cập nhật phần cấu hình DNS để trỏ domain gr2p11.site về các NameServers đó.

Sau khoảng 5–15 phút, hệ thống DNS toàn cầu cập nhật và domain chính thức được điều khiển bởi Route 53.

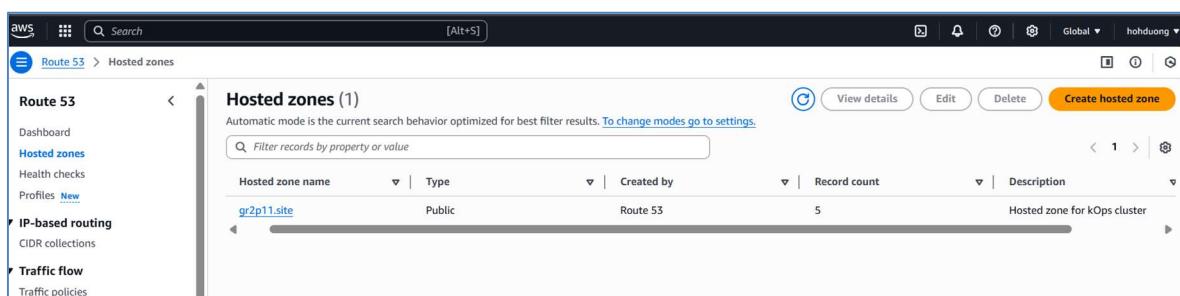


The screenshot shows the Hostinger DNS / Nameservers management interface. On the left sidebar, 'DNS / Nameservers' is selected. The main content area displays a list of four nameservers: ns-1015.awsdns-62.net, ns-1058.awsdns-04.org, ns-1865.awsdns-41.co.uk, and ns-345.awsdns-43.com. Below this list is a blue button labeled 'Change Nameservers'. Further down, there is a section titled 'Manage DNS records' with a note stating: 'These records define how your domain behaves. Common uses include pointing your domain at web servers or configuring email delivery for your domain.' A yellow warning box contains the message: 'Your domain is not pointing to Hostinger. To manage DNS records, change your nameservers to Hostinger's'.

Hình 28 – Giao diện website Hostinger sau khi thay đổi Nameservers

```
hohduong@hpHHD:~$ dig NS gr2p11.site +short
ns-1015.awsdns-62.net.
ns-1058.awsdns-04.org.
ns-1865.awsdns-41.co.uk.
ns-345.awsdns-43.com.
```

Hình 29 – Đã cập nhật lên hệ thống DNS toàn cầu



The screenshot shows the AWS Route 53 Hosted zones interface. The left sidebar shows 'Route 53' selected. The main area displays a table for 'Hosted zones (1)'. The table has columns: Hosted zone name, Type, Created by, Record count, and Description. One row is shown for 'gr2p11.site' with 'Public' type, 'Route 53' created by, 5 record count, and 'Hosted zone for kOps cluster' description. There are buttons for 'View details', 'Edit', 'Delete', and 'Create hosted zone'.

Hình 30 – Giao diện Route 53 trên AWS sau khi cập nhật tên miền

The screenshot shows the AWS Route 53 service interface. On the left, there's a sidebar with navigation links like 'Route 53', 'Hosted zones', 'New', 'Used routing', 'Collections', 'Traffic flow', 'Traffic policies', 'My records', and 'Domains'. The main area is titled 'Public gr2p11.site Info' and shows 'Hosted zone details'. Below this, there are tabs for 'Records (5)', 'DNSSEC signing', and 'Hosted zone tags (0)'. The 'Records (5)' tab is selected and shows a table of records:

	Record ...	Type	Routing type	Difference	Alias	Value/Route traffic to	TTL (s...)
<input type="checkbox"/>	gr2p11.site	NS	Simple	-	No	ns-1015.awsdns-62.net. ns-345.awsdns-43.com. ns-1865.awsdns-41.co.uk. ns-1058.awsdns-04.org.	172800
<input type="checkbox"/>	gr2p11.site	SOA	Simple	-	No	ns-1015.awsdns-62.net. aws...	900
<input type="checkbox"/>	api.k8s.gr...	A	Simple	-	No	47.129.218.254	60
<input type="checkbox"/>	api.internal...	A	Simple	-	No	172.20.88.232	60
<input type="checkbox"/>	kops-cont...	A	Simple	-	No	172.20.88.232	60

Hình 31 – Chi tiết các record của “gr2p11.site”

3.2.3. Tạo cụm Kubernetes bằng kOps

Sau khi hoàn tất việc cấu hình S3 và Route 53, nhóm thực hiện bước tiếp theo là khởi tạo và khai triển cụm Kubernetes trên AWS bằng công cụ **kOps**. Cụm này sẽ bao gồm 1 node điều khiển (control plane) và 2 node làm việc (worker nodes), được khai triển tại vùng **ap-southeast-1a (Singapore)**.

3.2.3.1. Tạo cụm Kubernetes với kOps

Nhóm đã sử dụng lệnh `kops create cluster` với các tham số cơ bản như sau:

```
kops create cluster \
--cloud aws \
--zones ap-southeast-1a \
--name k8s.gr2p11.site \
--state s3://kops-state-store-gr2p11 \
--dns-zone gr2p11.site \
```

```
--node-count 2 \
--node-size t3.medium \
--control-plane-size t3.medium \
--yes
```

