

TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT  
VIỆN ĐÀO TẠO CÔNG NGHỆ THÔNG TIN, CHUYÊN ĐỒI SÓ



ĐÒ ÁN MÔN HỌC  
ĐIỆN TOÁN ĐÁM MÂY

TÌM HIỂU VỀ ĐIỆN TOÁN ĐÁM MÂY VÀ  
TRIỂN KHAI WEBSITE BÁN GAMING GEAR  
TRÊN AMAZON WEB SERVICES

SVTH: Trần Bùi Quang Huy 2124801030189

Võ Văn An 2124801030171

GVHD: ThS. Hồ Ngọc Trung Kiên

BÌNH DƯƠNG - 12/2024

TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT  
VIỆN ĐÀO TẠO CÔNG NGHỆ THÔNG TIN, CHUYỂN ĐỔI SỐ



ĐỒ ÁN MÔN HỌC  
ĐIỆN TOÁN ĐÁM MÂY

TÌM HIỂU VỀ ĐIỆN TOÁN ĐÁM MÂY VÀ  
TRIỂN KHAI WEBSITE BÁN GAMING GEAR  
TRÊN AMAZON WEB SERVICES

SVTH: Trần Bùi Quang Huy 2124801030189

Võ Văn An 2124801030171

GVHD: ThS. Hồ Ngọc Trung Kiên

BÌNH DƯƠNG - 12/2024

## **NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**Họ và tên giảng viên: ThS.Hồ Ngọc Trung Kiên**

**Đề tài: Tìm hiểu về điện toán đám mây và triển khai Website bán Gaming Gear trên Amazon Web Services**

**Nội dung nhận xét:**

.....  
.....  
.....  
.....  
.....  
.....  
.....

**Điểm:**

Bằng số:

Bằng chữ:

Bình Dương, ngày 28 tháng 11 năm 2024

**GIẢNG VIÊN**

## MỤC LỤC

<b>MỤC LỤC .....</b>	<b>ii</b>
<b>DANH MỤC HÌNH .....</b>	<b>v</b>
<b>DANH MỤC BẢNG .....</b>	<b>viii</b>
<b>DANH MỤC VIẾT TẮT .....</b>	<b>ix</b>
<b>LỜI CẢM ƠN .....</b>	<b>x</b>
<b>LỜI MỞ ĐẦU .....</b>	<b>xi</b>
<b>CHƯƠNG 1.TỔNG QUAN VỀ ĐIỆN TOÁN ĐÁM MÂY .....</b>	<b>1</b>
1.1.  Giới thiệu chung.....	1
1.1.1. <i>Lịch sử ra đời</i> .....	1
1.1.2. <i>Quá trình phát triển</i> .....	2
1.1.3. <i>Khái niệm</i> .....	3
1.1.4. <i>Ưu điểm và nhược điểm</i> .....	4
1.1.5. <i>Tầm quan trọng đối với doanh nghiệp</i> .....	5
1.2.  Các mô hình triển khai và cung cấp dịch vụ.....	5
1.2.1. <i>Mô hình triển khai</i> .....	5
1.2.2. <i>Mô hình cung cấp dịch vụ</i> .....	6
1.3.  Liên điều hành trong điện toán đám mây .....	8
1.3.1. <i>Khái niệm</i> .....	8
1.3.2. <i>Vai trò</i> .....	8
1.3.3. <i>Thành phần và công cụ hỗ trợ</i> .....	9
1.3.4. <i>Chuẩn hóa liên điều hành</i> .....	9
1.3.5. <i>Thách thức</i> .....	10
1.4.  Đặc trưng kỹ thuật.....	11
1.5.  Các ứng dụng của điện toán đám mây .....	12
1.5.1. <i>E-Learning</i> .....	12
1.5.2. <i>ERP</i> .....	12
1.5.3. <i>E-Government</i> .....	13
1.5.4. <i>Hệ thống lưu trữ Amazon</i> .....	13
1.6.  Mô hình truyền thông.....	13
1.6.1. <i>Kết nối tĩnh</i> .....	14
1.6.2. <i>Mạng chuyển mạch</i> .....	16
1.7.  Kiến trúc song song .....	19
1.8.  Kiến trúc phân tán .....	21
1.9.  Cấu trúc điện toán đám mây .....	22
1.9.1. <i>Kiến trúc tương tác IaaS</i> .....	22
1.9.2. <i>Kiến trúc tương tác PaaS</i> .....	24
1.9.3. <i>Kiến trúc tương tác SaaS</i> .....	25
1.10.  Hạ tầng đám mây và ảo hóa .....	26
1.10.1. <i>Hạ tầng đám mây</i> .....	26

1.10.2. <i>Ảo hóa</i> .....	26
<b>CHƯƠNG 2.LUƯ TRỮ VÀ BẢO MẬT TRONG ĐIỆN TOÁN ĐÁM MÂY .....</b>	<b>30</b>
2.1. Tổng quan giải pháp mạng truy nhập .....	30
2.2. Các công nghệ lưu trữ và mô hình lưu trữ .....	31
2.2.1. <i>Mô hình lưu trữ</i> .....	31
2.2.2. <i>Công nghệ lưu trữ</i> .....	34
2.3. Hệ thống lưu trữ File.....	37
2.3.1. <i>Nguyên tắc truy nhập</i> .....	37
2.3.2. <i>Truy nhập File</i> .....	37
2.4. Hệ thống lưu trữ phân tán và tính nhất quán của bộ nhớ NFS và AFS .....	38
2.4.1. <i>Hệ thống lưu trữ phân tán</i> .....	38
2.4.2. <i>Bộ nhớ NFS</i> .....	39
2.4.3. <i>Bộ nhớ AFS</i> .....	42
2.5. Hệ thống lưu trữ HDFS & GFS .....	45
2.5.1. <i>Hệ thống tệp phân tán Hadoop</i> .....	45
2.5.2. <i>Hệ thống tệp Google</i> .....	52
2.6. Cơ sở dữ liệu NoSQL .....	54
2.6.1. <i>Khái niệm</i> .....	54
2.6.2. <i>Các dạng lưu trữ</i> .....	54
2.6.3. <i>Ưu điểm và nhược điểm</i> .....	57
2.7. Dữ liệu lớn và điện toán đám mây .....	57
2.7.1. <i>Dữ liệu lớn</i> .....	57
2.7.2. <i>Mối quan hệ</i> .....	58
2.7.3. <i>Công cụ</i> .....	59
2.8. Khái quát nguy cơ và tác động tới điện toán đám mây.....	60
2.8.1. <i>Nguyên lý bảo mật chung</i> .....	60
2.8.2. <i>Thỏa thuận ở mức dịch vụ SLA</i> .....	61
2.8.3. <i>Đe dọa và tính rủi ro</i> .....	62
2.8.4. <i>Mối đe dọa với điện toán đám mây</i> .....	62
2.9. Mã hóa dữ liệu đám mây.....	64
2.9.1. <i>Bảo mật dữ liệu</i> .....	64
2.9.2. <i>Kỹ thuật mã hóa trong đám mây</i> .....	65
2.9.3. <i>Bảo mật với cơ sở hạ tầng</i> .....	67
2.10. Bảo mật hệ thống điều hành.....	69
2.10.1. <i>Bảo mật máy ảo</i> .....	69
2.10.2. <i>Mối đe dọa bảo mật</i> .....	70
2.11. Bảo mật cho giải pháp ảo hóa .....	71
2.11.1. <i>Mối đe dọa</i> .....	72
2.11.2. <i>Khuyến nghị</i> .....	72
<b>CHƯƠNG 3.TRIỂN KHAI WEBSITE BÁN GAMING GEAR TRÊN AWS .....</b>	<b>74</b>
3.1. Amazon Web Service .....	74

3.1.1. <i>Lịch sử ra đời</i> .....	74
3.1.2. <i>Khái niệm</i> .....	74
3.1.3. <i>Dịch vụ</i> .....	75
3.2. Microsoft Azure .....	76
3.2.1. <i>Lịch sử ra đời</i> .....	76
3.2.2. <i>Khái niệm</i> .....	77
3.2.3. <i>Dịch vụ</i> .....	77
3.3. Google App Engine.....	79
3.3.1. <i>Lịch sử ra đời</i> .....	79
3.3.2. <i>Khái niệm</i> .....	80
3.3.3. <i>Dịch vụ</i> .....	80
3.4. Phân tích và thiết kế hệ thống .....	81
3.4.1. <i>Sơ đồ Use case</i> .....	81
3.4.2. <i>Sơ đồ lớp</i> .....	82
3.4.3. <i>Thiết kế cơ sở dữ liệu</i> .....	83
3.5. Triển khai và thực nghiệm .....	89
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>104</b>

## **DANH MỤC HÌNH**

Hình 1.1: Lịch sử ra đời Điện toán đám mây .....	1
Hình 1.2: Quá trình phát triển Điện toán đám mây .....	3
Hình 1.3: Đám mây lai (Hybrid Cloud).....	6
Hình 1.4: Mô hình cung cấp dịch vụ .....	7
Hình 1.5: Sơ đồ mạng bus .....	14
Hình 1.6: Sơ đồ mạng siêu khối .....	15
Hình 1.7: Sơ đồ mạng hình lưới 2D .....	16
Hình 1.8: Bộ chuyển mạch Crossbar .....	17
Hình 1.9: Sơ đồ mạng Omega .....	18
Hình 1.10: Kiến trúc song song .....	19
Hình 1.11: Mô hình máy khách/máy chủ .....	22
Hình 1.12: Mô hình Peer-to-Peer .....	22
Hình 1.13: Kiến trúc tương tác IaaS .....	23
Hình 1.14: Kiến trúc tương tác PaaS .....	24
Hình 1.15: Kiến trúc tương tác SaaS .....	25
Hình 2.1: Mô hình lưu trữ ô .....	32
Hình 2.2: Mô hình lưu trữ nhật ký.....	33
Hình 2.3: Hệ thống phân cấp lưu trữ .....	36
Hình 2.4: Ví dụ truy nhập File.....	38
Hình 2.5: Kiến trúc NFS .....	40
Hình 2.6: Kiến trúc AFS .....	43
Hình 2.7: Kiến trúc HDFS .....	45
Hình 2.8: Cài đặt thành công Hadoop .....	52
Hình 2.9: Kiến trúc GFS .....	53
Hình 2.10: Cơ sở dữ liệu dạng cột.....	55
Hình 2.11: Cơ sở dữ liệu dạng đồ thị .....	55
Hình 2.12: Cơ sở dữ liệu dạng tài liệu.....	56
Hình 2.13: Cơ sở dữ liệu dạng cặp khóa - giá trị .....	56
Hình 2.14: Quy trình thỏa thuận mức dịch vụ.....	61
Hình 2.15: Mã hóa dữ liệu .....	65

Hình 2.16: Mã hóa dựa trên thuộc tính chính sách khóa.....	66
Hình 2.17: Mã hóa dựa trên thuộc tính chính sách mã hóa.....	67
Hình 2.18: Mã hóa đồng hình hoàn toàn .....	68
Hình 2.19: Mã hóa có thể tìm kiếm.....	68
Hình 2.20: Kiến trúc dịch vụ bảo mật ảo.....	70
Hình 3.1: Kiến trúc cơ bản của AWS .....	75
Hình 3.2: Kiến trúc cơ bản của Microsoft Azure .....	78
Hình 3.3: Sơ đồ Use case tổng quát.....	81
Hình 3.4: Sơ đồ lớp .....	82
Hình 3.5: Lược đồ quan hệ thực thể .....	83
Hình 3.6: Trang đăng ký AWS:.....	89
Hình 3.7: Trang tài khoản thanh toán .....	90
Hình 3.8: Trang chọn hỗ trợ AWS .....	90
Hình 3.9: Dịch vụ Amazon RDS .....	91
Hình 3.10: Trang tạo cơ sở dữ liệu .....	91
Hình 3.11: Cấu hình cơ sở dữ liệu.....	92
Hình 3.12: Truy cập RDS trên SQL Server.....	92
Hình 3.13: Trang Amazon Elastic Beanstalk .....	93
Hình 3.14: Cấu hình môi trường Elastic Beanstalk.....	93
Hình 3.15: Chọn nền tảng triển khai Website .....	94
Hình 3.16: Cấu hình quyền truy cập dịch vụ.....	95
Hình 3.17: Trang Amazon IAM .....	95
Hình 3.18: Cấu hình vai trò .....	96
Hình 3.19: Thêm quyền EC2 .....	96
Hình 3.20: Cấu hình máy ảo .....	97
Hình 3.21: Cấu hình Amazon EC2 .....	97
Hình 3.22: Cấu hình nhóm bảo mật EC2 .....	98
Hình 3.23: Môi trường cài đặt thành công .....	98
Hình 3.24: Giao diện sau khi thiết lập môi trường .....	99
Hình 3.25: Cài đặt tiện ích AWS Toolkit .....	99
Hình 3.26: Thêm thông tin người dùng AWS Toolkit .....	100

Hình 3.27: Kết nối thành công AWS.....	101
Hình 3.28: Xuất thư mục code lên AWS.....	101
Hình 3.29: Cấu hình đường dẫn lưu trữ thư mục .....	102
Hình 3.30: Triển khai Website lên AWS Elastic Beanstalk.....	102
Hình 3.31: Trạng thái thành công.....	103
Hình 3.32: Giao diện trang chủ Website GearH .....	103

## **DANH MỤC BẢNG**

Bảng 3.1: Ý nghĩa Use case.....	82
Bảng 3.2: Mô tả bảng cơ sở dữ liệu.....	84
Bảng 3.3: Bảng ROLE.....	84
Bảng 3.4: Bảng ACCOUNT.....	84
Bảng 3.5: Bảng BRAND .....	85
Bảng 3.6: Bảng CATEGORY .....	86
Bảng 3.7: Bảng PRODUCT .....	86
Bảng 3.8: Bảng ORDER.....	87
Bảng 3.9: Bảng ORDERDETAIL .....	87
Bảng 3.10: Bảng COUPON.....	88

## **DANH MỤC VIẾT TẮT**

Từ khóa	Tiếng Anh	Tiếng việt
MIT	Massachusetts Institute of Technology	Học viện công nghệ Massachusetts
SaaS	Software as a Service	Phần mềm dưới dạng dịch vụ
PaaS	Platform as a Service	Nền tảng dưới dạng dịch vụ
IaaS	Infrastructure as a Service	Cơ sở hạ tầng dưới dạng dịch vụ
VPN	Virtual Private Network	Mạng ảo riêng
SDN	Software-Defined Networking	Mạng định nghĩa bằng phần mềm
OCCI	Open Cloud Computing Interface	Giao diện mở Điện toán đám mây
CDM	Cloud Data Management	Quản lý dữ liệu đám mây
SMTP	Simple Mail Transfer Protocol	Giao thức tiêu chuẩn truyền tải email
AFS	Andrew File System	Hệ thống tệp Andrew
NFS	Network File System	Hệ thống tệp mạng
RPC	Remote Procedure Call	Gọi thủ tục từ xa
HDFS	Hadoop Distributed File System	Hệ thống tệp phân tán Hadoop
GFS	Google File System	Hệ thống tệp Google
ABE	Attribute-Based Encryption	Mã hóa dựa trên thuộc tính
FHE	Fully Homomorphic Encryption	Mã hóa đồng hình hoàn toàn
SE	Searchable Encryption	Mã hóa có thể tìm kiếm
AWS	Amazon Web Service	Dịch vụ Web Amazon

## **LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến Thầy ThS.Hồ Ngọc Trung Kiên, giảng viên hướng dẫn, người đã dành thời gian và tâm huyết để hỗ trợ, chỉ bảo chúng em trong suốt quá trình thực hiện đồ án. Sự tận tình, kiến thức chuyên sâu và những lời khuyên quý báu của Thầy không chỉ giúp chúng em vượt qua các khó khăn trong quá trình nghiên cứu, mà còn mở rộng thêm tầm nhìn của chúng em về lĩnh vực điện toán đám mây.

## **LỜI MỞ ĐẦU**

Trong thời đại số hóa ngày nay, sự bùng nổ của công nghệ thông tin đã và đang thay đổi sâu sắc cách thức vận hành và hoạt động của các doanh nghiệp cũng như đời sống hàng ngày. Những bước tiến vượt bậc về công nghệ không chỉ mang lại nhiều cơ hội phát triển mà còn đặt ra yêu cầu cao hơn về việc tối ưu hóa tài nguyên và nâng cao hiệu quả hoạt động. Trong bối cảnh đó, điện toán đám mây nổi lên như một trong những giải pháp công nghệ tiên tiến, đáp ứng được nhu cầu ngày càng đa dạng của tổ chức và cá nhân.

Điện toán đám mây mang lại sự tiện lợi vượt trội trong việc quản lý, lưu trữ và xử lý dữ liệu. Bằng cách cung cấp tài nguyên tính toán linh hoạt, hiệu quả và tiết kiệm chi phí, công nghệ này đã giúp các tổ chức, doanh nghiệp giảm thiểu đáng kể chi phí đầu tư vào hạ tầng, đồng thời mở rộng khả năng triển khai các mô hình kinh doanh sáng tạo. Không những thế, điện toán đám mây còn đóng vai trò nền tảng trong sự phát triển của các công nghệ hiện đại như trí tuệ nhân tạo (AI), dữ liệu lớn (Big Data) và Internet vạn vật (IoT), tạo điều kiện thúc đẩy chuyển đổi số và nâng cao năng lực cạnh tranh trong nhiều lĩnh vực.