Sau khi thực thi, cụm bắt đầu được khởi tạo trên AWS, bao gồm cả các tài nguyên EC2, VPC, Load Balancer, IAM, v.v.

```
nhoduong@hpHD:~$ kops create cluster \
> --cloud aws \
> --zones ap-southeast-1a \
> --name k8s.gr2pl1.site \
> --state $KOPS_STATE_STORE \
> --dns-zone gr2pl1.site \
> --node-count 2 \
> --node-size t3.medium \
> --master-size t3.medium \
> --yes

Flag --master-size has been deprecated, use --control-plane-size instead
I0704 12:27:30.879677 6686 new_cluster.go:1463] Cloud Provider ID: "aws"
I0704 12:27:33.357149 6686 subnets.go:224] Assigned CIDR 172.20.0.0/16 to subnet ap-southeast-1a
W0704 12:27:41.225149 6686 update_cluster.go:359] error checking control plane running version, assuming no k8s upgrade in progress: cannot load kubeconfig settings for "k8s.gr2pl1.site": context "k8s.gr2pl1.site" does not exist
I0704 12:28:28.362393 6686 executor.go:113] Tasks: 0 done / 184 total; 45 can run
I0704 12:28:28.894531 6686 keypair.go:226] Issuing new certificate: "etcd-clients-ca"
I0704 12:28:28.894647 6686 keypair.go:226] Issuing new certificate: "apiserver-aggregator-ca"
I0704 12:28:28.994136 6686 keypair.go:226] Issuing new certificate: "etcd-peers-ca-main"
I0704 12:28:29.011456 6686 keypair.go:226] Issuing new certificate: "etcd-manager-ca-main"
I0704 12:28:29.092085 6686 keypair.go:226] Issuing new certificate: "etcd-peers-ca-events"
W0704 12:28:29.092680 6686 vfs_keystorereader.go:163] CA private key was not found
I0704 12:28:29.282584 6686 keypair.go:226] Issuing new certificate: "service-account"
I0704 12:28:29.465951 6686 keypair.go:226] Issuing new certificate: "etcd-manager-ca-events"
W0704 12:28:30.763687 6686 vfs_keystorereader.go:163] CA private key was not found
I0704 12:28:30.838617 6686 keypair.go:226] Issuing new certificate: "kubernetes-ca"
I0704 12:28:32.690783 6686 executor.go:113] Tasks: 45 done / 184 total; 23 can run
I0704 12:28:35.604282 6686 executor.go:113] Tasks: 68 done / 184 total; 28 can run
I0704 12:28:37.136710 6686 executor.go:113] Tasks: 96 done / 184 total; 2 can run
I0704 12:28:40.578521 6686 executor.go:113] Tasks: 98 done / 184 total; 4 can run
I0704 12:28:41.035785 6686 executor.go:113] Tasks: 102 done / 184 total; 2 can run
I0704 12:28:41.258754 6686 executor.go:113] Tasks: 104 done / 184 total; 0 can run
I0704 12:28:49.575811 6686 dns.go:235] Pre-creating DNS records
I0704 12:28:49.911627 6686 update_cluster.go:412] Exporting kubeconfig for cluster
kOps has set your kubectl context to k8s.gr2pl1.site

Cluster is starting. It should be ready in a few minutes.

Suggestions:
* validate cluster: kops validate cluster --wait 10m
* list nodes: Kubectl get nodes --show-labels
* ssh to a control-plane node: ssh -i ~/.ssh/id_rsa ubuntu@api.k8s.gr2pl1.site
* the ubuntu user is specific to Ubuntu. If not using Ubuntu please use the appropriate user based on your OS.
* read about installing addons at: https://kops.sigs.k8s.io/addons.
```

Hình 32 – Đã tạo thành công Kubernetes cluster trên AWS bằng kOps

3.2.3.2. Kiểm tra cụm đã sẵn sàng

Nhóm đã sử dụng lệnh dưới đây để xác minh rằng cụm đã hoạt động ổn định:

```
kops validate cluster --wait 10m
```

Kết quả hiển thị trạng thái “READY” cho toàn bộ các node:

Your cluster k8s.gr2p11.site is ready

```
hohduong@hpHHD:~$ kops validate cluster --wait 10m
Using cluster from kubectl context: k8s.gr2p11.site

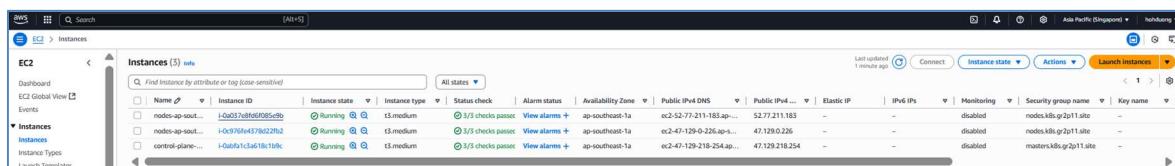
Validating cluster k8s.gr2p11.site

INSTANCE GROUPS
NAME          ROLE      MACHINETYPE  MIN  MAX  SUBNETS
control-plane-ap-southeast-1a  ControlPlane  t3.medium   1    1    ap-southeast-1a
nodes-ap-southeast-1a        Node       t3.medium   2    2    ap-southeast-1a

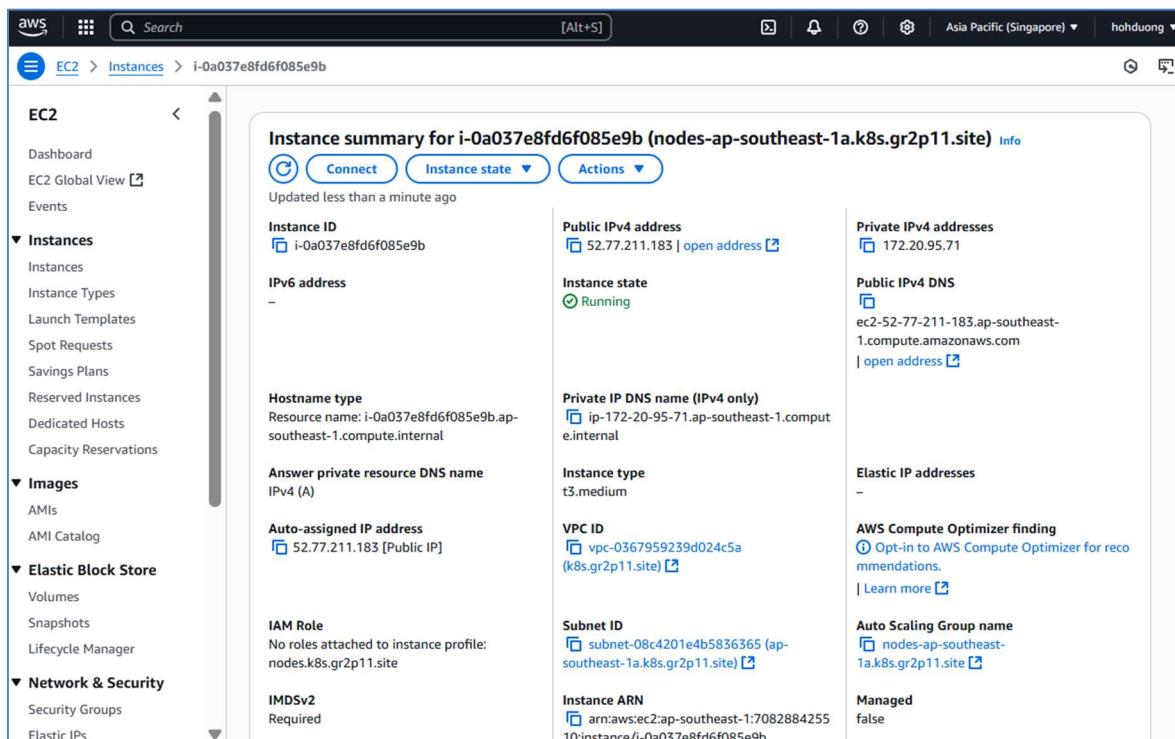
NODE STATUS
NAME          ROLE      READY
i-0a037e8fd6f085e9b  node     True
i-0abfa1c3a618c1b9c  control-plane  True
i-0c976fe4378d22fb2  node     True

Your cluster k8s.gr2p11.site is ready
```