Từ đó thực hiện đề tài ‘Tìm hiểu về điện toán đám mây và triển khai Website bán Gaming Gear trên Amazon Web Service’ nhằm mục đích tìm hiểu về điện toán đám mây và triển khai một Website trên nền tảng AWS với ba chương chính:

Chương 1: Tổng quan về điện toán đám mây

Chương 2: Lưu trữ và bảo mật trong điện toán đám mây

Chương 3: Triển khai Website bán Gaming Gear trên AWS

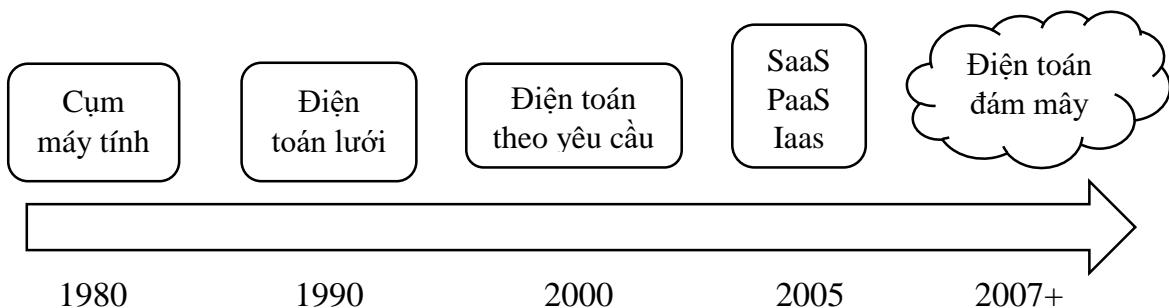
# CHƯƠNG 1. TỔNG QUAN VỀ ĐIỆN TOÁN ĐÁM MÂY

Trong chương này, sẽ tìm hiểu lịch sử hình thành, quá trình phát triển về điện toán đám mây. Phân tích ưu, nhược điểm, một số mô hình và cấu trúc của điện toán đám mây. Tìm hiểu về liên điều hành trong điện toán đám mây.

## 1.1. Giới thiệu chung

### 1.1.1. Lịch sử ra đời

Điện toán đám mây đã phát triển dựa trên các công nghệ truyền thống. Vì vậy, điện toán đám mây có thể được ánh xạ với các công nghệ cũ hơn trước khi nó trưởng thành đến mức hiện tại như điện toán tiện ích (Utility Computing) và điện toán lưới (Grid Computing) [1].



Hình 1.1: Lịch sử ra đời Điện toán đám mây

Trong những năm 1980, các máy tính lớn và các cụm máy tính được sử dụng để xử lý dữ liệu và thực hiện các tác vụ tính toán trong các tổ chức lớn. Hệ thống này yêu cầu người dùng phải truy cập trực tiếp vào các máy tính lớn. Mặc dù công nghệ này có khả năng xử lý dữ liệu mạnh mẽ, nhưng nó thiếu tính linh hoạt và khả năng mở rộng dễ dàng, điều này dẫn đến việc nghiên cứu các giải pháp khác [1].

Vào khoảng năm 1990, mức giá cho việc xử lý hoặc điện toán đều cao cho bất kỳ dịch vụ nào, vì vậy họ đã nảy ra ý tưởng chia sẻ tài nguyên máy tính. Mục tiêu của nó là tích hợp máy chủ, hệ thống lưu trữ và ứng dụng được phân phối trên toàn thế giới để chia sẻ với nhiều người dùng. Việc chia sẻ cho phép người dùng sử dụng và chia sẻ tài nguyên máy tính và khách hàng sử dụng tài nguyên đó có thể trả tiền cho các dịch vụ được sử dụng trong khoảng thời gian họ sử dụng dịch vụ cần thiết [2].

Năm 1990, khái niệm điện toán lưới (Grid Computing) bắt đầu xuất hiện. Điện toán lưới là quá trình giải quyết các vấn đề điện toán lớn thành các vấn đề nhỏ hơn và giải quyết các vấn đề này trên các máy hoặc máy tính đơn giản hoặc hiệu suất thấp để

có được kết quả cuối cùng cho các vấn đề lớn bằng cách phân phối nhiệm vụ giữa các máy khác nhau trên lưới [3].

Đến đầu những năm 2000, điện toán theo yêu cầu bắt đầu được phát triển, cho phép người dùng thuê tài nguyên tính toán (như bộ vi xử lý, bộ nhớ và dung lượng lưu trữ) từ các nhà cung cấp dịch vụ mà không cần phải đầu tư vào phần cứng. Khái niệm này dựa trên việc cung cấp tài nguyên linh hoạt, có thể mở rộng theo nhu cầu của người dùng mà không phải duy trì phần cứng cố định. Công nghệ này mở ra cơ hội lớn cho các doanh nghiệp nhỏ và lớn sử dụng tài nguyên tính toán mà không cần phải lo lắng về chi phí và quản lý hạ tầng [1].

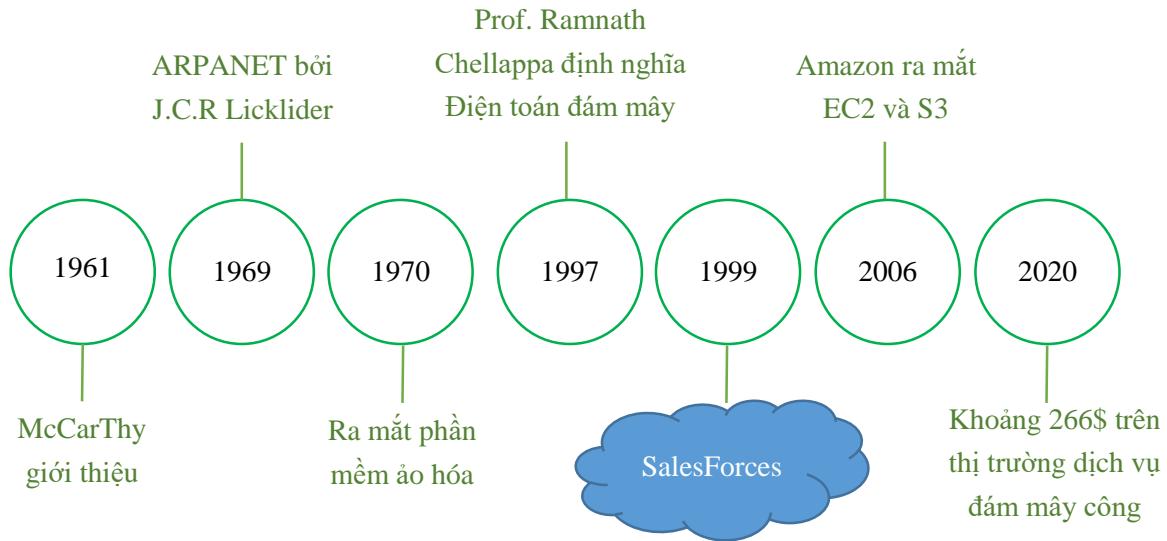
Vào năm 2005, các mô hình dịch vụ đám mây như Software as a Service (SaaS), Platform as a Service (PaaS) và Infrastructure as a Service (IaaS) bắt đầu phát triển mạnh mẽ. SaaS cung cấp phần mềm ứng dụng qua Internet (như Google Apps, Salesforce), PaaS cung cấp nền tảng để phát triển và triển khai ứng dụng mà không cần quan tâm đến hạ tầng, còn IaaS cung cấp cơ sở hạ tầng máy tính linh hoạt (như máy chủ ảo, lưu trữ) trên đám mây [3].

Đến cuối những năm 2000, điện toán đám mây (Cloud Computing) chính thức phát triển mạnh mẽ. Khái niệm điện toán đám mây rất giống với khái niệm điện toán tiện ích và điện toán lưới. Với sự phát triển của công nghệ trong vài thập kỷ, điện toán đám mây có thể đạt được mục tiêu trong vài năm qua. Trong thập kỷ này, điện toán đám mây đã trưởng thành hơn rất nhiều về mặt công nghệ để đáp ứng mọi nhu cầu của nó [1].

Sự phát triển này đã thay đổi hoàn toàn cách thức hoạt động của các tổ chức và cá nhân trong việc sử dụng và quản lý tài nguyên máy tính, thúc đẩy sự ra đời của các ứng dụng và dịch vụ trực tuyến, và tạo nền tảng cho sự bùng nổ của các công nghệ như big data, AI và IoT.

### **1.1.2. Quá trình phát triển**

Điện toán đám mây (Cloud Computing) được nhắc đến từ năm 1961 bởi John McCarThy tại MIT [1]. Sau đó các công ty công nghệ trên thế giới ra đời và mạng internet được hình thành và phát triển mạnh mẽ cho đến ngày nay.



Hình 1.2: Quá trình phát triển Điện toán đám mây

Bước ngoặt của Điện toán đám mây xảy ra vào cuối những năm 1990 khi internet phát triển mạnh mẽ, cho phép các doanh nghiệp bắt đầu cung cấp dịch vụ trực tuyến quy mô lớn. Theo báo cáo của Varghese vào năm 2019, cho rằng Internet chính là động lực giúp các mô hình dịch vụ điện toán đám mây như ngày nay phát triển mạnh mẽ, cung cấp khả năng truy cập và quản lý tài nguyên từ xa một cách dễ dàng [4].

Một trong những công ty đầu tiên làm việc với khái niệm Điện toán đám mây được thành lập bởi Salesforce vào cuối năm 1990 [5]. Công ty bắt đầu cung cấp Phần mềm dưới dạng Dịch vụ (SaaS), cung cấp dịch vụ quản lý quan hệ khách hàng cho người dùng. Mô hình Salesforce là một trong những mô hình điển hình của điện toán đám mây, bên cạnh PaaS.

Vào năm 2006, Amazon đã đóng vai trò quan trọng và tung ra thị trường công nghệ Amazon Web Service (AWS) vào năm 2006 [6]. Cùng với điều này Google và IBM cũng bắt đầu các dự án nghiên cứu về Điện toán đám mây Eucalyptus trở thành nền tảng mã nguồn mở đầu tiên triển khai các đám mây riêng [6] [7].

Vào tháng 7 năm 2010, NASA và Rackspace bắt đầu một dự án chung mang tên OpenStack với một số nhà cung cấp bao gồm AMD, Intel, Dell. Sau đó nhiều tổ chức khác đã tham gia dự án. Hiện có hơn 500 công ty đang hỗ trợ dự án và hơn 6800 công ty đang sử dụng OpenStack để triển khai dịch vụ đám mây của họ [1].

### 1.1.3. Khái niệm

Điện toán đám mây là một trong những thuật ngữ kỹ thuật mơ hồ nhất trong lịch sử. Một trong số những lý do là điện toán đám mây có thể được sử dụng trong nhiều

tính huống ứng dụng, lý do khác là điện toán đám mây được nhiều công ty quảng cáo rầm rộ để thúc đẩy kinh doanh. Có khá nhiều định nghĩa và ẩn dụ về điện toán đám mây.

Theo Ling Qian và các đồng sự, điện toán đám mây là một loại kỹ thuật điện toán trong đó các dịch vụ IT được cung cấp bởi các đơn vị tính toán không lò có chi phí thấp hơn được kết nối bằng IP [8]. Theo Ian Foster và các đồng sự và Rajkumar Buyya, điện toán đám mây là mô hình điện toán phân tán có tính co giãn lớn mà hướng theo co giãn về mặt kinh tế, là nơi chứa các sức mạnh tính toán, kho lưu trữ, các nền tảng và các dịch vụ được trực quan, ảo hóa và co giãn linh động, sẽ được phân phối theo nhu cầu cho các khách hàng thông qua mạng Internet [9] [10].

Tuy nhiên khái niệm được nhiều sự công nhận và bao quát nhất là của NIST (Viện tiêu chuẩn và công nghệ quốc gia Bộ Thương mại Mỹ), Điện toán đám mây là một mô hình cho phép truy cập mạng thuận tiện, theo nhu cầu đến một kho tài nguyên điện toán dùng chung, có thể định cấu hình (ví dụ như mạng, máy chủ, lưu trữ, ứng dụng) có thể được cung cấp và thu hồi một cách nhanh chóng với yêu cầu tối thiểu về quản lý hoặc can thiệp của nhà cung cấp dịch vụ [11].

#### **1.1.4. Ưu điểm và nhược điểm**

Ưu điểm của điện toán đám mây bao gồm: Giảm chi phí triển khai và bảo trì; Hạ tầng linh hoạt, dễ dàng mở rộng quy mô; Đẩy nhanh tốc độ đưa sản phẩm ra thị trường; Tái định hướng bộ phận IT, tập trung vào đổi mới thay vì bảo trì và triển khai; Tăng cường khả năng tiếp cận ứng dụng hiệu suất cao cho các doanh nghiệp vừa và nhỏ; Sao lưu và khôi phục dễ dàng hơn; Tích hợp sẵn phần mềm tự động hóa; Dễ dàng truy cập dữ liệu [12].

Nhược điểm của điện toán đám mây liên quan đến vấn đề kỹ thuật, vấn đề bảo mật, dễ bị tấn công. Mặc dù dữ liệu trên đám mây có thể truy cập mọi lúc, mọi nơi nhưng vẫn có lúc hệ thống có thể gặp các lỗi nghiêm trọng. Có thể bị ngừng hoạt động và gặp các vấn đề kỹ thuật khác. Ngay cả những nhà cung cấp dịch vụ đám mây cũng gặp những lỗi này, mặc dù luôn duy trì bảo trì.

Trước khi sử dụng đám mây, người dùng phải biết rằng họ sẽ giao tất cả thông tin nhạy cảm của doanh nghiệp mình cho nhà cung cấp dịch vụ. Do đó, doanh nghiệp cần lựa chọn nhà cung cấp dịch vụ uy tín, đáng tin cậy. Việc lưu trữ dữ liệu trên đám mây có thể khiến doanh nghiệp dễ bị tấn công và đánh cắp dữ liệu nhạy cảm. Có thể bị ngừng hoạt động: Phụ thuộc vào kết nối Internet của doanh nghiệp. Không linh hoạt: Có nghĩa là việc chọn nhà cung cấp dịch vụ thường các doanh nghiệp buộc phải sử dụng những ứng dụng hoặc định dạng độc quyền của nhà cung cấp dịch vụ [13].

### **1.1.5. *Tầm quan trọng đối với doanh nghiệp***

Điện toán đám mây giúp doanh nghiệp giảm thiểu các khoản đầu tư vào cơ sở hạ tầng phần cứng và phần mềm. Hệ thống đám mây cung cấp dịch vụ cho doanh nghiệp, chỉ cần trả tiền cho dịch vụ và tài nguyên mà họ thực sự sử dụng, giúp giảm đáng kể chi phí ban đầu và chi phí vận hành. Đám mây cho phép doanh nghiệp làm việc từ xa, chia sẻ dữ liệu và công tác một cách dễ dàng. Điều này giúp các quy trình trong doanh nghiệp được tự động hóa và thời gian quản lý được tối ưu.

Đám mây cho phép doanh nghiệp dễ dàng mở rộng hoặc giảm tải các tài nguyên công nghệ tùy thuộc vào nhu cầu. Điều này giúp đáp ứng nhanh chóng các thay đổi của thị trường và nhu cầu khách hàng. Dễ dàng tiếp cận cho doanh nghiệp vừa và nhỏ với mức chi phí phù hợp để triển khai phần mềm, ứng dụng. Điện toán đám mây hỗ trợ nhiều công nghệ tiên tiến như AI, phân tích dữ liệu và các công cụ phát triển phần mềm mới, doanh nghiệp có thể thử nghiệm các sáng kiến và cải tiến quy trình mà không cần phải đầu tư vào hạ tầng phức tạp.

## **1.2. Các mô hình triển khai và cung cấp dịch vụ**

### **1.2.1. *Mô hình triển khai***

#### **1.2.1.1. *Đám mây công cộng***

Điện toán đám mây công cộng là mô hình dịch vụ mà tài nguyên và dịch vụ được cung cấp từ các bên thứ ba và không nằm trong cơ sở của doanh nghiệp. Nhiều doanh nghiệp khác nhau có thể sử dụng tài nguyên này để cung cấp dịch vụ cho người dùng. Hầu hết các dịch vụ đều được cung cấp trong môi trường công khai, trong đó người tiêu dùng có thể truy cập vào nhóm tài nguyên được quản lý bởi một công ty tổng [14].

Người dùng chỉ cần trả tiền cho khoảng thời gian họ sử dụng dịch vụ, có nghĩa là họ phải trả tiền cho mỗi lần sử dụng [6]. Điều này giúp giảm chi phí vận hành đối với chi tiêu công nghệ thông tin. Đám mây công cộng (Public Cloud) kém an toàn hơn các mô hình đám mây khác vì tất cả dữ liệu và ứng dụng trên đám mây công cộng đều dễ bị tấn công độc hại. Giải pháp cho vấn đề này là việc kiểm tra bảo mật có thể được thực hiện thông qua xác thực của cả hai bên, bởi nhà cung cấp dịch vụ đám mây cũng như khách hàng [15].

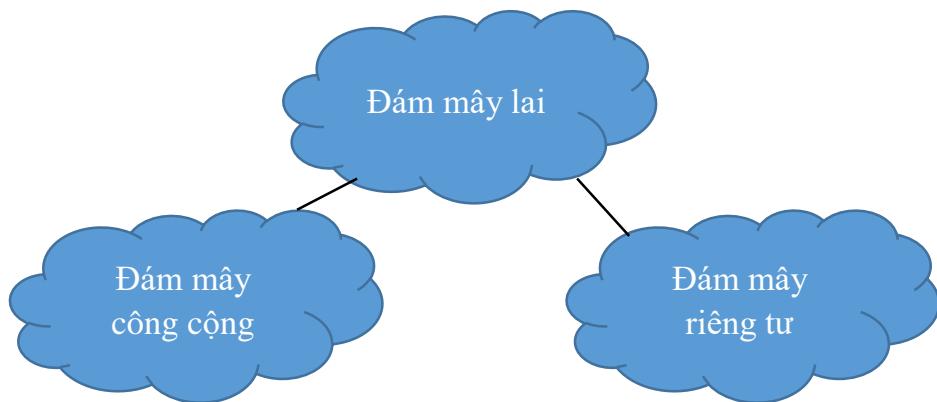
#### **1.2.1.2. *Đám mây riêng tư***

Bên thứ ba sẽ cung cấp các dịch vụ phân biệt nó với các quyền truy cập công cộng. Điều này giúp đám mây riêng tư tốt hơn đám mây công cộng vì nó ngăn chặn sự truy cập trái phép [14]. Hoạt động của đám mây riêng tư nằm trong trung tâm dữ liệu doanh nghiệp nội bộ của một tổ chức.