Hình 33 – Kubernetes Cluster trên AWS đã hoạt động hoàn chỉnh và sẵn sàng khai triển ứng dụng



Hình 34 – Kiểm tra trên giao diện Instances EC2 của AWS



Hình 35 – Chi tiết thông tin của Control Plane trên AWS

Instance summary for i-0a037e8fd6f085e9b (nodes-ap-southeast-1a.k8s.gr2p11.site) [Info](#)

Instance ID i-0a037e8fd6f085e9b	Public IPv4 address 52.77.211.183 open address	Private IPv4 addresses 172.20.95.71
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-77-211-183.ap-southeast-1.compute.amazonaws.com open address
Hostname type Resource name: i-0a037e8fd6f085e9b.ap-southeast-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-20-95-71.ap-southeast-1.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t3.medium	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 52.77.211.183 [Public IP]	VPC ID vpc-0367959239d024c5a (k8s.gr2p11.site)	Auto Scaling Group name nodes-ap-southeast-1a.k8s.gr2p11.site
IAM Role No roles attached to instance profile: nodes.k8s.gr2p11.site	Subnet ID subnet-08c4201e4b5836365 (ap-southeast-1a.k8s.gr2p11.site)	Managed false
IMDSv2 Required	Instance ARN arn:aws:ec2:ap-southeast-1:7082884255:instance/i-0a037e8fd6f085e9b	

Hình 36 – Chi tiết của Worker Node 1 trên AWS

Instance summary for i-0abfa1c3a618c1b9c (control-plane-ap-southeast-1a.masters.k8s.gr2p11.site) [Info](#)

Instance ID i-0abfa1c3a618c1b9c	Public IPv4 address 47.129.218.254 open address	Private IPv4 addresses 172.20.88.232
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-47-129-218-254.ap-southeast-1.compute.amazonaws.com open address
Hostname type Resource name: i-0abfa1c3a618c1b9c.ap-southeast-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-20-88-232.ap-southeast-1.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t3.medium	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 47.129.218.254 [Public IP]	VPC ID vpc-0367959239d024c5a (k8s.gr2p11.site)	Auto Scaling Group name control-plane-ap-southeast-1a.masters.k8s.gr2p11.site
IAM Role masters.k8s.gr2p11.site	Subnet ID subnet-08c4201e4b5836365 (ap-southeast-1a.k8s.gr2p11.site)	Managed false
IMDSv2 Required	Instance ARN arn:aws:ec2:ap-southeast-1:7082884255:instance/i-0abfa1c3a618c1b9c	

Hình 37 – Chi tiết của Worker Node 2 trên AWS

3.2.3.3. Tệp kubeconfig và truy cập kubectl

Sau khi cụm được tạo thành công, nhóm sử dụng lệnh sau để xuất file cấu hình kubeconfig:

```
kops export kubecfg --admin --name k8s.gr2p11.site
```

Nhờ đó, lệnh kubectl có thể sử dụng để giao tiếp trực tiếp với cụm Kubernetes.

```
hohduong@hpHHD:~$ kubectl version --client
Client Version: v1.29.3
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

Hình 38 – Cài đặt kubectl thành công và đang dùng phiên bản mới nhất

```
hohduong@hpHHD:~$ kops version
Client version: 1.32.0 (git-v1.32.0)
```

Hình 39 – Cài bản v1.32.0 và chạy kops version thành công

3.3. Khai triển dịch vụ WordPress và MariaDB

Sau khi cụm Kubernetes đã sẵn sàng, nhóm tiến hành khai triển hai thành phần quan trọng của ứng dụng là **WordPress** (nền tảng web/blog) và **MariaDB** (cơ sở dữ liệu). Cả hai dịch vụ này được khai triển độc lập trong namespace default, kết nối với nhau thông qua các Service nội bộ.

3.3.1. Khai triển MariaDB

MariaDB được khai triển thông qua một file YAML khai báo cả Deployment, PersistentVolumeClaim, và Service. Một số thông tin chính:

- **Image:** mariadb:10.6
- **Tên database:** wordpress
- **User/password:** wpuser / wppass
- **PVC:** Dung lượng 20Gi, gắn vào /var/lib/mysql

Câu lệnh áp dụng cấu hình:

```
kubectl apply -f mariadb-deployment.yaml
```

```
hohduong@hpHHD:~$ mkdir ~/k8s-wordpress
hohduong@hpHHD:~$ cd ~/k8s-wordpress
hohduong@hpHHD:~/k8s-wordpress$ nano mariadb-deployment.yaml
hohduong@hpHHD:~/k8s-wordpress$ kubectl apply -f mariadb-deployment.yaml
persistentvolumeclaim/mariadb-pv-claim created
deployment.apps/mariadb created
service/mariadb created
```

Hình 40 – MariaDB đã được deploy thành công vào Kubernetes cluster

```
hohduong@hpHHD:~/k8s-wordpress$ kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
mariadb-77cc5dbf7c-c4rzz   1/1     Running   0          6m36s
```

Hình 41 – Kiểm tra pod đã hoạt động

3.3.2. Khai triển WordPress

WordPress cũng được khai triển với cách tương tự, có cấu hình kết nối đến MariaDB thông qua biến môi trường:

- **Image:** wordpress:6.1.1-apache
- **Liên kết DB:** thông qua WORDPRESS_DB_HOST = mariadb
- **PVC:** Dung lượng 20Gi, gắn vào /var/www/html

Câu lệnh áp dụng cấu hình:

```
kubectl apply -f wordpress-deployment.yaml
```

Một Service kiểu LoadBalancer được tạo ra để cung cấp ứng dụng ra internet thông qua địa chỉ ELB (Elastic Load Balancer).

```
hohduong@hpHHD:~/k8s-wordpress$ nano wordpress-deployment.yaml
hohduong@hpHHD:~/k8s-wordpress$ kubectl apply -f wordpress-deployment.yaml
persistentvolumeclaim/wordpress-pv-claim created
deployment.apps/wordpress created
service/wordpress created
```

Hình 42 – Đã khai triển WordPress frontend thành công trên Kubernetes cluster

```
hoeduong@hpHHD:~/k8s-wordpress$ kubectl get svc
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP
PORT(S)   AGE
kubernetes  ClusterIP  100.64.0.1    <none>
           443/TCP   36m
mariadb    ClusterIP  100.64.243.110  <none>
           3306/TCP  15m
wordpress   LoadBalancer 100.65.203.33  a14e53bb8754f424c854ea47b9edb8cc-745214291.ap-southeast-1.elb.amazonaws.com  80:32096/TCP  5m51s
```

Hình 43 – WordPress đã được public thành công ra internet thông qua AWS Load Balancer

3.3.3. Truy cập website từ domain tùy chỉnh

Sau khi có địa chỉ ELB từ AWS, nhóm tạo bản ghi CNAME trên Route 53 để ánh xạ subdomain blog.gr2p11.site trỏ đến địa chỉ ELB.

aws route53 change-resource-record-sets --hosted-zone-id <ZONE_ID> ...

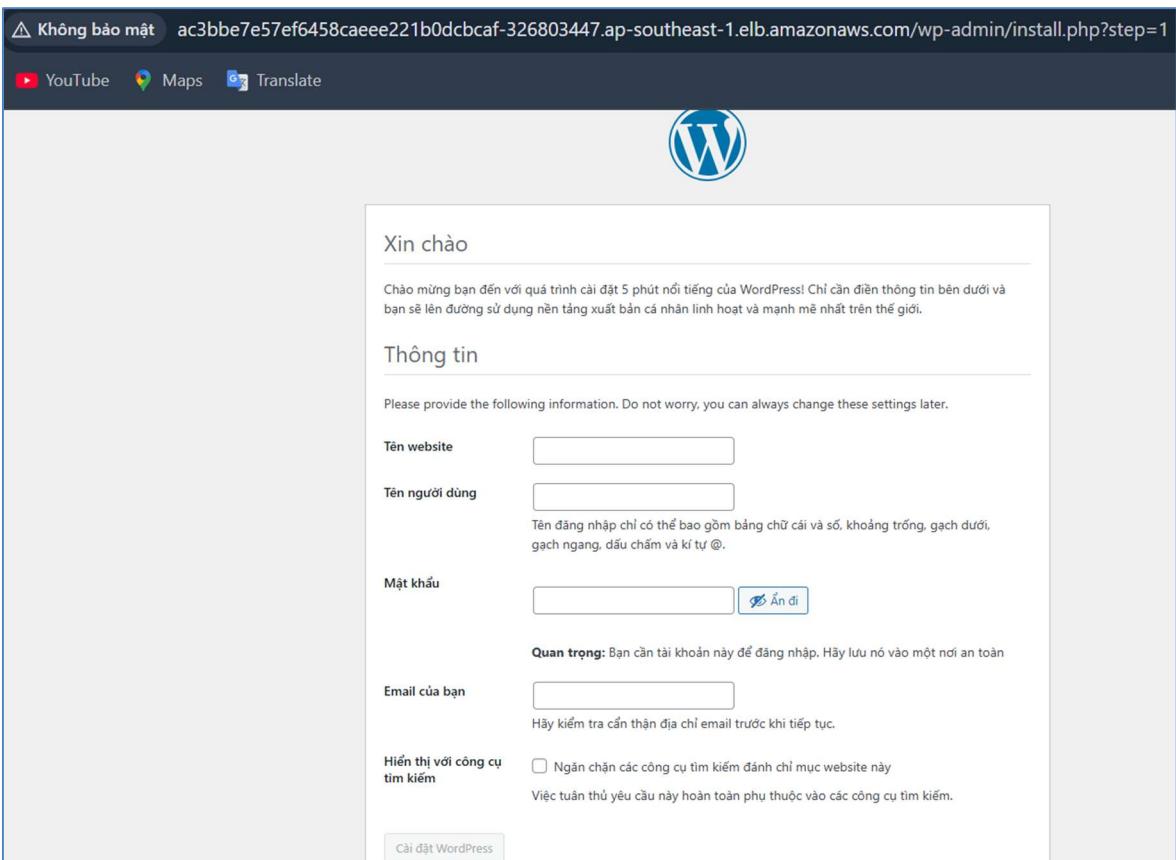
Sau khi DNS được cập nhật, người dùng có thể truy cập WordPress dễ dàng tại địa chỉ: <http://blog.gr2p11.site>