Ưu điểm chính là dễ dàng quản lý bảo mật, bảo trì và nâng cấp hơn, đồng thời cung cấp khả năng kiểm soát việc triển khai và sử dụng. So với đám mây công cộng nơi tất cả tài nguyên và ứng dụng được quản lý bởi nhà cung cấp dịch vụ, thì trong đám mây riêng tư, các dịch vụ này được gộp lại với nhau và được cung cấp cho người dùng ở cấp độ tổ chức. Những tài nguyên này sẽ được quản lý bởi chính tổ chức [6].

#### 1.2.1.3. *Đám mây lai*

Đám mây lai (Hybrid Cloud) là dạng đám mây các dịch vụ đám mây được cung cấp dưới dạng kết hợp nhiều loại đám mây (đám mây công cộng, đám mây riêng tư, đám mây cộng đồng). Loại đám mây này có thể chỉ kế thừa bất kỳ loại lỗ hổng hoặc rủi ro từ những loại đám mây được kết hợp [14].



Hình 1.3: *Đám mây lai (Hybrid Cloud)*

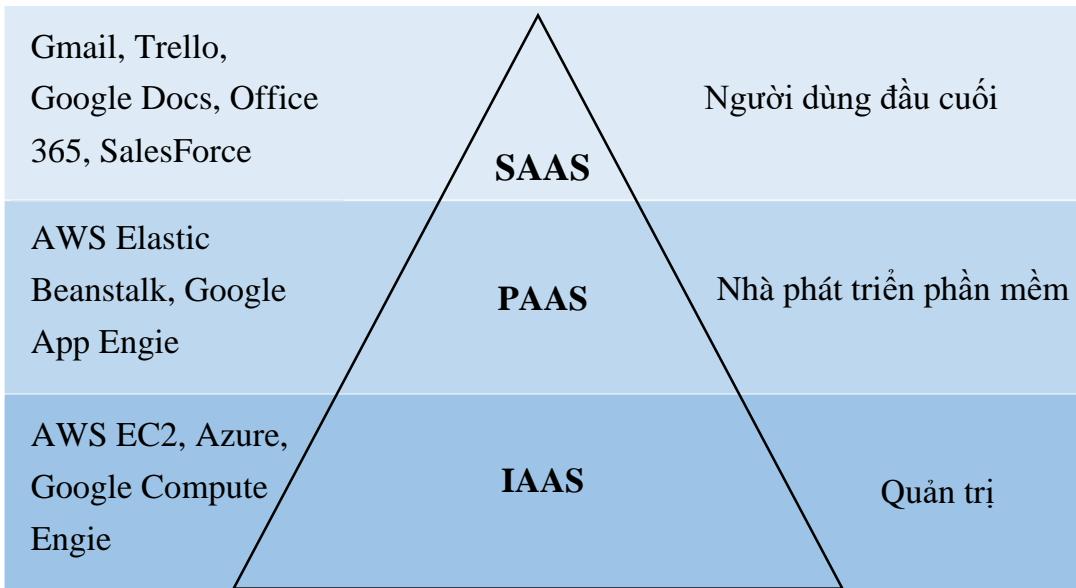
Đám mây lai (Hybrid Cloud) là cách an toàn hơn để kiểm soát dữ liệu và ứng dụng, đồng thời cho phép truy cập thông tin qua Internet. Nó cho phép tổ chức phục vụ nhu cầu của mình trên đám mây riêng và nếu có nhu cầu, tổ chức có thể yêu cầu đám mây công cộng cung cấp tài nguyên máy tính [6].

#### 1.2.1.4. *Đám mây cộng đồng*

Đám mây cộng đồng (Community Cloud) cho phép môi trường điện toán đám mây được chia sẻ hoặc quản lý bởi nhiều tổ chức liên quan. Khi nhiều tổ chức kết hợp với nhau xây dựng và chia sẻ cơ sở hạ tầng đám mây. Cơ sở hạ tầng đám mây được lưu trữ bởi bên thứ ba hoặc một trong các tổ chức trong cộng đồng [6]. Trong đám mây cộng đồng, tất cả các thành viên đều có quyền truy cập bình đẳng như nhau vào các dịch vụ được chia sẻ [14].

#### 1.2.2. *Mô hình cung cấp dịch vụ*

Mô hình dịch vụ điện toán đám mây chủ yếu gồm 2 loại chính: Phần mềm dưới dạng dịch vụ (SaaS), Nền tảng dưới dạng dịch vụ (PaaS), Cơ sở hạ tầng dưới dạng dịch vụ (IaaS).



Hình 1.4: Mô hình cung cấp dịch vụ

#### 1.2.2.1. Phần mềm dưới dạng dịch vụ

Người dùng và khách hàng có thể sử dụng SaaS thông qua Internet. Thông qua mô hình cung cấp dịch vụ này, người dùng có thể sử dụng các dịch vụ ứng dụng phần mềm qua mạng theo yêu cầu [6]. Người dùng không quản lý hoặc kiểm soát cơ sở hạ tầng đám mây cơ bản bao gồm mạng, máy chủ, hệ điều hành, bộ lưu trữ hoặc thậm chí các khả năng của những ứng dụng riêng lẻ, ngoại trừ các cài đặt cấu hình ứng dụng dành riêng cho người dùng cụ thể [11].

**Đặc điểm của SaaS [6]:** Truy cập phần mềm qua Internet; Phần mềm sẽ được quản lý từ một trung tâm; Cung cấp phần mềm theo mô hình ‘one to many’; Không cần phải nâng cấp phần mềm thường xuyên; APIs cho phép tích hợp các phần mềm khác nhau.

#### 1.2.2.2. Nền tảng dưới dạng dịch vụ

Trong PaaS, nền tảng này được cung cấp cho khách hàng và họ sẽ triển khai phần mềm, ứng dụng của riêng mình trên Đám mây. Cách tiếp cận của dịch vụ này là hướng tới phần mềm và các công cụ phát triển [6]. Khách hàng không quản lý hoặc kiểm soát cơ sở hạ tầng đám mây cơ bản bao gồm mạng, máy chủ, hệ điều hành, bộ lưu trữ nhưng có thể kiểm soát các ứng dụng đã triển khai và có thể cài đặt cấu hình cho môi trường lưu trữ ứng dụng [11].

**Đặc điểm của PaaS [6]:** Giảm chi phí phát triển và bảo trì khi phát triển, triển khai và thử nghiệm bất kỳ phần mềm, ứng dụng nào trên cùng môi trường tích hợp; Cung cấp dịch vụ bảo mật; Quản lý cơ sở dữ liệu; Không cần phải tải xuống hoặc cài đặt để trải nghiệm phần mềm trực tuyến. Ví dụ như Facebook, Gmail; Cung cấp khả năng mở

rộng, độ tin cậy và bảo mật được tích hợp sẵn; Kiến trúc được chia sẽ có nghĩa là người dùng có thể truy cập ứng dụng. Ví dụ như Microsoft Azure, Amazon EC2, ...

#### *1.2.2.3. Cơ sở hạ tầng dưới dạng dịch vụ*

IaaS cung cấp dịch vụ để có được một máy chủ ảo và chỉ cần trả tiền cho tài nguyên này [16]. Cho phép khả năng tiếp cận cơ sở hạ tầng sử dụng Internet bao gồm máy chủ, thiết bị lưu trữ và các thiết bị ngoại vi khác. Có thể kết hợp với các dịch vụ được quản lý để hỗ trợ hệ điều hành và ứng dụng.

Trong mô hình IaaS, khách hàng có thể trực tiếp sử dụng các chức năng cơ sở hạ tầng (lưu trữ, tường lửa, mạng, ...). Cung cấp cho khách hàng khả năng vật lý hoặc tài nguyên ảo bao gồm CPU, bộ nhớ, hệ điều hành để đáp ứng nhu cầu của người dùng. Khách hàng có thể quản lý hoạt động cho môi trường tổng thể qua giao diện Web.

**Đặc điểm** của IaaS [6]: Phân phối tài nguyên như một dịch vụ; Trong IaaS chi phí cho từng tài nguyên khác nhau; Nhiều khách hàng có thể truy cập trên cùng một phần cứng; Có thể mở rộng; Không cần quản trị và bảo trì phần cứng của hệ thống.

### **1.3. Liên điều hành trong điện toán đám mây**

#### **1.3.1. Khái niệm**

Liên điều hành trong điện toán đám mây là khả năng để các hệ thống đám mây khác nhau có thể giao tiếp, trao đổi dữ liệu và hoạt động cùng nhau một cách hiệu quả. Liên điều hành không chỉ đơn giản là di chuyển dữ liệu giữa các đám mây mà còn bao gồm việc tích hợp các dịch vụ và ứng dụng trên các nền tảng đám mây khác nhau để tạo thành một hệ thống liền mạch [17]. Điều này bao gồm việc cho phép các ứng dụng và dịch vụ trên đám mây có thể trao đổi dữ liệu, giao tiếp và tương tác với nhau mà không cần sự can thiệp phức tạp từ người dùng [18].

#### **1.3.2. Vai trò**

Liên điều hành trong điện toán đám mây đóng vai trò quan trọng trong việc hỗ trợ khả năng hợp tác và tích hợp dịch vụ giữa các hệ thống đám mây khác nhau. Điều này giúp cho doanh nghiệp và tổ chức có thể tận dụng đồng thời nhiều dịch vụ từ các nhà cung cấp đám mây khác nhau mà không bị giới hạn bởi sự phụ thuộc vào một nền tảng cụ thể. Điều này không chỉ gia tăng hiệu quả sử dụng tài nguyên mà còn mở rộng khả năng đáp ứng nhu cầu phức tạp trong hệ thống [17].

Liên điều hành trong điện toán đám mây là chìa khóa giúp các tổ chức khai thác tối đa lợi ích của đa đám mây (multi-cloud). Khả năng tương thích này không chỉ đảm bảo các đám mây có thể giao tiếp với nhau mà còn giảm thiểu nguy cơ khóa chặt công

nghệ (vendor lock-in), khi các doanh nghiệp bị ràng buộc vào một nhà cung cấp duy nhất [18].

Liên điều hành thúc đẩy khả năng mở rộng và đáp ứng nhu cầu ngày càng đa dạng, đồng thời cho phép tích hợp các dịch vụ và ứng dụng trên nhiều đám mây khác nhau. Giúp doanh nghiệp tăng cường khả năng phục hồi, bảo mật và kiểm soát dữ liệu, đặc biệt trong hệ thống phân tán và phức tạp [18].

### **1.3.3. Thành phần và công cụ hỗ trợ**

Thành phần hỗ trợ liên điều hành trong điện toán đám mây [19]. API là thành phần quan trọng cho phép dịch vụ và ứng dụng khác nhau giao tiếp với nhau. Cung cấp các phương thức để truy cập chức năng và dữ liệu từ dịch vụ đám mây. Mạng liên đám mây cung cấp kết nối ổn định, bảo mật và hiệu quả giữa các nền tảng đám mây.

Các thành phần như VPN, SDN và mạng lai giúp quản lý luồng dữ liệu. Cloud Broker đóng vai trò trung gian, hỗ trợ việc quản lý, tích hợp và điều phối các dịch vụ từ nhiều nhà cung cấp đám mây. Thường hỗ trợ doanh nghiệp theo dõi, chọn lựa, và kết hợp các dịch vụ đám mây, giúp quá trình chuyển đổi linh hoạt và dễ dàng hơn.

Một số công cụ hỗ trợ liên điều hành phổ biến trong điện toán đám mây: Kubernetes giúp quản lý các container và ứng dụng trên nhiều nền tảng đám mây khác nhau, hỗ trợ tính di động và liên điều hành bằng cách triển khai và quản lý các ứng dụng đám mây đa nền tảng [20]. OpenStack là nền tảng mã nguồn mở để xây dựng đám mây công cộng và đám mây riêng. Cung cấp API chuẩn hóa, giúp các đám mây khác nhau dễ dàng giao tiếp [21].

### **1.3.4. Chuẩn hóa liên điều hành**

Chuẩn hóa liên điều hành là một quá trình cần thiết để đảm bảo rằng các dịch vụ và ứng dụng điện toán đám mây có thể tương tác một cách hiệu quả liền mạch. Quá trình này giúp người dùng dễ dàng chuyển đổi giữa các nhà cung cấp dịch vụ khác nhau mà không gặp khó khăn về tính tương thích. Ví dụ OCCI và CDMI đã được đề xuất trong việc chuẩn hóa liên điều hành trong điện toán đám mây.

Các yếu tố chính của chuẩn hóa liên điều hành [22]: Giao thức và API: Việc thiết lập các giao thức và API tiêu chuẩn là cần thiết để các dịch vụ có thể giao tiếp với nhau. Định dạng dữ liệu: Sử dụng định dạng dữ liệu tiêu chuẩn như JSON hoặc XML để đảm bảo dữ liệu có thể được chia sẻ một cách dễ dàng. Quản lý danh tính: Cần có tiêu chuẩn cho việc quản lý danh tính và truy cập, cho phép người dùng duy trì quyền truy cập vào các dịch vụ khác nhau.

### **1.3.5. Thách thức**

#### *1.3.5.1. Tính di động và khả năng di chuyển*

Khả năng di chuyển (Portability) là khả năng di chuyển một hình ảnh trong trạng thái tắt từ một máy chủ này sang máy chủ khác và sau đó khởi động tại điểm đến. Tính di động (Mobility) là khả năng chuyển một khối lượng công việc đang hoạt động từ máy chủ này sang máy chủ khác mà không mất kết nối với người dùng hoặc trạng thái đang xử lý.

Khả năng di chuyển và tính di động là chỉ số chính cho mức độ liên điều hành giữa các đám mây, và di động giữa các nhà cung cấp đám mây sẽ là một mục tiêu của một hệ thống đám mây liên điều hành trưởng thành [17].

#### *1.3.5.2. Tích hợp dịch vụ đám mây*

Để đạt được ‘các giải pháp tốt nhất’ và duy trì kiểm soát đối với các hoạt động và dữ liệu quan trọng, các doanh nghiệp có thể cần tích hợp cả ứng dụng tại doanh nghiệp và ứng dụng SaaS để đáp ứng nhu cầu của doanh nghiệp. Việc tích hợp các phần mềm thông qua API hiện nay đòi hỏi số lượng mã hóa lớn cũng như bảo trì liên tục do phải sửa đổi và cập nhật thường xuyên. Việc kết hợp ứng dụng SaaS và ứng dụng triển khai tại chỗ tương tác thông qua dịch vụ Web và áp dụng các nguyên tắc kiến trúc SOA để triển khai logic nghiệp vụ có thể giải quyết vấn đề này [17].

#### *1.3.5.3. Bảo mật, quyền riêng tư*

Khách hàng sử dụng dịch vụ điện toán đám mây thường mong muốn từ các nhà cung cấp dịch vụ về mức độ an toàn và bảo mật dữ liệu cao, ví dụ như kiểm soát quyền truy cập của người dùng đối với thông tin nhạy cảm. Điều này đòi hỏi các giải pháp hiệu quả để giải quyết các vấn đề bảo mật truyền thống xuất hiện trong các hệ thống bảo mật đa cấp và đa miền, như quản lý danh tính liên kết, kiểm soát truy cập dựa trên vai trò, cùng với giám sát, ghi nhật ký [17].

Mặc dù điện toán đám mây có thể giúp giảm chi phí đáng kể nhờ vào việc hợp nhất và tối ưu hóa tài nguyên, nhưng nó cũng tạo ra các vấn đề an ninh và bảo mật đặc thù mà cần được giải quyết để phát triển bền vững. Ngoài ra, các nhà cung cấp dịch vụ điện toán đám mây cần đảm bảo mức độ minh bạch về bảo mật, nhằm giải tỏa lo ngại của khách hàng về tính an toàn của các dịch vụ đám mây. Tối thiểu, các nhà cung cấp cần cung cấp cho khách hàng sự đảm bảo an ninh tương đương với hệ thống công nghệ thông tin nội bộ [17].

Các vấn đề về an ninh có liên quan chặt chẽ đến việc quản trị đám mây, bao gồm quản lý người dùng, tài nguyên và dữ liệu, thông qua các chính sách bảo mật nhằm xử

lý xác thực, kiểm soát truy cập, quản lý phiên và truyền thông mạng. Việc chuyển đổi từ mô hình máy chủ khách truyền thống sang mô hình đám mây có thể giải quyết một số vấn đề an ninh hiện có, nhưng cũng sẽ phát sinh những vấn đề mới.

Nếu không hiểu rõ các vấn đề này hoặc áp dụng một cách mù quáng các chính sách bảo mật cũ vào môi trường đám mây sẽ dẫn đến các rủi ro an ninh tiềm tàng [17]. Cần thiết lập các mối quan hệ tin cậy chính thức giữa các đám mây để người dùng có thể truy cập và kiểm soát tài nguyên từ xa một cách an toàn [23].

#### 1.3.5.4. Quản lý, giám sát

Bên cạnh việc áp dụng một bộ công cụ quản lý bảo mật và danh tính người dùng duy nhất cho các ứng dụng chạy trên nhiều nền tảng đám mây khác nhau, các công ty cần một bộ công cụ thống nhất để tự động cung cấp dịch vụ, quản lý các phiên bản máy ảo, và làm việc với cả ứng dụng trên đám mây và tại chỗ. Điều này giúp tối ưu hóa quy trình và đảm bảo rằng quản lý an ninh và tuân thủ chính sách được duy trì nhất quán giữa các nền tảng khác nhau [17].

Người dùng dịch vụ đám mây cũng cần đảm bảo rằng các chính sách bảo mật và quyền riêng tư luôn được áp dụng một cách đồng bộ và các cam kết cấp độ dịch vụ luôn được thực hiện đầy đủ ngay cả khi dịch vụ đám mây được di chuyển qua các biên giới đám mây khác nhau. Việc này đòi hỏi các quy trình và bộ công cụ nhất quán để giám sát và báo cáo mức độ dịch vụ, cũng như sự tuân thủ hoặc vi phạm các chính sách bảo mật và quyền riêng tư của các đám mây từ xa [17].