```
hoeduong@hpHHD:~/k8s-wordpress$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
mariadb-fd5788ff9-fpq98   1/1     Running   0          5m14s
wordpress-69c74d7f78-4fwtg   1/1     Running   0          5m2s
```

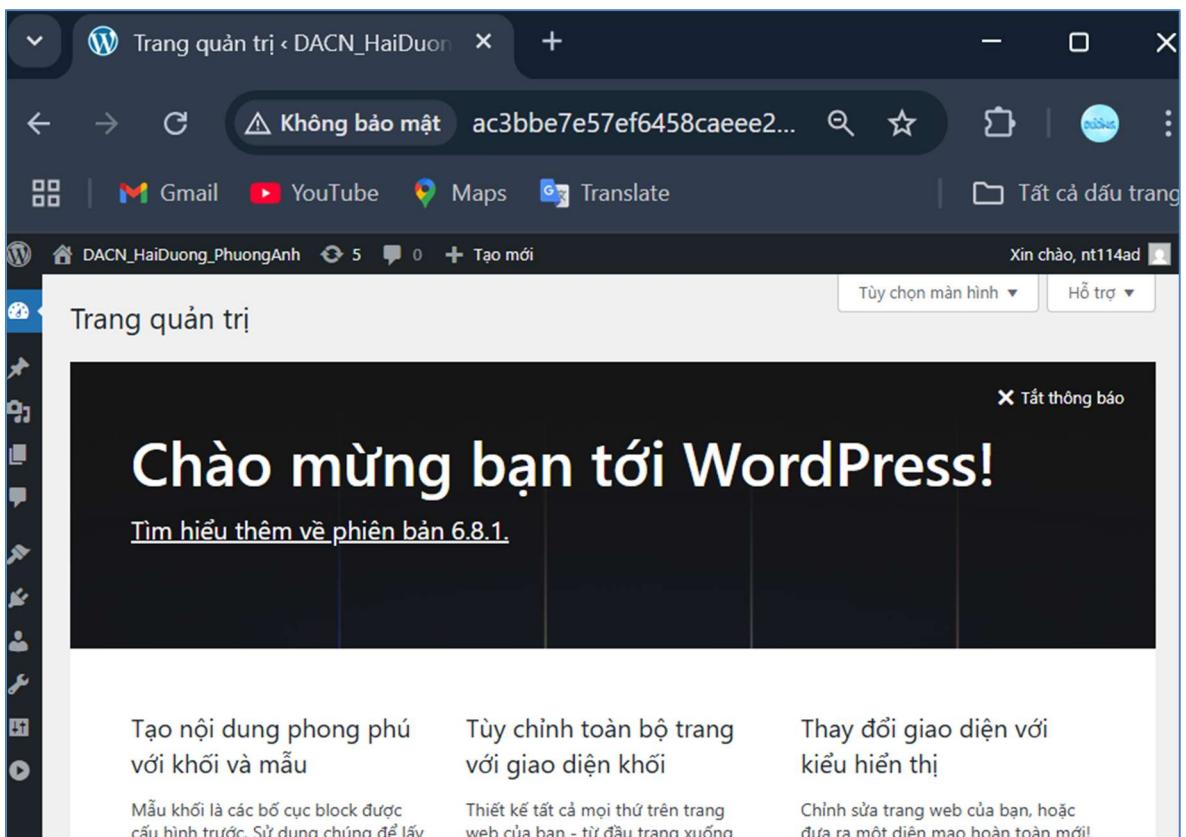
Hình 44 – Cả 2 ứng dụng đều đã Running

```
hoeduong@hpHHD:~/k8s-wordpress$ kubectl get svc
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP
PORT(S)   AGE
kubernetes  ClusterIP  100.64.0.1    <none>
           443/TCP   68m
mariadb    ClusterIP  100.64.205.143  <none>
           3306/TCP  8m31s
wordpress   LoadBalancer 100.64.103.214  ac3bbe7e57ef6458caeee221b0dcbaaf-326803447.ap-southeast-1.elb.amazonaws.com  80:31951/TCP  8m19s
```

Hình 45 – Địa chỉ Load Balancer mà AWS đã cấp để truy cập trình duyệt



Hình 46 – Deploy thành công WordPress lên Kubernetes chạy trên AWS



Hình 47 – Vào được dashboard “WordPress Admin Panel” cho thấy hệ thống đã hoạt động hoàn chỉnh

3.4. Khai triển hệ thống giám sát với Prometheus và Grafana

Để hỗ trợ việc theo dõi tình trạng hoạt động của các pod, cluster, và tài nguyên trong hệ thống Kubernetes, nhóm đã khai triển thêm hai công cụ giám sát phổ biến là **Prometheus** và **Grafana**. Hai công cụ này được khai triển trong namespace riêng tên là monitoring để dễ quản lý và phân tách chức năng với các dịch vụ khác như WordPress hay MariaDB.

3.4.1. Tạo namespace và khai triển Prometheus

Prometheus được khai triển dưới dạng một Deployment cùng với một Service kiểu ClusterIP. File cấu hình sử dụng ConfigMap để chỉ định các targets sẽ được theo dõi, bao gồm các node, pod, và metrics từ chính Kubernetes.

Các bước thực hiện chính:

```
kubectl create namespace monitoring
```

```
kubectl apply -f prometheus-deployment.yaml
```

Thành phần khai triển:

- Deployment: tạo một Pod chạy Prometheus
- ConfigMap: chứa cấu hình prometheus.yml
- Service: kiểu ClusterIP, dùng để Grafana kết nối