### 1.4. Đặc trưng kỹ thuật

Tự phục vụ theo yêu cầu (On-demand self-service), khách hàng có thể tự động yêu cầu và nhận các tài nguyên như máy chủ và lưu trữ, mà không cần đến sự can thiệp của nhà cung cấp dịch vụ [11]. Truy cập rộng (Broad network access), tài nguyên đám mây có thể truy cập qua Internet và trên các thiết bị tiêu chuẩn như điện thoại, máy tính bảng và máy tính cá nhân, giúp người dùng linh hoạt hơn trong việc sử dụng dịch vụ [11].

Chia sẻ tài nguyên (Resource pooling), tài nguyên đám mây được gộp chung và phân bổ linh hoạt cho nhiều người dùng bằng cách sử dụng công nghệ đa thuê. Điều này giúp cung cấp tài nguyên một cách linh động theo nhu cầu từng người dùng mà không cần định vị cụ thể về tài nguyên [11]. Độ co giãn nhanh (Rapid Elasticity), tài nguyên máy tính có thể được cung cấp và phát hành nhanh chóng và linh hoạt dựa trên nhu cầu của người tiêu dùng. Khách hàng xem những tài nguyên này như thể chúng là vô hạn và có thể mua với số lượng bất kỳ vào bất kỳ lúc nào [24].

Dịch vụ được đo lường (Measured Services), tài nguyên và dịch vụ đám mây được nhà cung cấp dịch vụ giám sát, kiểm soát và tối ưu hóa thông qua mô hình kinh doanh trả tiền cho mỗi lần sử dụng. Đa bên thuê (Multi-tenancy), trong môi trường đám mây, các dịch vụ thuộc sở hữu của nhiều nhà cung cấp được đặt cùng vị trí trong một trung tâm dữ liệu. Các vấn đề về hiệu suất và quản lý của các dịch vụ được chia sẻ giữa các nhà cung cấp dịch vụ và nhà cung cấp cơ sở hạ tầng [25].

Cung cấp tài nguyên động (Dynamic resource provisioning): Một trong những tính năng chính của điện toán đám mây là tài nguyên máy tính có thể được lấy và giải phóng một cách nhanh chóng. So với mô hình truyền thống cung cấp tài nguyên theo nhu cầu cao nhất, việc cung cấp tài nguyên động cho phép các nhà cung cấp dịch vụ có được tài nguyên dựa trên nhu cầu hiện tại, điều này có thể giảm đáng kể chi phí vận hành [25].

## 1.5. Các ứng dụng của điện toán đám mây

### 1.5.1. E-Learning

E-learning là một xu hướng mới trong giáo dục nhằm tận dụng tốt nhất công nghệ thông tin. Điện toán đám mây là môi trường hấp dẫn đối với sinh viên, giảng viên và nhà nghiên cứu. Là một lĩnh vực CNTT mới nổi, điện toán đám mây có thể cung cấp cho các trường đại học và trung tâm nghiên cứu cơ sở hạ tầng tính toán mạnh mẽ và tiết kiệm chi phí. Sinh viên có thể kết nối với các dịch vụ giáo dục của trường thông qua thiết bị di động cá nhân từ bất cứ đâu. Các giảng viên có thể truy cập hiệu quả và linh hoạt vào tài liệu khóa học của họ trong phòng học [24].

### 1.5.2. ERP

Hệ thống ERP có một số hạn chế khi doanh nghiệp phát triển bên trong một tổ chức, có thể cần đến các ứng dụng phần mềm khác nhau để quản lý thông tin trong nhiều lĩnh vực như nhân sự, tiền lương, tài chính và quản trị. Rõ ràng, việc mua, cài đặt và bảo trì nhiều loại ứng dụng phần mềm như vậy là một thách thức đối với sự phát triển của doanh nghiệp. Hơn nữa, các hệ thống ERP truyền thống bị hạn chế về khả năng tiếp cận của nhiều người dùng, hiệu suất và tính sẵn có của tài nguyên.

Đám mây ERP để cập đến việc cài đặt các ứng dụng ERP trên cơ sở hạ tầng đám mây (ví dụ: máy chủ trong trung tâm dữ liệu) để các tổ chức có thể truy cập chung ở mọi nơi thông qua kết nối mạng. Do đó, với công nghệ này, người quản lý dự án không cần phải lo lắng về việc cài đặt, nâng cấp và bảo trì các ứng dụng trong tổ chức của mình. Ngoài ra, đám mây ERP cung cấp cho các tổ chức các tài nguyên có thể mở rộng hiệu quả về mặt chi phí, tính sẵn sàng cao của dữ liệu và ứng dụng cũng như khả năng truy cập đa dạng [24].

### **1.5.3. E-Government**

Công nghệ điện toán đám mây có thể cải thiện đáng kể cách thức hoạt động của chính phủ, các dịch vụ mà chính phủ cung cấp cho người dân và các tổ chức cũng như sự hợp tác của chính phủ với các chính phủ khác. Nó có thể giúp giải quyết những thách thức này bằng cách cung cấp môi trường có khả năng mở rộng linh hoạt, tùy chỉnh và có tính sẵn sàng cao. Nó cũng giúp các chính phủ giảm bớt gánh nặng nâng cấp, bảo trì và cấp phép và cho phép họ tập trung vào công việc cốt lõi.

Khả năng lưu trữ dữ liệu có thể mở rộng và hiệu quả về mặt chi phí cũng có thể được cung cấp, đồng thời có thể sử dụng tính năng sao chép tệp và nhiều cài đặt ở các vị trí tách biệt về mặt địa lý để khôi phục dữ liệu trong trường hợp thảm họa. Các phương pháp bảo mật là một kỹ thuật có thể được triển khai trên Đám mây để ngăn chặn hành vi độc hại và bảo vệ tính toàn vẹn và bảo mật dữ liệu [24].

### **1.5.4. Hệ thống lưu trữ Amazon**

Tất cả các cá thể máy Amazon đi kèm với một lượng lưu trữ tạm thời nhất định để tính toán. Lưu trữ này không liên tục và chỉ tồn tại khi cá thể máy còn sống. Lưu trữ này tham gia vào nhóm hệ thống lưu trữ AWS khi chấm dứt cá thể máy. Do đó, để lưu trữ dữ liệu vĩnh viễn, một số hệ thống lưu trữ liên tục phải được sử dụng. Hệ thống lưu trữ đơn giản Amazon (Amazon S3) và kho lưu trữ dữ liệu khỏi Amazon (Amazon EBS).

## **1.6. Mô hình truyền thông**

Trong kiến trúc điện toán đám mây, mô hình truyền thông được xây dựng trên các thành phần chính gồm các nút và các kênh hoặc kiênn kết truyền thông. Mỗi nút có thể là bộ xử lý, bộ nhớ hoặc máy chủ, và số lượng liên kết mà một nút kết nối được gọi là bậc các nút. Thiết bị chuyển mạch và kênh truyền thông đóng vai trò quan trọng trong việc hình thành cấu trúc mạng, chịu trách nhiệm truyền tải dữ liệu giữa các nút. Khi một thiết bị chuyển mạch nhận được gói dữ liệu, nó kiểm tra địa chỉ IP đích trong gói, sau đó sử dụng bảng định tuyến để xác định hướng truyền tiếp theo cho gói dữ liệu đến đích.

Các mạng kết nối có thể phân biệt qua cấu trúc liên kết, định tuyến và điều khiển luồng. Cấu trúc liên kết xác định cách các nút kết nối với nhau, trong khi thuật toán định tuyến quyết định cách thức truyền dữ liệu từ nguồn đến đích. Bên cạnh đó, kỹ thuật điều khiển luồng quản lý việc phân bổ tài nguyên để đảm bảo quá trình truyền thông diễn ra suôn sẻ và hiệu quả. Trong kiến trúc điện toán đám mây, có hai loại cấu trúc liên kết mạng cơ bản: mạng tĩnh và mạng chuyển mạch. Hai loại mạng này có cấu trúc liên kết phổ biến trong hệ thống mạng của kiến trúc điện toán đám mây, mỗi loại có đặc điểm và ứng dụng riêng.

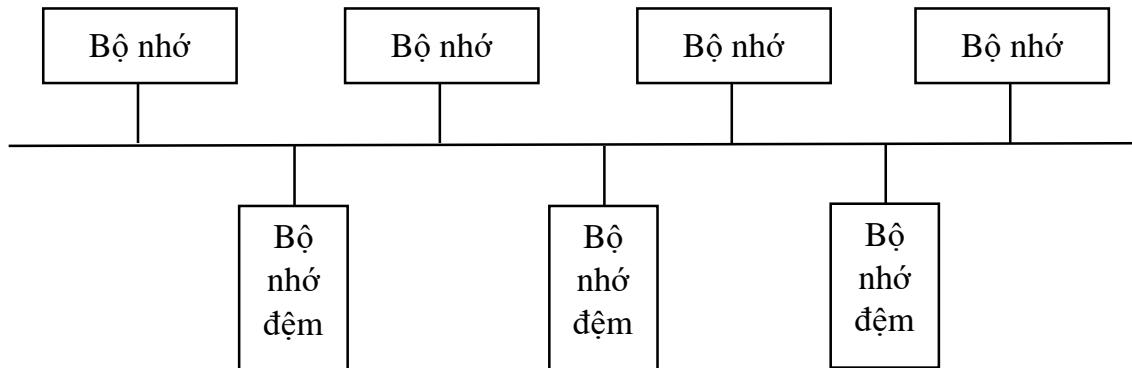
### 1.6.1. Kết nối tinh

Kết nối tinh là một loại cấu trúc liên kết trong hệ thống mạng mà các máy chủ hoặc nút được kết nối cố định với nhau bằng các liên kết vật lý. Trong kiến trúc kết nối tinh, các đường truyền giữa các nút được xác định trước và không có sự thay đổi linh hoạt theo thời gian thực. Điều này có nghĩa là khi một hệ thống sử dụng kết nối tinh, mỗi nút trong mạng có các liên kết được xác định để kết nối với những nút khác và không cần thông qua các thiết bị chuyển mạch hay các thành phần trung gian. Một số cấu hình kết nối tinh phổ biến: mạng bus, mạng siêu khồi, mạng hình lưới 2D, mạng torus.

Ưu điểm của kết nối tinh là sự đơn giản trong thiết kế và triển khai, đồng thời có độ trễ thấp khi truyền dữ liệu giữa các nút lân cận. Tuy nhiên, nhược điểm của nó là kém linh hoạt và khó mở rộng, bởi vì mỗi nút chỉ có thể kết nối với một số nút cố định. Khi quy mô hệ thống lớn hơn hoặc nhu cầu kết nối phức tạp hơn, việc duy trì các liên kết cố định sẽ trở nên khó khăn và hạn chế tính hiệu quả của mạng.

#### 1.6.1.1. Mạng bus

Mạng bus là cấu trúc đơn giản trong đó tất cả các nút được chia sẻ một đường truyền dữ liệu duy nhất. Khi một nút muốn truyền dữ liệu, nó sẽ sử dụng đường truyền này và các nút khác có thể tiếp nhận hoặc bỏ qua dữ liệu trên địa chỉ. Mạng bus có ưu điểm là dễ thiết lập và chi phí thấp, nhưng hiệu suất bị hạn chế khi có nhiều nút truyền dữ liệu đồng thời [26].



Hình 1.5: Sơ đồ mạng bus

Bên trong mạng bus mỗi thiết bị trong mạng đều kết nối trực tiếp với đường bus và chia sẻ đường truyền này để gửi và nhận dữ liệu. Khi một thiết bị muốn truyền dữ liệu, nó gửi tín hiệu vào bus và tất cả các thiết bị khác có thể 'lắng nghe' tín hiệu đó. Tuy nhiên, chỉ thiết bị có địa chỉ phù hợp với gói dữ liệu mới tiếp nhận dữ liệu; các thiết bị khác sẽ bỏ qua gói này. Do tất cả các thiết bị đều được kết nối qua một dây cáp duy

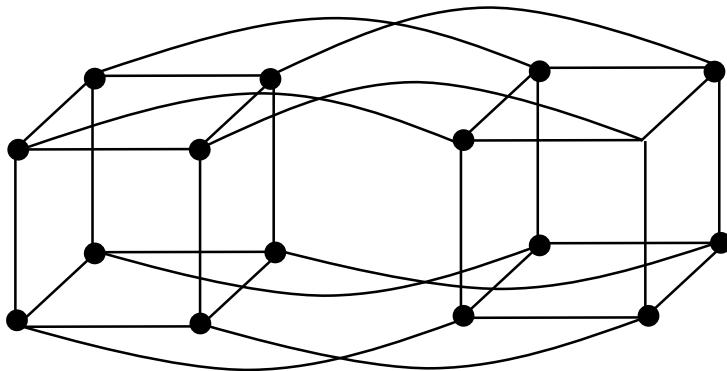
nhất, nên mạng bus có chi phí triển khai thấp và thiết kế đơn giản. Việc thêm một thiết bị mới vào mạng cũng dễ dàng, chỉ cần kết nối thiết bị đó vào mạng bus [27].

Bên cạnh những tiện ích mang lại của mạng bus thì việc tất cả các thiết bị dùng chung một đường truyền, nên khi có quá nhiều thiết bị có găng truyền dữ liệu đồng thời, có thể xảy ra xung đột. Do tất cả các thiết bị phụ thuộc vào một đường truyền duy nhất, nếu bus bị lỗi hoặc hỏng, toàn bộ mạng sẽ ngừng hoạt động. Độ dài và chất lượng của dây cáp cũng ảnh hưởng đến hiệu suất mạng, bởi vì bus dài quá sẽ làm suy giảm tín hiệu, gây ra mất dữ liệu [26].

Mạng bus thích hợp cho các hệ thống mạng nhỏ, ít thiết bị, vì khi số lượng thiết bị tăng lên, hiệu suất mạng sẽ giảm do các xung đột và độ trễ khi truyền dữ liệu.

#### 1.6.1.2. Mạng siêu khối

Mạng siêu khối là cấu trúc mạng có tính đối xứng cao, trong đó mỗi nút có một số lượng kết nối cố định với các nút lân cận trong một không gian hình học giống như khối hypercube. Mạng siêu khối có khả năng mở rộng tốt và giúp truyền dữ liệu hiệu quả giữa các nút vì khoảng cách giữa các nút được giảm thiểu [28].



Hình 1.6: Sơ đồ mạng siêu khối

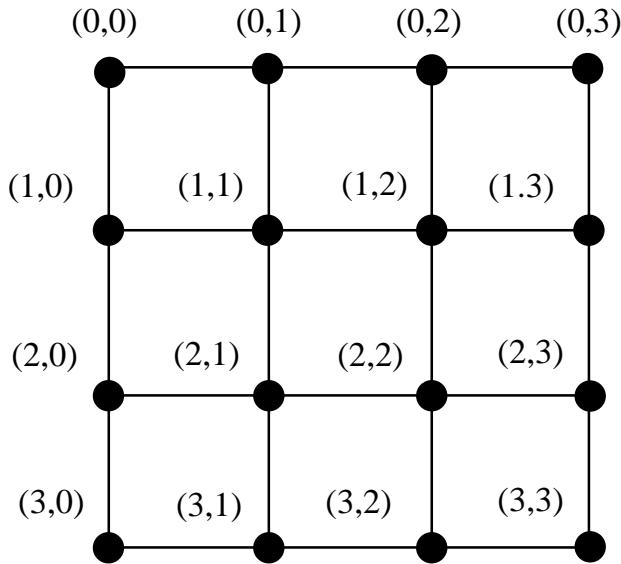
Trong mạng siêu khối, các nút được sắp xếp theo các chiều khác nhau. Mỗi nút kết nối với một số nút khác dựa trên vị trí của nó trong cấu trúc khối. Một khối siêu khối có  $n$  và  $2^n$  nút và mỗi nút được kết nối với  $n$  nút khác. Số chiều càng cao thì số lượng kết nối giữa các nút càng nhiều, tạo ra cấu trúc mạng rất đối xứng và phức tạp hơn. Do cấu trúc đặc biệt, khoảng cách tối đa giữa hai nút trong mạng siêu khối là tương đối ngắn so với số lượng nút trong mạng. Điều này có nghĩa là khi mạng có nhiều nút, thời gian để truyền dữ liệu giữa các nút vẫn ngắn hơn so với các cấu trúc mạng khác. Mạng siêu khối có tính đối xứng cao, nên các nút có vai trò và độ quan trọng như nhau trong cấu trúc mạng. Việc này giúp mạng dễ dàng mở rộng bằng cách tăng số chiều lên mà không cần

thay đổi các liên kết hiện có. Mỗi khi thêm một chiều, số lượng nút sẽ tăng lên gấp đôi, do đó mạng có khả năng mở rộng và linh hoạt và nhanh chóng [28].

Tuy nhiên do cấu trúc phức tạp, mạng siêu khối đòi hỏi chi phí xây dựng và quản lý cao hơn, đặc biệt khi số chiều tăng lên. Số lượng liên kết giữa các nút tăng nhanh khi thêm chiều mới, làm tăng độ phức tạp trong việc thiết lập và bảo trì [28].

#### 1.6.1.3. Mạng hình lưới 2D

Trong mạng hình lưới 2D, các nút được kết nối theo dạng lưới, mỗi nút kết nối với các nút lân cận của nó trong một cấu trúc hai chiều. Mạng lưới 2D được ứng dụng nhiều trong các hệ thống tính toán song song và mạng trên chip, vì nó cho phép truyền dữ liệu trực tiếp giữa các nút gần nhau và hỗ trợ phân tán dễ dàng [29].



Hình 1.7: Sơ đồ mạng hình lưới 2D

Trong mạng lưới 2D, mỗi nút (ngoài trừ các nút biên) đều có kết nối trực tiếp với các nút lân cận nằm ở bốn hướng: trên, dưới, trái, phải. Điều này giúp mỗi nút dễ dàng giao tiếp với các nút gần đó mà không cần thông qua nhiều thiết bị chuyển tiếp. Mạng hình lưới 2D dễ mở rộng bằng cách thêm các hàng hoặc cột mới, giúp tăng số lượng nút mà không cần thay đổi đáng kể các kết nối hiện tại [29].

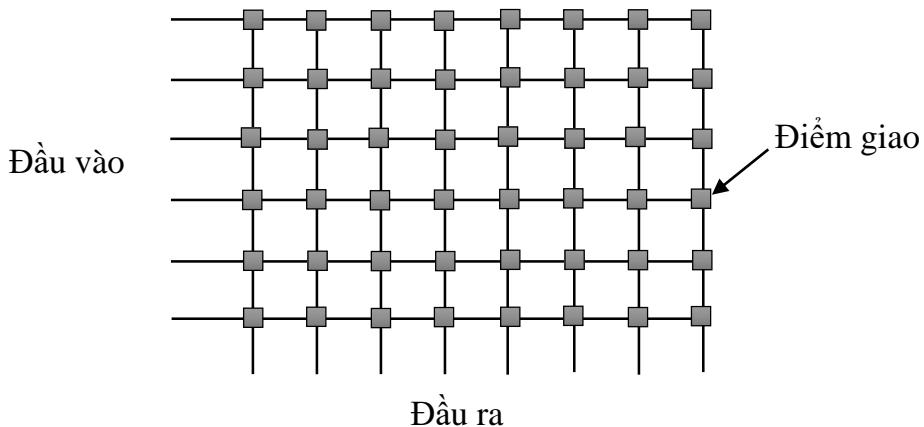
#### 1.6.2. Mạng chuyển mạch

Mạng chuyển mạch sử dụng các thiết bị chuyển mạch để kết nối với các máy chủ với nhau tạo ra hệ thống linh hoạt và dễ mở rộng hơn. Trong mạng chuyển mạch, các gói dữ liệu được chuyển tiếp từ máy chủ này đến máy chủ khác thông qua các thiết bị chuyển mạch. Thiết bị chuyển mạch sẽ xác định đường truyền tối ưu bằng cách sử dụng các bảng định tuyến, giúp dữ liệu được truyền tải hiệu quả và nhanh chóng đến đích.

Ưu điểm lớn nhất của mạng chuyển mạch là khả năng mở rộng và quản lý lưu lượng cao, phù hợp với các hệ thống mạng có quy mô lớn, nơi mà các máy chủ cần xử lý lượng dữ liệu lớn và yêu cầu khả năng chịu tải cao. Mạng chuyển mạch cũng có thể tối ưu hóa việc phân bổ tài nguyên nhờ các thuật toán định tuyến và điều khiển luồng, giúp quản lý mạng dễ dàng hơn trong các môi trường có nhiều thay đổi. Các lớp chuyển mạch như: bộ chuyển mạch Crossbar và mạng Omega.

#### 1.6.2.1. Bộ chuyển mạch Crossbar

Bộ chuyển mạch crossbar hoạt động như một ma trận kết nối giữa đầu vào và đầu ra, cho phép kết nối trực tiếp giữa từng cặp đầu vào - đầu ra thông qua các công tắc tại mỗi điểm giao. Mỗi cặp đầu vào và đầu ra có thể được nối với nhau theo yêu cầu mà không cần qua trung gian. Crossbar bao gồm các hàng (tương ứng với đầu vào) và cột (tương ứng với đầu ra), với các công tắc nằm ở mỗi giao điểm. Các công tắc này có thể bật/tắt để kiểm soát việc kết nối [30].



Hình 1.8: Bộ chuyển mạch Crossbar

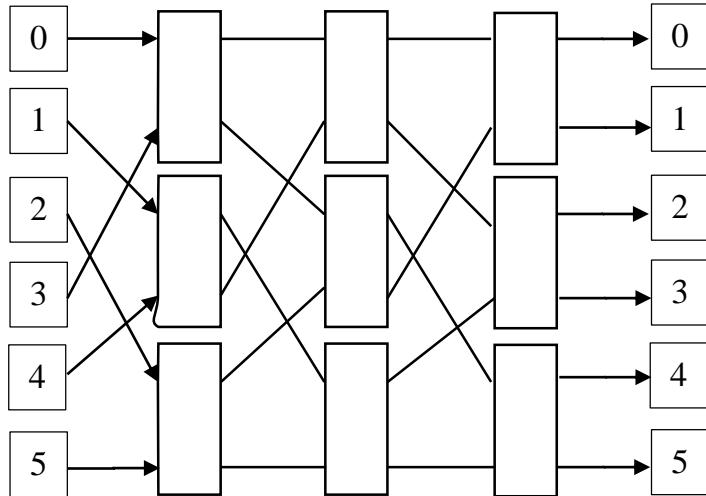
Bộ chuyển mạch Crossbar kết nối trực tiếp giúp giảm độ trễ, vì không cần truyền qua nhiều bộ chuyển mạch. Crossbar cho phép nhiều cặp đầu vào-đầu ra hoạt động đồng thời mà không ảnh hưởng đến nhau. Vì là kết nối trực tiếp nên là bộ chuyển mạch Crossbar ít phụ thuộc vào các thuật toán định tuyến phức tạp, giảm khả năng lỗi trong hệ thống [30]. Tuy nhiên do cấu trúc của bộ chuyển mạch Crossbar, mỗi giao điểm yêu cầu một công tắc, do đó số lượng công tắc tăng theo số đầu vào và đầu ra. Khi số lượng này tăng lên, chi phí phần cứng cũng tăng lên đáng kể.

Bộ chuyển mạch Crossbar thường được sử dụng ở các hệ thống yêu cầu băng thông cao và khả năng kết nối ổn định. Sử dụng trong trung tâm dữ liệu, crossbar thường được dùng để kết nối với các máy chủ với nhau hoặc các thiết bị lưu trữ. Trong hệ thống siêu

máy tính, đảm bảo được độ trễ thấp và tốc độ cao trong việc xử lý dữ liệu giữa các bộ phân khác nhau [30].

#### 1.6.2.2. Mạng Omega

Mạng Omega được tổ chức thành nhiều tầng, trong đó mỗi tầng bao gồm các bộ chuyển mạch  $2 \times 2$  (mỗi bộ chuyển mạch có hai đầu vào và hai đầu ra). Cấu trúc này cho phép kết nối mọi cặp đầu vào và đầu ra bằng cách định tuyến qua các tầng [31].



Hình 1.9: Sơ đồ mạng Omega

Tầng chuyển mạch, các tầng chuyển mạch này tạo ra các đường truyền khác nhau cho dữ liệu từ nguồn đến đích. Mỗi tầng sử dụng các bộ chuyển mạch đơn giản để định tuyến dữ liệu dựa trên địa chỉ đích. Dữ liệu đi qua mỗi tầng theo một quy tắc cố định để xác định đường đi. Điều này giúp đảm bảo rằng mọi cặp đầu vào và đầu ra đều có thể kết nối thông qua một con đường duy nhất trong mạng Omega. Mạng Omega có nhiều biến thể như Butterfly, Benes, Banyan, ... với mỗi loại có cấu trúc và khả năng định tuyến riêng để tối ưu hóa cho các mục đích cụ thể [31]:

Mạng Butterfly có cấu trúc tương tự như mạng Omega, có dạng hình cánh bướm với các bộ chuyển mạch  $2 \times 2$  ở mỗi tầng. Đặc biệt hiệu quả cho các ứng dụng yêu cầu tính song song cao, mạng Buffterfly cho phép kết nối nhanh và sử dụng tối ưu số lượng bộ chuyển mạch. Tuy nhiên nếu có xung đột trên một đường truyền, các dữ liệu khác có thể phải chờ, làm tăng độ trễ [31].

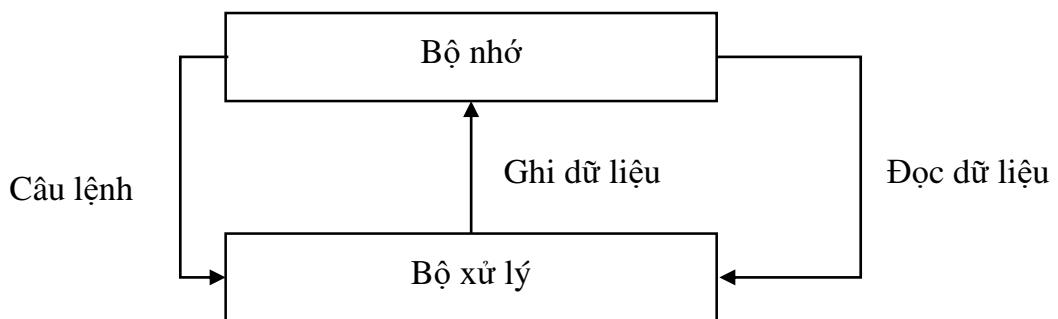
Mạng Benes có nhiều đường dẫn thay thế hơn, nhờ đó có khả năng cung cấp các đường truyền dữ liệu song song mà không xung đột. Hỗ trợ định tuyến nhiều đường truyền cùng lúc, giảm thiểu khả năng tắc nghẽn, phù hợp với các hệ thống yêu cầu độ

ổn định cao. Phức tạp hơn trong thiết kế và yêu cầu nhiều bộ chuyển mạch hơn so với mạng Butterfly [31].

Mạng Banyan có cấu trúc dạng cây với các nhánh được kết nối qua các bộ chuyển mạch, tạo ra nhiều đường dẫn từ đầu vào đến đầu ra. Với cấu trúc dạng cây giúp truyền tải dữ liệu theo nhiều tuyến, giảm độ trễ và hỗ trợ tốt cho các ứng dụng yêu cầu tốc độ cao. Do có nhiều tầng, việc bảo trì và quản lý có thể trở nên phức tạp hơn, và nếu một nhánh bị lỗi, cần có cơ chế dự phòng để tránh gián đoạn kết nối [31].

### 1.7. Kiến trúc song song

Tính toán song song (Parallel computing) là một mô hình tính toán trong đó có nhiều tác vụ hoặc quá trình được thực hiện đồng thời (song song) để giải quyết các vấn đề tính toán phức tạp một cách hiệu quả. Thay vì xử lý từng tác vụ một cách tuần tự, hệ thống sẽ phân chia công việc thành các phần nhỏ và thực hiện chúng đồng thời trên nhiều bộ xử lý hoặc máy tính [32].



*Hình 1.10: Kiến trúc song song*

Dữ liệu được phân chia thành nhiều phần nhỏ và được phân phối đến các nút khác nhau. Mỗi nút có thể xử lý một phần dữ liệu riêng biệt, từ đó giảm thiểu thời gian xử lý tổng thể. Một cách phân chia phổ biến là MapReduce. Trong đó, các tác vụ ‘Map’ sẽ xử lý các phần nhỏ của dữ liệu song song, sau đó tác vụ sẽ ‘Reduce’ sẽ tổng hợp kết quả [33].

Xử lý dữ liệu song song là một phần quan trọng của bất kỳ mô hình máy tính hiệu suất nào, sử dụng một lượng lớn tài nguyên máy tính để hoàn thành một nhiệm vụ hoặc vấn đề phức tạp. Hệ thống có thể chạy nhiều tác vụ cùng lúc trên các tài nguyên độc lập, thường là trên các máy ảo được triển khai trên nhiều vùng hoặc khu vực khác nhau trong đám mây [34].

Song song mức bit là một kỹ thuật trong xử lý song song nhằm tăng tốc độ thực hiện các phép toán số học bằng cách thao tác trên nhiều bit cùng một lúc trong một chu

kỳ xử lý, thay vì xử lý từng bit một cách tuần tự. Song song mức bit tận dụng khả năng của bộ xử lý để thực hiện nhiều thao tác cùng lúc trên nhiều bit của dữ liệu. Trong các hệ thống cũ, dữ liệu thường được xử lý từng bit một hoặc 8 bit (1 byte) một lần, dẫn đến nhiều chu kỳ xử lý.

Với kỹ thuật song song mức bit, hệ thống có thể mở rộng bề rộng của bộ xử lý (ví dụ, từ 8-bit lên 16-bit, 32-bit hoặc 64-bit) để xử lý nhiều bit đồng thời, giúp giảm số chu kỳ cần thiết cho các phép tính. Ví dụ, nếu một bộ xử lý mở rộng từ 32-bit lên 64-bit, thì các phép tính số học như cộng, trừ, nhân hoặc chia có thể thực hiện trên 64-bit dữ liệu trong một lần thay vì phải thực hiện nhiều lần cho các phần dữ liệu 32-bit.

Song song mức tác vụ là một kỹ thuật trong xử lý song song, trong đó nhiều tác vụ khác nhau được thực hiện đồng thời trên các bộ xử lý hoặc các phần cứng độc lập. Mỗi tác vụ có thể có chức năng và logic riêng biệt, và các tác vụ này có thể được thực hiện song song để tối ưu hóa thời gian hoàn thành. Mỗi tác vụ là một phần riêng biệt của bài toán tổng thể và có thể thực hiện độc lập với các tác vụ khác.

Các tác vụ này được phân bổ cho các bộ xử lý khác nhau để chạy đồng thời, nhằm giảm thời gian xử lý chung cho toàn bộ công việc. Kỹ thuật này đặc biệt hiệu quả khi các tác vụ có các chức năng khác nhau hoặc khi chúng có thể chạy độc lập mà không phải trao đổi dữ liệu thường xuyên. Ví dụ, trong một hệ thống xử lý hình ảnh, một tác vụ có thể chịu trách nhiệm phát hiện biên của ảnh trong khi một tác vụ khác xử lý độ sáng của ảnh. Hai tác vụ này có thể được thực hiện đồng thời vì chúng không phụ thuộc trực tiếp vào nhau.

Song song cấp lệnh là một kỹ thuật trong kiến trúc máy tính nhằm tối ưu hóa hiệu suất xử lý bằng cách cho phép nhiều lệnh được thực hiện đồng thời trong một chu kỳ đồng hồ. Song song cấp lệnh đạt được thông qua việc xác định và thực hiện các lệnh có thể thực hiện song song, tối ưu hóa khả năng sử dụng tài nguyên của bộ xử lý. Bộ xử lý xác định các lệnh trong một khối mã mà có thể được thực hiện song song.

Điều này thường được thực hiện thông qua các công cụ phân tích tĩnh hoặc động, phát hiện ra các lệnh không phụ thuộc vào nhau. Các lệnh có thể được tái sắp xếp để tối đa hóa việc sử dụng tài nguyên, giảm thiểu độ trễ trong quá trình thực thi. Việc tái sắp xếp này có thể thực hiện bởi trình biên dịch hoặc bộ xử lý. Sau khi các lệnh đã được phát hiện và tái sắp xếp, bộ xử lý thực hiện nhiều lệnh cùng một lúc bằng cách sử dụng nhiều đơn vị thực hiện.

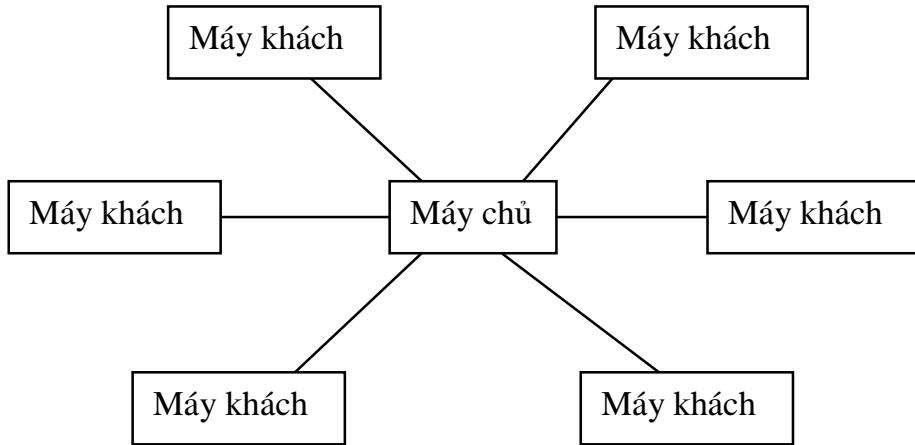
## 1.8. Kiến trúc phân tán

Điện toán đám mây là một lĩnh vực gắn bó chặt chẽ với điện toán song song và phân tán. Các ứng dụng đám mây thường dựa trên mô hình máy khách - máy chủ, trong đó phần mềm chạy trên máy khách thường rất đơn giản, chỉ cần một thiết bị mỏng để người dùng tương tác. Tất cả các phép toán phức tạp và lưu trữ dữ liệu được thực hiện và quản lý trên đám mây.

Nhiều ứng dụng đám mây xử lý một lượng lớn dữ liệu và có khả năng chạy nhiều phiên bản đồng thời. Để phối hợp các quy trình phân tán, các giao thức truyền thông đóng vai trò quan trọng trong việc xử lý thông tin qua các kênh truyền thông có thể gặp nhiều rào cản, như độ nhiễu và không đáng tin cậy. Điều này có thể dẫn đến tình trạng mất mát bản tin, gửi bản tin trùng lặp hoặc bị sai lệch thứ tự.

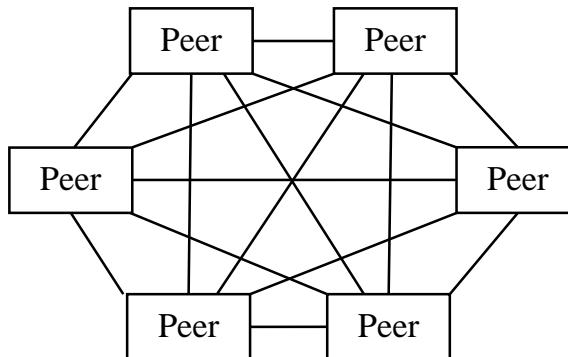
Hệ thống phân tán được định nghĩa là một mạng lưới các máy tính tự trị, chúng giao tiếp với nhau nhằm đạt được các mục tiêu chung. Trong hệ thống này, mỗi máy tính là độc lập, không chia sẻ bộ nhớ hay bộ xử lý với các máy tính khác. Mỗi máy tính có bộ nhớ cục bộ riêng và thông tin chỉ được trao đổi thông qua việc truyền tải bản tin giữa các nút thông qua các liên kết truyền thông có sẵn. Điều này tạo nên một hệ thống phân tán mạnh mẽ, có khả năng mở rộng và linh hoạt trong việc xử lý dữ liệu. Một số mô hình tổ chức hệ thống: mô hình máy khách/máy chủ, mô hình ngang hàng [35].