```
hohduong@hpHHD:~/k8s-monitoring$ nano prometheus-deployment.yaml
hohduong@hpHHD:~/k8s-monitoring$ kubectl apply -f prometheus-deployment.yaml
configmap/prometheus-config created
deployment.apps/prometheus created
service/prometheus created
```

Hình 48 – Khai triển thành công Prometheus vào namespace monitoring

3.4.2. Khai triển Grafana

Grafana được khai triển tương tự như Prometheus nhưng sử dụng Service kiểu LoadBalancer để có thể truy cập qua internet. Grafana sử dụng ConfigMap để chỉ định kết nối đến Prometheus như một nguồn dữ liệu (Data Source).

Các bước thực hiện chính:

```
kubectl apply -f grafana-deployment.yaml
```

Thành phần khai triển:

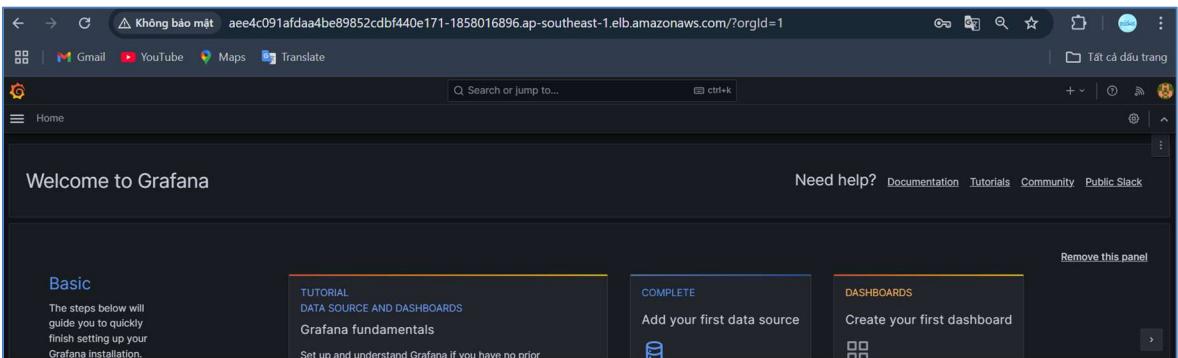
- Deployment: một Pod chạy container grafana/grafana
- ConfigMap: chỉ định Prometheus là nguồn dữ liệu
- Service: kiểu LoadBalancer, mở cổng 80 để truy cập