Mô hình máy khách/máy chủ bao gồm hai thành phần chính là máy khách và máy chủ. Máy khách gửi yêu cầu tới máy chủ để nhận dữ liệu hoặc dịch vụ. Máy chủ sẽ xử lý yêu cầu và gửi phản hồi trở lại máy khách. Máy khách thường là những thiết bị mỏng, trong khi máy chủ có thể có tài nguyên tính toán và lưu trữ lớn hơn. Mô hình này dễ dàng mở rộng bằng cách thêm nhiều máy chủ để xử lý yêu cầu từ nhiều máy khách. Sẽ có một máy chủ duy nhất cung cấp dịch vụ và nhiều máy khách giao tiếp với máy chủ để tiêu thụ sản phẩm của nó [36].



*Hình 1.11: Mô hình máy khách/máy chủ*

Mô hình Peer-to-Peer gồm nhiều nút (peer), mỗi nút trong mạng lưới có thể hoạt động như một máy khách hoặc máy chủ. Tất cả các nút đều có thể giao tiếp với nhau mà không cần một máy chủ trung tâm. Mô hình P2P thường được sử dụng trong các ứng dụng chia sẻ tệp, hệ thống giao dịch và blockchain, nơi tính linh hoạt và khả năng mở rộng rất quan trọng. Các nút gửi và nhận dữ liệu đều đóng góp một số công suất và bộ nhớ cho quá trình xử lý [37].



*Hình 1.12: Mô hình Peer-to-Peer*

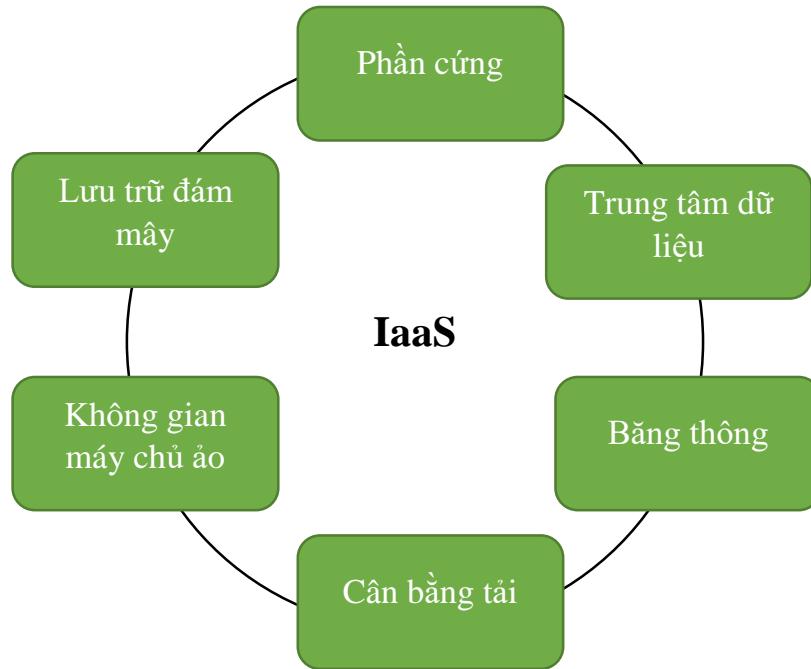
Thuộc tính cơ bản của kiến trúc phân tán là module hóa và truyền bản tin. Module hóa giúp dễ dàng thay đổi và mở rộng, mang lại cho hệ thống nhiều lợi thế và là một đặc tính của thiết kế hệ thống đầy đủ. Truyền bản tin là chức năng thể hiện sự tương tác giữa người gửi, người nhận và nội dung, bản tin cần được gửi qua mạng và có thể cần gửi nhiều loại tín hiệu khác nhau dưới dạng ‘dữ liệu’.

## 1.9. Cấu trúc điện toán đám mây

### 1.9.1. Kiến trúc tương tác IaaS

Kiến trúc tương tác IaaS bao gồm nhiều thành phần khác nhau, tạo nên một hệ sinh thái cho phép người dùng truy cập và quản lý các tài nguyên hạ tầng ảo hóa trên đám

mây. Trong kiến trúc này, người dùng, có thể là cá nhân hoặc tổ chức, tương tác với các dịch vụ thông qua giao diện người dùng hoặc API. Giao diện người dùng cung cấp cho người dùng các công cụ trực quan để quản lý và tương tác với tài nguyên, trong khi API cho phép các ứng dụng thực hiện các lệnh gọi để tạo, cấu hình và quản lý tài nguyên IaaS [38].



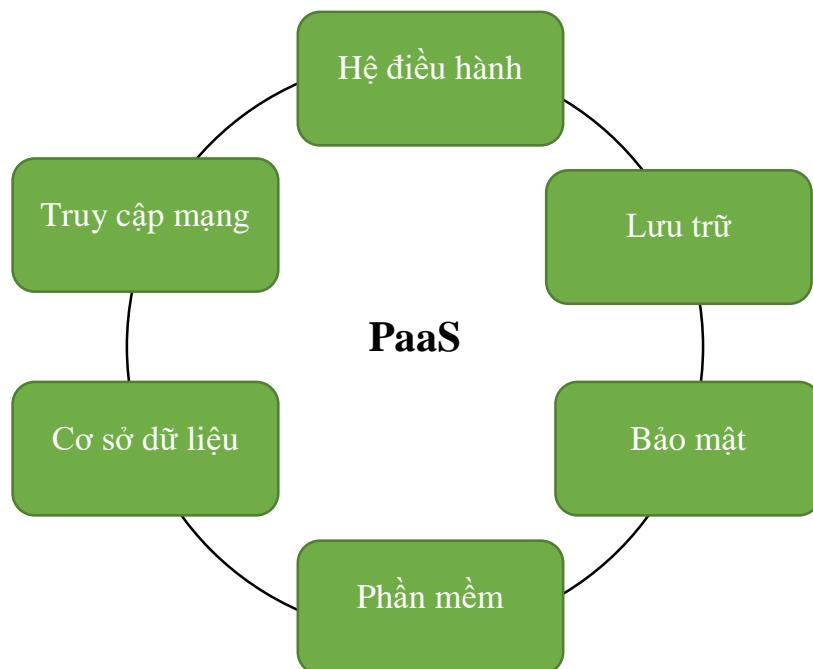
*Hình 1.13: Kiến trúc tương tác IaaS*

Các thành phần chính của kiến trúc IaaS bao gồm máy chủ ảo, lưu trữ và mạng. Máy chủ ảo là tài nguyên tính toán chính, cho phép người dùng triển khai và chạy ứng dụng. Người dùng có thể dễ dàng tạo, cấu hình và xóa máy chủ ảo theo nhu cầu của mình. Lưu trữ đám mây cung cấp khả năng lưu trữ dữ liệu trên các đĩa ảo, với nhiều tùy chọn như lưu trữ khối, lưu trữ đối tượng và lưu trữ tệp. Mạng trong IaaS là hệ thống cho phép các máy ảo và dịch vụ kết nối với nhau, bao gồm các thành phần như mạng riêng ảo, địa chỉ IP, tường lửa và bộ cân bằng tải [38].

Lợi ích của mô hình IaaS rất đa dạng. Đầu tiên, nó mang lại sự linh hoạt, cho phép người dùng mở rộng hoặc thu hẹp tài nguyên theo nhu cầu mà không cần đầu tư vào phần cứng vật lý. Thứ hai, mô hình này giúp tiết kiệm chi phí nhờ vào hình thức trả tiền theo mức sử dụng, giảm thiểu chi phí đầu tư ban đầu.Thêm vào đó, các công cụ quản lý và giám sát trong IaaS giúp người dùng dễ dàng theo dõi và tối ưu hóa tài nguyên, đồng thời hệ thống cũng có khả năng mở rộng một cách dễ dàng để đáp ứng nhu cầu tăng trưởng [38].

### 1.9.2. Kiến trúc tương tác PaaS

Kiến trúc tương tác của PaaS cung cấp một môi trường phát triển toàn diện cho các nhà phát triển ứng dụng, cho phép họ xây dựng, triển khai và quản lý ứng dụng mà không cần phải quản lý hạ tầng bên dưới. Kiến trúc này tích hợp các thành phần phần mềm và phần cứng, cung cấp một loạt các dịch vụ cần thiết để phát triển ứng dụng một cách hiệu quả [39].



Hình 1.14: Kiến trúc tương tác PaaS

Trong kiến trúc PaaS, người dùng thường là các nhà phát triển phần mềm và tổ chức, họ truy cập vào nền tảng thông qua giao diện người dùng hoặc API. Giao diện này cho phép họ dễ dàng tạo và quản lý ứng dụng, cấu hình các dịch vụ cần thiết, và theo dõi hiệu suất của ứng dụng. API cung cấp khả năng tương tác lập trình giữa các ứng dụng và dịch vụ PaaS, cho phép tự động hóa nhiều quy trình phát triển và triển khai [39].

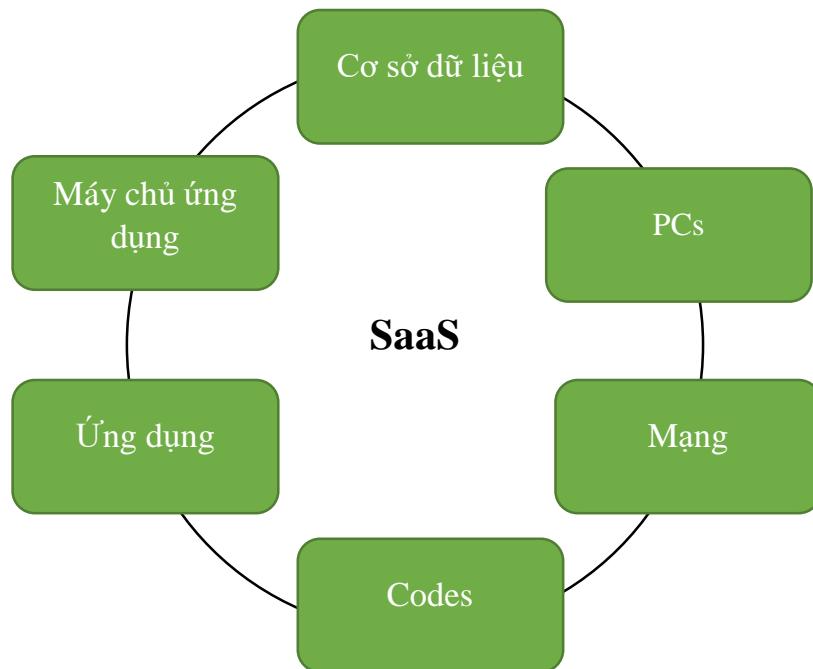
Kiến trúc PaaS bao gồm nhiều thành phần quan trọng, như môi trường phát triển tích hợp, dịch vụ lưu trữ, cơ sở dữ liệu, và các dịch vụ bổ sung khác như kiểm thử tự động và phân tích. Môi trường phát triển tích hợp cho phép các nhà phát triển viết mã, kiểm tra và gỡ lỗi ứng dụng một cách thuận tiện. Dịch vụ lưu trữ cung cấp không gian lưu trữ cho ứng dụng và dữ liệu, trong khi cơ sở dữ liệu giúp quản lý và truy xuất thông tin một cách hiệu quả [39].

Một trong những lợi ích lớn nhất của PaaS là khả năng tăng tốc quá trình phát triển ứng dụng. Bằng cách cung cấp sẵn các công cụ và dịch vụ cần thiết, PaaS giúp giảm

thiểu thời gian và công sức cần thiết để phát triển và triển khai ứng dụng. Ngoài ra, PaaS cũng cho phép các nhà phát triển tập trung vào việc xây dựng và cải thiện ứng dụng thay vì phải lo lắng về việc quản lý hạ tầng. PaaS còn hỗ trợ tính linh hoạt và mở rộng, cho phép các nhà phát triển điều chỉnh tài nguyên theo nhu cầu ứng dụng mà không phải lo lắng về việc duy trì phần cứng. Điều này cũng giúp tiết kiệm chi phí, vì nhà phát triển chỉ phải trả cho các tài nguyên và dịch vụ mà họ thực sự sử dụng [39].

### 1.9.3. Kiến trúc tương tác SaaS

Kiến trúc tương tác SaaS là mô hình cung cấp phần mềm qua Internet, cho phép người dùng truy cập và sử dụng ứng dụng mà không cần cài đặt hoặc quản lý hạ tầng phần cứng hay phần mềm. Trong mô hình này, ứng dụng được lưu trữ trên máy chủ của nhà cung cấp dịch vụ và được truy cập thông qua trình duyệt web hoặc giao diện API, mang lại sự tiện lợi và tính linh hoạt cao cho người dùng [40].



Hình 1.15: Kiến trúc tương tác SaaS

Người dùng SaaS, thường là cá nhân hoặc tổ chức, có thể dễ dàng đăng ký và truy cập vào ứng dụng qua giao diện người dùng thân thiện. Quá trình này thường bắt đầu bằng việc đăng ký tài khoản, trong đó người dùng nhập thông tin cá nhân hoặc thông tin công ty. Sau khi hoàn tất, họ sẽ nhận được quyền truy cập vào ứng dụng và có thể bắt đầu sử dụng ngay lập tức mà không cần thực hiện các bước cài đặt phức tạp [40].

Kiến trúc SaaS thường bao gồm ba thành phần chính: giao diện người dùng (UI), logic ứng dụng và hạ tầng. Giao diện người dùng thường được thiết kế trực quan và dễ sử dụng, cho phép người dùng tương tác với ứng dụng một cách thuận lợi. Logic ứng

dụng, được triển khai trên máy chủ của nhà cung cấp dịch vụ, thực hiện các chức năng xử lý dữ liệu và nghiệp vụ của ứng dụng. Hạ tầng, bao gồm máy chủ, cơ sở dữ liệu và các dịch vụ lưu trữ, là nơi lưu trữ dữ liệu và đảm bảo hiệu suất cao cho ứng dụng [40].

Một trong những đặc điểm nổi bật của SaaS là khả năng mở rộng và tính sẵn có cao. Nhà cung cấp dịch vụ có thể dễ dàng mở rộng hạ tầng để đáp ứng nhu cầu gia tăng từ người dùng mà không ảnh hưởng đến hiệu suất của ứng dụng. Các bản cập nhật và nâng cấp phần mềm cũng được thực hiện tự động, giúp người dùng luôn sử dụng phiên bản mới nhất mà không cần can thiệp [18].

Lợi ích lớn của SaaS bao gồm tiết kiệm chi phí, vì người dùng chỉ phải trả tiền cho những dịch vụ mà họ thực sự sử dụng mà không cần đầu tư vào phần cứng hay phần mềm. Ngoài ra, SaaS cũng giúp giảm thiểu thời gian và công sức cần thiết để triển khai và quản lý ứng dụng, vì mọi thứ đã được nhà cung cấp dịch vụ xử lý. Hơn nữa, tính năng hợp tác cũng được nâng cao trong môi trường SaaS, cho phép nhiều người dùng cùng truy cập và làm việc trên cùng một ứng dụng hoặc tài liệu theo thời gian thực. Điều này thúc đẩy sự tương tác và trao đổi thông tin nhanh chóng trong tổ chức.

## **1.10. Hạ tầng đám mây và ảo hóa**

### **1.10.1. Hạ tầng đám mây**

Hạ tầng đám mây bao gồm tất cả các thành phần vật lý và logic cần thiết để cung cấp dịch vụ điện toán đám mây. Điều này bao gồm máy chủ, thiết bị lưu trữ, mạng, và các thành phần phần mềm cần thiết để quản lý và điều phối các tài nguyên này. Hạ tầng đám mây có thể được triển khai theo nhiều mô hình khác nhau, như: công cộng (Public Cloud), riêng tư (Private Cloud), cộng đồng (Community Cloud).

Hạ tầng đám mây cung cấp khả năng mở rộng linh hoạt, cho phép người dùng dễ dàng điều chỉnh tài nguyên theo nhu cầu mà không phải lo lắng về việc đầu tư vào phần cứng vật lý. Điều này giúp tiết kiệm chi phí và tối ưu hóa hiệu suất.

### **1.10.2. Ảo hóa**

Ảo hóa là một công nghệ quan trọng trong điện toán đám mây, cho phép tách biệt hợp lý các tài nguyên vật lý khỏi sự truy cập trực tiếp của người dùng nhằm đáp ứng nhu cầu dịch vụ của họ. Thay vì cài đặt và quản lý tài nguyên trên phần cứng vật lý, người dùng có thể thực hiện mọi hoạt động trên một môi trường hoạt động logic mà ảo hóa tạo ra. Điều này có nghĩa là các tài nguyên điện toán vật lý như bộ xử lý, bộ nhớ và đĩa lưu trữ đều được ảo hóa để người dùng có thể truy cập và sử dụng một cách linh hoạt và hiệu quả hơn [41] [42].

Trong ảo hóa, không chỉ có các thiết bị điện toán cơ bản như bộ xử lý và bộ nhớ chính mà còn nhiều tài nguyên khác như thiết bị lưu trữ, thiết bị mạng (chẳng hạn như bộ chuyển mạch và bộ định tuyến), các liên kết giao tiếp, và các thiết bị ngoại vi (như bàn phím, chuột, và máy in) cũng có thể được ảo hóa. Quá trình ảo hóa này tạo ra một lớp trừu tượng cho các tài nguyên điện toán khác nhau, như bộ xử lý, bộ nhớ, lưu trữ và mạng [41].

Trừu tượng hóa trong điện toán được hiểu là quá trình che giấu những đặc điểm phức tạp và không thiết yếu của hệ thống, giúp người dùng tiếp cận và sử dụng hệ thống một cách đơn giản hơn. Thông qua sự trừu tượng này, người dùng có thể tập trung vào các chức năng chính mà không phải lo lắng về các chi tiết kỹ thuật phức tạp. Quá trình này thường được thực hiện qua các lớp phần mềm, trong đó lớp hệ điều hành được xem là một lớp trừu tượng.

Một trong những dạng phổ biến của ảo hóa là ảo hóa máy (hoặc ảo hóa máy chủ), trong đó tạo ra các máy ảo trên một máy chủ vật lý thực tế. Hệ thống máy chủ, nơi các máy ảo này hoạt động, được gọi là hệ thống mẹ, trong khi các máy ảo được gọi là hệ thống khách. Mỗi máy ảo chạy độc lập và có thể cài đặt hệ điều hành riêng biệt, cho phép người dùng truy cập vào phần cứng của hệ thống máy chủ và chạy các ứng dụng trong môi trường hoạt động của chính nó [42].

Để hỗ trợ hoạt động của các máy ảo, lớp ảo hóa đóng vai trò như một tập hợp các chương trình điều khiển tạo ra môi trường cho các máy ảo. Lớp này cung cấp quyền truy cập vào các tài nguyên hệ thống cho các máy ảo, đồng thời kiểm soát và giám sát việc thực hiện các máy ảo trên nó. Phần mềm quản lý này được gọi là Giám sát viên ảo (Virtual Machine Monitor).

Ưu điểm của ảo hóa trong điện toán đám mây thể hiện qua những lợi ích to lớn về hiệu quả, chi phí và quản lý hệ thống. Đầu tiên, ảo hóa giúp sử dụng hiệu quả các tài nguyên hiện có, cho phép chạy nhiều máy ảo trên cùng một máy chủ vật lý. Điều này không chỉ tối ưu hóa việc sử dụng phần cứng mà còn giảm chi phí, vì các máy ảo có thể chia sẻ tài nguyên trên một nền tảng vật lý duy nhất, từ đó giảm chi phí phần cứng cũng như chi phí vận hành tổng thể [41] [42].

Về mặt hạ tầng, ảo hóa cũng góp phần đáng kể trong việc giảm thiểu chi phí cơ sở hạ tầng điện toán, nhờ vào khả năng dùng chung tài nguyên và cắt giảm yêu cầu về phần cứng vật lý. Hơn nữa, nhờ vào các cơ chế sao lưu và phục hồi dễ dàng, ảo hóa giúp cải

thiện khả năng chịu lỗi của hệ thống và hỗ trợ bảo trì mà không làm gián đoạn hoạt động, giúp duy trì thời gian chết ở mức tối thiểu hoặc bằng không [42].

Việc quản trị hệ thống cũng trở nên đơn giản hơn với ảo hóa, bởi các quản trị viên có thể dễ dàng quản lý và mở rộng hệ thống theo nhu cầu mà không gặp khó khăn trong việc cài đặt và triển khai. Khả năng này cũng giúp duy trì, nâng cấp các ứng dụng cũ hoặc các hệ thống kế thừa mà không cần thay đổi phần cứng mới.

Ảo hóa cũng tạo điều kiện thuận lợi cho quá trình phát triển và kiểm thử ứng dụng, cho phép mô phỏng các môi trường làm việc khác nhau một cách nhanh chóng và an toàn mà không cần ảnh hưởng đến hệ thống chính. Cuối cùng, ảo hóa tăng cường bảo mật bằng cách giới hạn tác động của phần mềm độc hại. Khi một máy ảo bị ảnh hưởng, nó sẽ không gây hại đến toàn bộ hệ thống, giúp ngăn chặn sự phá hủy hoặc mất mát dữ liệu nghiêm trọng [42].

Nhược điểm của ảo hóa trong điện toán đám mây chủ yếu xoay quanh các vấn đề về độ tin cậy và hiệu suất. Đầu tiên, ảo hóa tạo ra một điểm chết duy nhất – khi một máy chủ vật lý gặp sự cố, các máy ảo chạy trên đó cũng sẽ bị ảnh hưởng, gây gián đoạn hoặc làm mất dữ liệu trên nhiều máy ảo cùng lúc. Điều này làm tăng khả năng thất bại của toàn bộ hệ thống nếu không có các giải pháp dự phòng [42].

Về mặt hiệu suất, các máy ảo có thể hoạt động kém hiệu quả hơn so với máy chủ vật lý trực tiếp, do phải chia sẻ tài nguyên hệ thống và chịu thêm chi phí xử lý từ lớp ảo hóa. Khi xảy ra lỗi hoặc khi phần mềm không hoạt động như mong muốn, việc khắc phục có thể phức tạp và tốn thời gian để xác định nguyên nhân gốc rễ, đặc biệt trong môi trường ảo hóa với nhiều lớp hệ thống [42].

Trong các mô hình điện toán đám mây, việc triển khai ảo hóa còn gặp nhiều thách thức về kỹ thuật, bao gồm sự kết nối và truyền thông giữa các thành phần trong môi trường tính toán song song và phân tán. Các hệ thống này đòi hỏi các phương pháp tối ưu để đảm bảo các thành phần có thể giao tiếp hiệu quả mà không gặp trở ngại về độ trễ hoặc mất dữ liệu.

Bên cạnh đó, việc sử dụng công nghệ ảo hóa cũng đặt ra yêu cầu về khả năng truy cập và lưu trữ dữ liệu trên đám mây một cách bảo mật và ổn định. Dù ảo hóa mang lại khả năng linh hoạt và dễ dàng mở rộng, các ứng dụng cần được tối ưu hóa để tương thích với môi trường này, nhằm đảm bảo trải nghiệm người dùng ổn định và đáp ứng nhu cầu dịch vụ một cách tốt nhất.

## **Tiêu kết chương**

Điện toán đám mây là một mô hình công nghệ mạnh mẽ đã thay đổi cách thức cung cấp và sử dụng dịch vụ công nghệ thông tin trên toàn cầu. Điện toán đám mây giúp mở ra khả năng lưu trữ, xử lý và truy cập dữ liệu từ bất kỳ đâu, mang đến những lợi ích vượt trội như sự linh hoạt, khả năng mở rộng và giảm thiểu chi phí đầu tư hạ tầng. Mặc dù điện toán đám mây mang lại nhiều ưu điểm như tối ưu hóa tài nguyên, giảm chi phí, nâng cao tính linh hoạt và bảo mật, nhưng cũng không thiếu những nhược điểm.

Tóm lại, điện toán đám mây không chỉ là xu hướng công nghệ hiện đại mà còn là nền tảng quan trọng giúp các tổ chức và doanh nghiệp tối ưu hóa hoạt động công nghệ thông tin, giảm chi phí vận hành và tăng khả năng linh hoạt trong việc cung cấp dịch vụ. Tuy nhiên, việc triển khai và quản lý các hệ thống đám mây yêu cầu phải có chiến lược rõ ràng, cùng với việc giải quyết các thách thức về hiệu suất và bảo mật để đạt được hiệu quả tối ưu.

## **CHƯƠNG 2. LUU TRỮ VÀ BẢO MẬT TRONG ĐIỆN TOÁN ĐÁM MÂY**

Trong chương này sẽ giới thiệu mô hình lưu trữ, các công nghệ lưu trữ, tính nhất quán dữ liệu trong dữ liệu. Đồng thời giới thiệu vấn đề bảo mật trong điện toán đám mây và một số giải pháp phổ biến trong bảo mật điện toán đám mây,

### **2.1. Tổng quan giải pháp mạng truy nhập**

Giải pháp mạng truy nhập đóng vai trò quan trọng trong việc cung cấp kết nối giữa người dùng cuối và hệ thống mạng lõi của nhà cung cấp dịch vụ, giúp thúc đẩy sự phát triển của các dịch vụ kỹ thuật số như thoại, dữ liệu và video. Trong bối cảnh công nghệ ngày càng phát triển, mạng truy nhập đã trở thành yếu tố thiết yếu để hiện thực hóa trải nghiệm số toàn diện, từ kết nối internet băng thông rộng cho các hộ gia đình đến dịch vụ di động tốc độ cao phục vụ các thiết bị thông minh [43].

Mạng truy nhập bao gồm nhiều loại hình, mỗi loại lại đáp ứng các nhu cầu khác nhau của người dùng và phù hợp với điều kiện địa lý, cơ sở hạ tầng của từng khu vực. Mạng truy nhập di động, với sự hiện diện của các công nghệ tiên tiến như 4G LTE và 5G NR, tạo ra kết nối không dây tốc độ cao cho các thiết bị di động và mang đến khả năng truy cập không gián đoạn trong vùng phủ sóng rộng [43] [44]. Các trạm gốc di động không chỉ mang lại sự linh hoạt cho người dùng mà còn là giải pháp lý tưởng trong việc mở rộng vùng phủ sóng ở các khu vực dân cư đông đúc.

Trong khi đó, mạng truy nhập băng rộng cố định là lựa chọn phổ biến cho các hộ gia đình và doanh nghiệp nhờ khả năng cung cấp kết nối internet ổn định với tốc độ cao. Các công nghệ như xDSL (Digital Subscriber Line) và cáp quang FTTH (Fiber to the Home) giúp cải thiện tốc độ truyền tải, đồng thời giảm thiểu hiện tượng trễ và nhiễu tín hiệu [44]. Sự phát triển của các giải pháp cáp quang đến tận nhà đã nâng cấp đáng kể chất lượng kết nối, đáp ứng nhu cầu về tốc độ truy cập cao cho các hoạt động như xem video trực tuyến, làm việc từ xa và học tập trực tuyến.

Bên cạnh mạng cố định và di động, mạng truy nhập không dây cố định đã trở thành giải pháp quan trọng tại các khu vực khó triển khai cáp, hoặc nơi người dùng cần sự linh động trong việc di chuyển thiết bị. Sử dụng sóng vô tuyến để tạo ra kết nối cố định cho các hộ gia đình hoặc văn phòng nhỏ, đồng thời đóng vai trò quan trọng trong việc giảm bớt sự phụ thuộc vào các công nghệ truyền thống [45]. Ngoài ra, các mạng truy nhập vệ tinh, đặc biệt là các vệ tinh tầm thấp như của Starlink, cung cấp khả năng kết nối internet cho các khu vực xa xôi và địa hình khó khăn, giảm độ trễ so với vệ tinh truyền thống [46] [47].

Bên cạnh những giải pháp mạng truy nhập tiên tiến thì công nghệ đĩa cứng đã trải qua nhiều thập kỷ phát triển, trở thành một trong những thành phần quan trọng trong hệ thống lưu trữ của máy tính. Trong đó hai yếu tố chính trong công nghệ đĩa cứng là dung lượng lưu trữ và hiệu suất hoạt động. Dung lượng của đĩa cứng là một yếu tố thiết yếu để đáp ứng nhu cầu lưu trữ dữ liệu của người dùng. Dung lượng được đo bằng các đơn vị như MB (Megabyte), TB (Terabyte) hoặc Gigabyte (GB), và hiện nay đã có những ổ đĩa cứng có thể đạt được mức dung lượng hàng chục Terabyte, phục vụ các ứng dụng đòi hỏi dung lượng lớn như cơ sở dữ liệu, lưu trữ video, và sao lưu hệ thống.

Ngoài việc khả năng lưu trữ thì hiệu suất của ổ đĩa cứng cũng rất quan trọng trong việc đọc ghi dữ liệu. Hiệu suất của đĩa cứng được đo chủ yếu thông qua thời gian truy cập trung bình và tốc độ truyền dữ liệu. Thời gian truy cập trung bình là thời gian cần thiết để đầu đọc ghi định vị đến vị trí của dữ liệu cần đọc trên đĩa. Đối với HDD, thời gian này dao động từ khoảng 5 đến 15 mili giây, tùy thuộc vào tốc độ quay của đĩa và độ chính xác của cơ chế đầu đọc ghi. Ngoài ra, bộ nhớ đệm cũng góp phần nâng cao hiệu suất, giúp tạm lưu dữ liệu gần nhất để giảm thời gian truy xuất [48].

Để xây dựng một giải pháp mạng truy nhập hiệu quả, các yếu tố kỹ thuật như băng thông, độ trễ, độ phủ sóng, chi phí triển khai và bảo mật đều cần được xem xét kỹ lưỡng. Băng thông lớn đảm bảo mạng có thể đáp ứng nhu cầu truyền tải dữ liệu khối lượng lớn, đặc biệt khi người dùng ngày càng phụ thuộc vào các ứng dụng như xem phim trực tuyến hay chơi game trên nền tảng đám mây. Độ trễ thấp là yếu tố không thể thiếu để mang lại trải nghiệm tốt nhất cho các ứng dụng yêu cầu thời gian thực, từ dịch vụ hội nghị truyền hình đến kết nối cho các thiết bị IoT trong môi trường công nghiệp.

## 2.2. Các công nghệ lưu trữ và mô hình lưu trữ

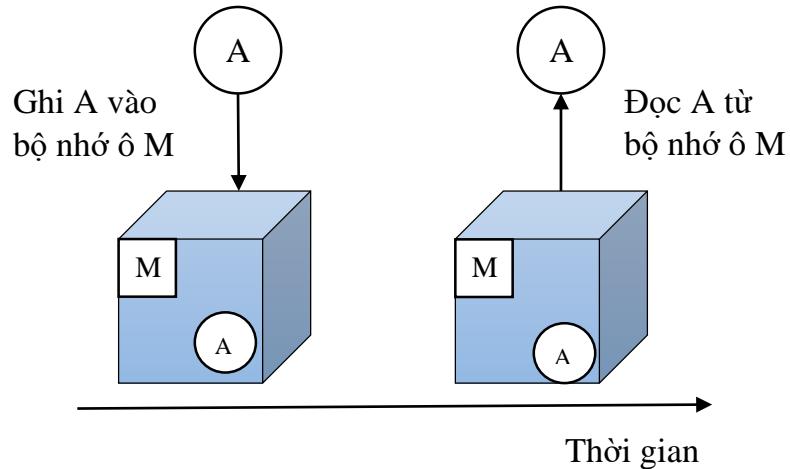
### 2.2.1. Mô hình lưu trữ

Mô hình lưu trữ trong cơ sở dữ liệu liên quan đến cách bố trí và quản lý dữ liệu trong không gian lưu trữ vật lý, giúp xác định cách dữ liệu được lưu trữ và truy cập trên hệ thống phần cứng. Trong khi đó, mô hình dữ liệu tập trung vào cách tổ chức dữ liệu dưới dạng các đối tượng hoặc bảng logic, nhằm đáp ứng các yêu cầu ứng dụng. Các mô hình lưu trữ giúp xác định cách thức thực hiện và tối ưu hóa việc lưu trữ, cập nhật và truy vấn dữ liệu trong cơ sở dữ liệu. Hai mô hình lưu trữ phổ biến là lưu trữ ô và lưu trữ nhặt ký, mỗi mô hình đều có đặc điểm và ứng dụng riêng [49].

#### 2.2.1.1. Lưu trữ ô

Mô hình lưu trữ ô thường được sử dụng trong các hệ thống cơ sở dữ liệu quan hệ, nơi dữ liệu được tổ chức thành các ô có kích thước cố định. Mỗi ô sẽ chứa một đơn vị

dữ liệu, thường là một giá trị từ một trường dữ liệu trong bảng của cơ sở dữ liệu. Dữ liệu trong mỗi ô được tổ chức sao cho có thể dễ dàng truy cập và cập nhật khi cần thiết. Đặc điểm của mô hình lưu trữ ô là mỗi đối tượng hoặc bản ghi dữ liệu sẽ nằm gọn trong một ô duy nhất. Kích thước của ô thường được xác định trước và có thể thay đổi tùy vào các đặc tính của hệ thống, như độ lớn của dữ liệu [49].



*Hình 2.1: Mô hình lưu trữ ô*

Theo hình 2.1 có thể thấy quy trình lưu trữ ô theo trình tự chia nhỏ dữ liệu thành các ô; đặt dữ liệu vào ô phù hợp và truy xuất và cập nhật dữ liệu. Quá trình chia nhỏ dữ liệu, khi có dữ liệu cần được lưu trữ, hệ thống sẽ chia dữ liệu thành các phần nhỏ và phân bổ vào các ô có kích thước cố định. Mỗi ô sẽ chứa một đơn vị dữ liệu, ví dụ như giá trị của một trường trong bảng cơ sở dữ liệu.

Quá trình đặt dữ liệu vào ô phù hợp, hệ thống sẽ lưu trữ mỗi phần dữ liệu vào một ô riêng biệt, mỗi ô sẽ có địa chỉ riêng giúp dễ dàng tìm kiếm và truy xuất dữ liệu. Quá trình truy xuất và cập nhật dữ liệu, khi người dùng hoặc ứng dụng yêu cầu truy xuất hoặc cập nhật dữ liệu, hệ thống sẽ tra cứu ô dữ liệu dựa trên chỉ mục của nó. Nếu cần thiết, dữ liệu trong ô có thể được thay đổi hoặc cập nhật [49].

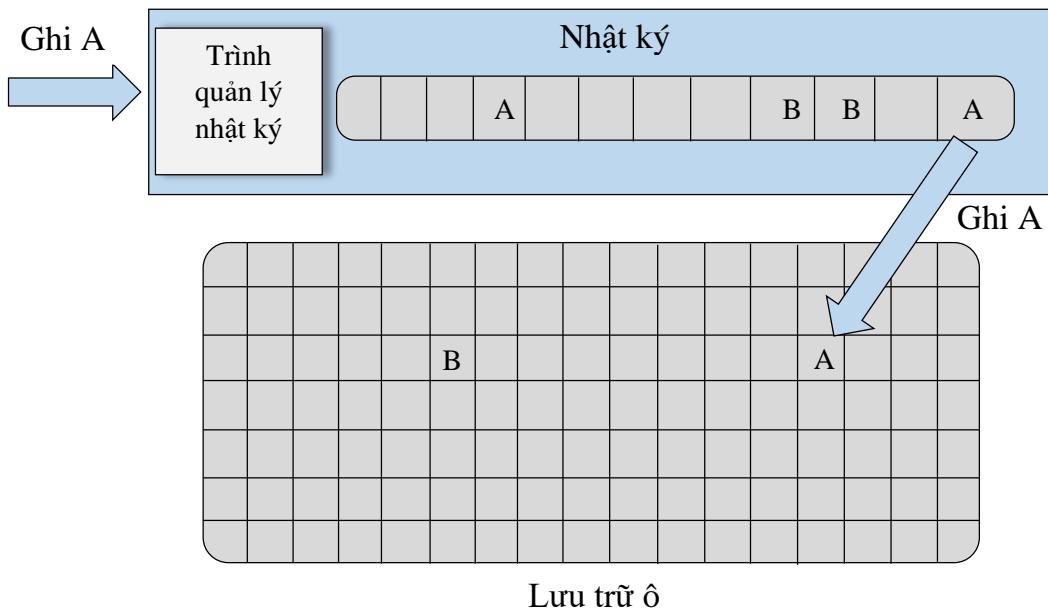
Bên cạnh những ưu điểm, mô hình lưu trữ ô cũng có một số nhược điểm. Do các ô có kích thước cố định, dữ liệu có thể bị lãng phí khi kích thước của đối tượng lưu trữ không tương thích với kích thước của ô. Do đó mô hình lưu trữ ô rất hiệu quả trong một số trường hợp nhưng không phải lúc nào cũng linh hoạt trong việc xử lý dữ liệu có kích thước không đồng đều [49].