```

hohduong@hpHHD:~/k8s-monitoring$ nano grafana-deployment.yaml
hohduong@hpHHD:~/k8s-monitoring$ kubectl apply -f grafana-deployment.yaml
configmap/grafana-datasource created
deployment.apps/grafana created
service/grafana created
hohduong@hpHHD:~/k8s-monitoring$ kubectl get svc -n monitoring
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP
grafana   LoadBalancer   100.64.44.32   aee4c091afdaa4be89852cdbf440e171-1858016896.ap-southeast-1.elb.amazonaws.com   80:32190/TCP   5s
prometheus   ClusterIP    100.64.24.9    <none>
                           9090/TCP    69s

```

Hình 49 – Grafana đã được khai triển thành công với LoadBalancer – Có thể truy cập từ trình duyệt



Hình 50 – Giao diện Grafana

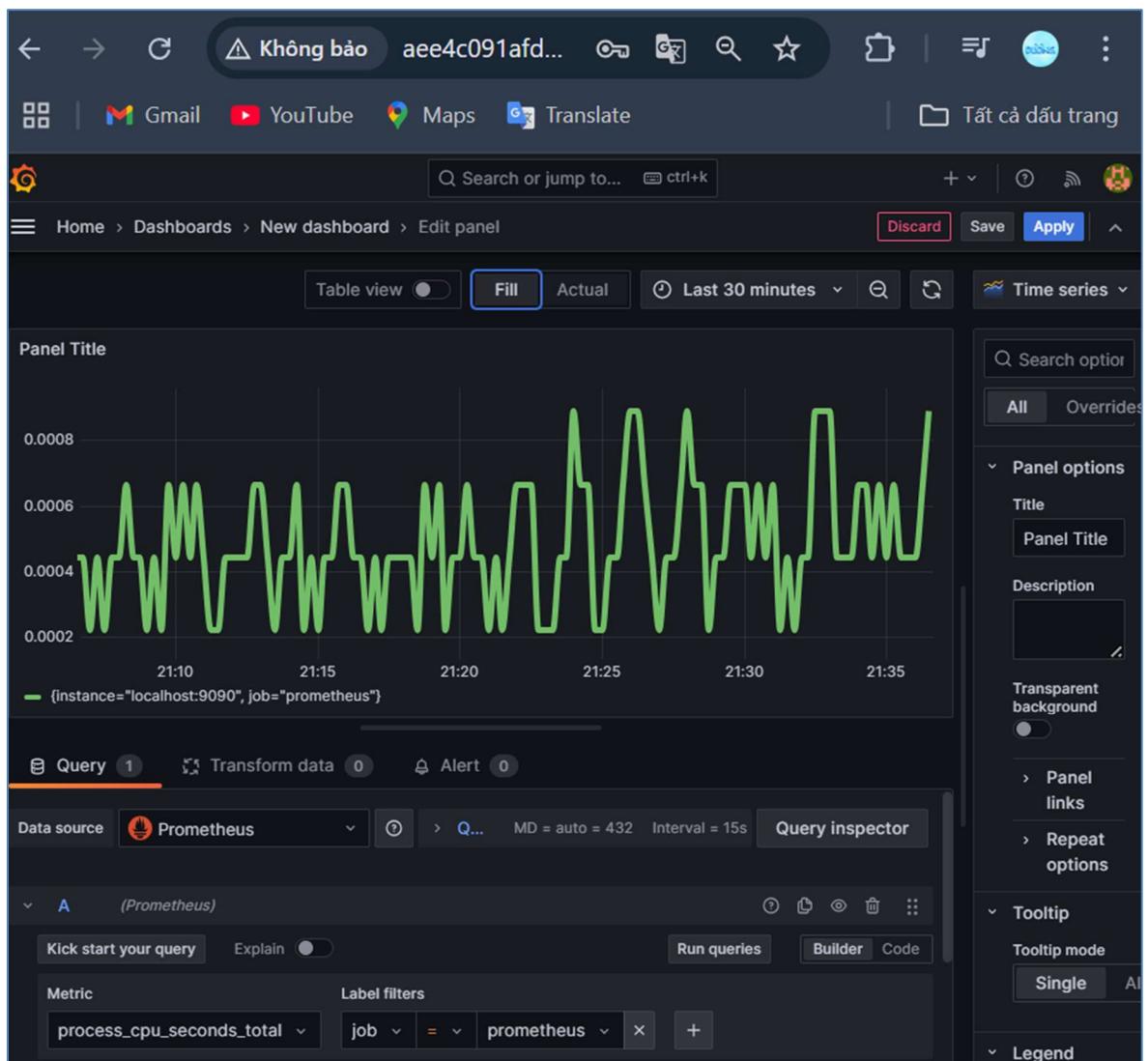
3.4.3. Tạo dashboard giám sát cơ bản

Sau khi hoàn tất khai triển Prometheus và Grafana, nhóm đã truy cập vào giao diện web của Grafana thông qua địa chỉ LoadBalancer do Kubernetes cung cấp. Tại đây, nhóm tiến hành tạo một dashboard giám sát cơ bản để theo dõi tình trạng hoạt động của các pod trong namespace default.

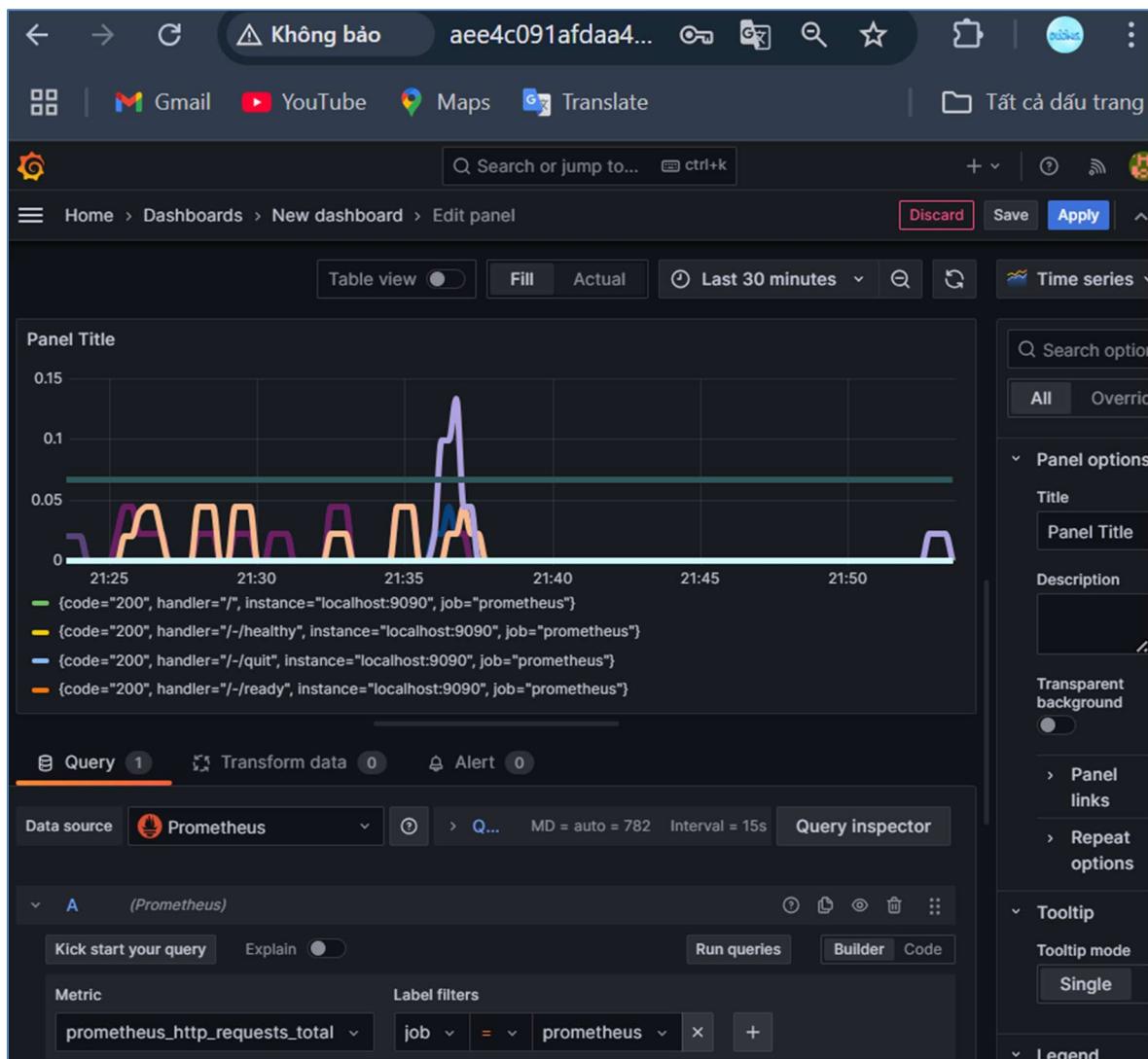
Thay vì sử dụng truy vấn PromQL thủ công, nhóm sử dụng công cụ **Builder** có sẵn trong giao diện tạo biểu đồ của Grafana. Đây là một công cụ trực quan giúp người dùng lựa chọn các trường dữ liệu và điều kiện lọc mà không cần phải viết mã.

Mục đích của biểu đồ là cung cấp cái nhìn tổng quan về số lượng pod đang ở các trạng thái khác nhau.

Grafana đã vẽ biểu đồ theo metric process_cpu_seconds_total, chứng tỏ hệ thống Prometheus ↔ Grafana hoạt động đúng:



Hình 51 – Biểu đồ giám sát mức sử dụng CPU của Prometheus



Hình 52 – Biểu đồ giám sát số lượng HTTP request đến Prometheus

C. KẾT LUẬN

1. KẾT QUẢ ĐẠT ĐƯỢC

STT	Nội dung khai triển	Kết quả đạt được	Mức độ hoàn thành
1	Tạo tài khoản AWS và thiết lập công cụ CLI	Đã cấu hình AWS CLI trên môi trường WSL2, kết nối thành công	100%
2	Tạo S3 bucket và cấu hình Route53 với tên miền từ Hostinger	Sử dụng tên miền gr2p11.site để khai triển cụm Kubernetes thành công	100%
3	Cài đặt và cấu hình kOps, kubectl, Prometheus, Grafana	Khai triển thành công cụ Kubernetes với các công cụ giám sát	100%
4	Khởi tạo và vận hành cụm Kubernetes trên AWS	Tạo thành công 1 control-plane node và 2 worker node	100%
5	Khai triển dịch vụ WordPress và MariaDB trên cụm	Cấu hình đầy đủ, truy cập được qua tên miền phụ blog.gr2p11.site	100%
6	Khai triển Prometheus và Grafana để giám sát cụm	Truy cập được Grafana qua internet, tạo dashboard và biểu đồ PromQL cơ bản	100%
7	Giao tiếp giữa các node, bảo đảm tính ổn định	Các pod hoạt động ổn định, giám sát thành công dữ liệu hệ thống	100%
8	Khai triển thêm HA/Scale-out	Đã thử nghiệm scale-out WordPress nhưng gặp giới hạn về volume	60%

Bảng 4 – Tổng kết các nội dung đã khai triển thành công

2. NHỮNG HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN

2.1. Khó khăn và hạn chế khi thực hiện

Quá trình thiết lập cụm Kubernetes bằng kOps yêu cầu nhiều bước cấu hình như IAM, Route53, S3 và đòi hỏi sự chính xác cao. Những sai sót nhỏ cũng có thể dẫn đến lỗi khó phát hiện.

Trong quá trình mở rộng (scale-out), nhóm gặp vấn đề về PersistentVolumeClaim do hạn chế của chế độ truy cập RWO khiến không thể mount cùng lúc lên nhiều pod WordPress.

Trong quá trình mở rộng (scale-out), nhóm gặp vấn đề về PersistentVolumeClaim do hạn chế của chế độ truy cập RWO khiến không thể mount cùng lúc lên nhiều pod WordPress.

Hệ thống yêu cầu tài nguyên điện toán và chi phí cloud thực tế (AWS), nên việc giám sát sát sao thời gian hoạt động và chi phí phát sinh là cần thiết (nhóm đã tốn 23 USD để duy trì hệ thống dịch vụ AWS).

2.2. Đề xuất hướng phát triển

Thực hiện High Availability thực sự bằng cách khai triển cấu hình multi-AZ, kết hợp với auto scaling và sử dụng EFS (Elastic File System) để chia sẻ volume giữa các pod WordPress.

Tích hợp công cụ Alertmanager để cấu hình cảnh báo (notification) gửi qua email hoặc các ứng dụng chat (Slack, Telegram) khi hệ thống gặp sự cố.

Tự động hóa toàn bộ quy trình khai triển bằng các công cụ như Terraform, Helm, GitHub Actions để dễ dàng tái sử dụng và mở rộng hệ thống.

Bảo mật hệ thống tốt hơn bằng cách cấu hình SSL với Let's Encrypt và kiểm soát truy cập thông qua RBAC, network policy.

TÀI LIỆU THAM KHẢO

- [1] TechWorld with Nana, “Learn Prometheus in 10 minutes 🎥,” *YouTube*, Apr. 10, 2022. [Video file]. Available: <https://www.youtube.com/watch?v=SWShhJ4138I>
- [2] Just me and Opensource, “How to deploy Prometheus and Grafana on Kubernetes,” *YouTube*, Oct. 2, 2022. [Video file]. Available: <https://youtu.be/zZf4YH4WiZo?si=dmp9EhLFiGl8mszJ>
- [3] Tuan Anh Dinh, “Khai triển kOps Kubernetes trên AWS (Kubernetes Production Cluster),” *YouTube*, Jan. 17, 2023. [Video file]. Available: <https://youtu.be/u0JyzUGzvJA?si=3uFQM1RK8LMUIURN>
- [4] Amazon Web Services, “What is Amazon EC2?”, AWS Documentation, 2023. [Online]. Available: <https://docs.aws.amazon.com/ec2/>
- [5] Kubernetes Project, “Production-Grade Container Orchestration”, Kubernetes.io, 2024. [Online]. Available: <https://kubernetes.io/>
- [6] kOps Authors, “Kubernetes Operations (kOps)”, GitHub Repository, 2024. [Online]. Available: <https://github.com/kubernetes/kops>
- [7] Prometheus Authors, “Prometheus Monitoring System”, 2024. [Online]. Available: <https://prometheus.io/>
- [8] Grafana Labs, “Grafana: The open observability platform”, 2024. [Online]. Available: <https://grafana.com/>
- [9] WordPress Foundation, “WordPress: Blog Tool, Publishing Platform, and CMS”, 2024. [Online]. Available: <https://wordpress.org/>
- [10] MariaDB Foundation, “MariaDB Server Documentation”, 2024. [Online]. Available: <https://mariadb.org/>

- [11] AWS CLI Team, “AWS Command Line Interface User Guide”, Amazon, 2024. [Online]. Available: <https://docs.aws.amazon.com/cli/>
- [12] Microsoft, “Install WSL on Windows 11”, Microsoft Docs, 2024. [Online]. Available: <https://learn.microsoft.com/windows/wsl/>
- [13] Hostinger, “How to Manage DNS Records on Hostinger”, Hostinger Tutorials, 2024. [Online]. Available: <https://www.hostinger.com/tutorials/>

HÉT./.