#### 2.2.1.2. Lưu trữ nhật ký

Mô hình lưu trữ nhật ký, hay còn gọi là mô hình lưu trữ theo dạng lịch sử, là một phương pháp lưu trữ dữ liệu trong đó mỗi thay đổi về dữ liệu hoặc biến đều được ghi lại

liên tục trong một chuỗi các bản ghi (hay còn gọi là nhật ký). Mô hình này thường được sử dụng trong các hệ thống cơ sở dữ liệu ghi thêm hoặc các hệ thống quản lý giao dịch.

Lưu trữ nhật ký là một mô hình khía cạnh phức tạp để lưu trữ đổi tượng tổng hợp như các bản ghi bao gồm nhiều trường. Bộ lưu trữ nhật ký bao gồm một trình quản lý và bộ lưu trữ ô nơi chứa toàn bộ lịch sử của một biến được duy trì, thay vì chỉ giá trị hiện tại. Trình quản lý lưu trữ của hệ thống sẽ xử lý và lưu trữ các ô dữ liệu mới theo dạng chuỗi nhật ký, nơi dữ liệu thay đổi sẽ không ghi đè lên các giá trị cũ mà luôn được thêm vào theo thời gian. Việc này giúp theo dõi mọi thay đổi trong lịch sử của dữ liệu, từ đó có thể khôi phục lại trạng thái cũ nếu có sự cố xảy ra [49].



*Hình 2.2: Mô hình lưu trữ nhật ký*

Theo hình 2.2 có thấy thấy quy trình lưu trữ dữ liệu trong mô hình lưu trữ nhật ký. Ghi nhận thay đổi vào nhật ký, mỗi khi có thay đổi trên hệ thống hoặc trên cơ sở dữ liệu sẽ được ghi lại trong nhật ký dưới dạng một bản ghi đơn giản. Các thay đổi sẽ được ghi lại theo thứ tự thời gian để hệ thống có thể truy vết các thao tác từ quá khứ và phục hồi lại trạng thái hệ thống trước sự cố. Dữ liệu được lưu trữ vào trình quản lý nhật ký đồng thời sẽ được lưu trữ vào mô hình lưu trữ ô nhằm cải thiện việc tổ chức dữ liệu trong nhật ký và dễ dàng truy xuất khi cần thiết.

Với ưu điểm cho phép theo dõi và phục hồi dữ liệu một cách chính xác nhờ vào lịch sử thay đổi đầy đủ. Điều này rất hữu ích trong các hệ thống yêu cầu tính toàn vẹn và khả năng phục hồi sau sự cố. Đồng thời, quản lý dữ liệu hiệu quả trong môi trường đa người dùng, vì mỗi thay đổi đều được ghi nhận và có thể phục hồi nếu cần [49].

Bên cạnh các ưu điểm cũng có những nhược điểm đối với lưu trữ nhật ký. Lưu trữ nhật ký có thể tiêu tốn không gian lưu trữ đáng kể vì mỗi thay đổi đều được ghi lại một cách chi tiết. Việc xử lý và truy xuất dữ liệu có thể chậm hơn vì cần phải duyệt qua các bản ghi nhật ký liên quan đến thay đổi dữ liệu [49].

### 2.2.2. Công nghệ lưu trữ

Máy chủ hay những máy tính là một yếu tố quan trọng khi đề cập đến đám mây, ảo hóa và lưu trữ dữ liệu. Chúng không chỉ là những thiết bị phần cứng đơn thuần mà còn là nền tảng cho các dịch vụ và ứng dụng khác nhau, cung cấp các chức năng thiết yếu phục vụ nhu cầu của người dùng và doanh nghiệp. Máy chủ có nhiều loại, mỗi loại đảm nhận một chức năng đặc biệt trong hệ thống, tùy thuộc vào mục đích sử dụng cụ thể. Một số loại máy chủ phổ biến như máy chủ email; máy chủ cơ sở dữ liệu; máy chủ ứng dụng; máy chủ Web, ngoài ra còn có các máy chủ video/tệp và máy chủ sao lưu; máy chủ bảo mật và máy chủ mạng

Máy chủ email là thiết bị chịu trách nhiệm cho việc gửi, nhận và lưu trữ email. Những máy chủ này thường xuyên sử dụng các giao thức như SMTP để gửi thư, và POP3 hay IMAP để nhận và lưu trữ thư điện tử [50]. Trong khí đó, máy chủ cơ sở dữ liệu lại đóng vai trò quan trọng trong việc lưu trữ và quản lý các dữ liệu quan trọng của hệ thống, như thông tin khách hàng, dữ liệu giao dịch và các thông tin khác [51]. Máy chủ cơ sở dữ liệu giúp đảm bảo rằng dữ liệu có thể được truy xuất nhanh chóng và hiệu quả, nhờ vào các phần mềm quản lý cơ sở dữ liệu như MySQL, Oracle hay PostgreSQL.

Máy chủ ứng dụng chuyên việc chạy các ứng dụng phần mềm mà người dùng truy cập qua mạng, cung cấp các dịch vụ và tính năng đặc thù [52]. Các ứng dụng này có thể là phần mềm doanh nghiệp, các ứng dụng web hoặc các hệ thống quản lý. Một loại máy chủ khác tương tự là máy chủ web, có vai trò phân phối các trang web và ứng dụng web tới người dùng qua internet [53]. Các máy chủ web, sử dụng các phần mềm như Apache HTTP Server hay Nginx, giúp duy trì hoạt động ổn định của các dịch vụ web và đảm bảo khả năng phục vụ hàng triệu yêu cầu từ người dùng [53] .

Máy chủ video/tệp và máy chủ sao lưu, đóng vai trò lưu trữ và phân phối các tệp dữ liệu lớn, bao gồm video, tài liệu, hình ảnh và các tệp khác. Các máy chủ này có dung lượng lưu trữ lớn và tốc độ truy xuất nhanh, giúp người dùng có thể chia sẻ tệp một cách hiệu quả [54]. Máy chủ sao lưu, mặt khác, giúp bảo vệ dữ liệu khỏi các sự cố không lường trước như hỏng hóc phần cứng hay mất dữ liệu, đảm bảo việc khôi phục dữ liệu khi cần thiết.

Máy chủ bảo mật và máy chủ mạng cũng đóng vai trò quan trọng trong việc bảo vệ hệ thống và duy trì kết nối giữa các thiết bị trong mạng. Máy chủ bảo mật cung cấp các dịch vụ như tường lửa, mã hóa và xác thực, giúp bảo vệ hệ thống khỏi các mối đe dọa từ bên ngoài, trong khi máy chủ mạng giúp quản lý và duy trì các kết nối mạng, đảm bảo rằng tất cả các thiết bị trong hệ thống có thể giao tiếp với nhau một cách ổn định.

#### 2.2.2.1. *Kiến trúc máy chủ*

Các máy chủ, dù có nhiều loại và vai trò khác nhau, đều có một kiến trúc phần cứng chung cơ bản, gồm các thành phần chủ yếu như bộ xử lý trung tâm, bộ nhớ, các bus nội bộ/chip giao tiếp, và các cổng I/O để giao tiếp với các hệ thống bên ngoài, ví dụ như mạng và thiết bị lưu trữ. Kiến trúc này giúp máy chủ có thể xử lý các tác vụ tính toán, lưu trữ và truyền dữ liệu hiệu quả trong các môi trường công nghệ phức tạp [55].

Bộ xử lý trung tâm (CPU) là ‘bộ não’ của máy chủ, chịu trách nhiệm thực hiện các lệnh và xử lý dữ liệu. CPU có vai trò quyết định tốc độ và hiệu suất tổng thể của máy chủ. Các máy chủ hiện đại thường sử dụng các CPU đa lõi với khả năng xử lý song song, giúp tăng cường khả năng xử lý tác vụ phức tạp như quản lý cơ sở dữ liệu, ứng dụng đám mây, hoặc các quy trình tính toán cao cấp. Các máy chủ cũng có thể sử dụng các bộ xử lý chuyên dụng như GPU (đơn vị xử lý đồ họa) trong các ứng dụng đòi hỏi tính toán cao về đồ họa hoặc học máy [56].

Bộ nhớ trong máy chủ bao gồm bộ nhớ RAM (Random Access Memory) và bộ nhớ lưu trữ (storage). Bộ nhớ RAM giúp lưu trữ tạm thời các dữ liệu và chương trình mà CPU cần truy xuất nhanh chóng trong quá trình hoạt động. Các máy chủ thường có dung lượng RAM rất lớn, giúp chạy nhiều ứng dụng và xử lý các lượng dữ liệu khổng lồ mà không bị gián đoạn. Bên cạnh đó, bộ nhớ lưu trữ giúp máy chủ lưu trữ dữ liệu dài hạn, với các hệ thống ổ đĩa cứng (HDD) hoặc ổ đĩa trạng thái rắn (SSD), hỗ trợ truy xuất dữ liệu nhanh chóng và an toàn [57].

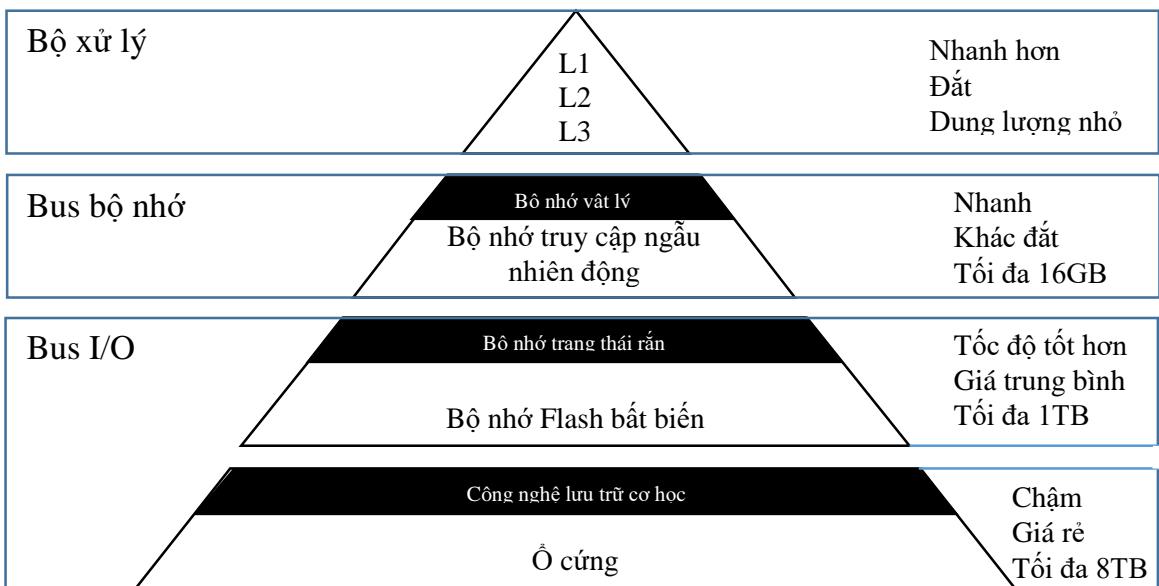
Bus nội bộ và các chip giao tiếp đóng vai trò kết nối các thành phần phần cứng khác nhau trong máy chủ. Bus nội bộ giúp truyền tải dữ liệu giữa CPU, bộ nhớ, và các thiết bị ngoại vi. Các chip giao tiếp giúp đảm bảo các thành phần này có thể giao tiếp hiệu quả với nhau, từ việc điều khiển dữ liệu tới các thiết bị lưu trữ cho đến việc truyền nhận tín hiệu giữa các thành phần của máy chủ [56].

Các cổng I/O là các giao diện vật lý giúp máy chủ kết nối với thế giới bên ngoài, bao gồm các thiết bị lưu trữ (như ổ đĩa, thẻ nhớ) và mạng (như Ethernet, Wi-Fi, hoặc cổng USB). Các cổng I/O giúp máy chủ nhận và gửi dữ liệu qua mạng hoặc đến các thiết bị lưu trữ bên ngoài. Chúng có thể bao gồm các cổng USB, các cổng mạng, cổng

video, cổng âm thanh và các cổng khác để hỗ trợ kết nối máy chủ với các thiết bị ngoại vi và mạng [56].

#### 2.2.2.2. Hệ thống phân cấp lưu trữ

Hệ thống phân cấp lưu trữ là một cách tổ chức các phương thức lưu trữ dữ liệu dựa trên tốc độ, chi phí, dung lượng và khả năng chia sẻ. Hệ thống này thường được mô tả dưới dạng một kim tự tháp, với các lớp lưu trữ có đặc điểm từ nhanh nhất và đắt nhất ở trên cùng, đến chậm nhất và có chi phí thấp nhất ở dưới cùng. Mô hình này phản ánh cách thức phân bổ tài nguyên bộ nhớ trong hệ thống máy tính, từ các bộ nhớ gần CPU nhất đến các hệ thống lưu trữ có dung lượng lớn và chi phí thấp hơn.



Hình 2.3: Hệ thống phân cấp lưu trữ

Ở đỉnh kim tự tháp, là bộ nhớ nhanh nhất và đắt nhất, bao gồm các loại bộ nhớ như bộ nhớ cache L1, L2 và L3, được tích hợp trực tiếp vào lõi của bộ xử lý (CPU). Đây là các bộ nhớ có độ trễ thấp nhất, phục vụ cho việc lưu trữ các dữ liệu tạm thời mà CPU cần xử lý ngay lập tức. Tuy nhiên, vì chi phí sản xuất cao và dung lượng hạn chế, các bộ nhớ này không thể chia sẻ rộng rãi giữa các máy chủ hay bộ xử lý khác mà không gây tốn kém đáng kể.

Dưới lớp bộ nhớ cache là bộ nhớ chính hoặc bộ nhớ truy cập ngẫu nhiên (RAM). Đây là bộ nhớ có dung lượng lớn hơn cache, giúp lưu trữ tạm thời các chương trình và dữ liệu đang được sử dụng. RAM có tốc độ truy cập nhanh nhưng không thể sánh bằng cache, và có thể được chia sẻ giữa các bộ xử lý trong hệ thống.

Lớp cuối cùng trong hệ thống phân cấp lưu trữ là hệ thống lưu trữ đầu vào/đầu ra như ổ đĩa cứng (HDD) và ổ đĩa trạng thái rắn (SSD). Dù tốc độ truy cập của các thiết bị

lưu trữ này chậm hơn nhiều so với RAM và cache, chúng cung cấp dung lượng lưu trữ lớn với chi phí hợp lý. Các hệ thống lưu trữ này phù hợp để lưu trữ dữ liệu dài hạn hoặc dữ liệu ít khi truy cập.

### **2.3. Hệ thống lưu trữ File**

#### **2.3.1. Nguyên tắc truy nhập**

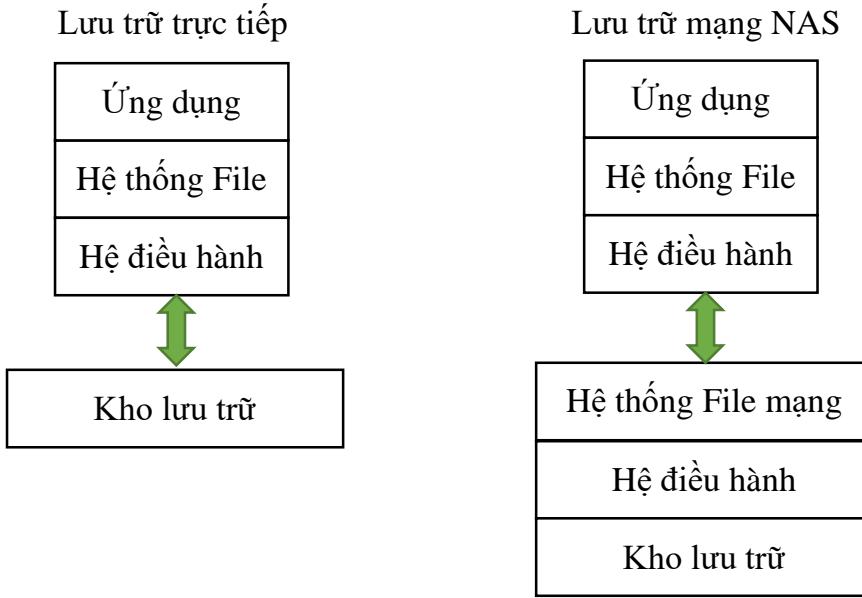
Nguyên tắc truy nhập hệ thống lưu trữ file nhằm đảm bảo rằng dữ liệu được ghi và truy cập một cách an toàn, hiệu quả, và có tổ chức trên các thiết bị lưu trữ. Khi một ứng dụng tạo ra một file, file đó sẽ được lưu vào đĩa theo vị trí được chỉ định bởi hệ thống file, đảm bảo rằng dữ liệu được ghi vào đúng nơi. Hệ điều hành và hệ thống file sẽ hợp tác để duy trì cấu trúc thư mục, nơi file được tổ chức theo các đường dẫn để người dùng và ứng dụng có thể dễ dàng truy cập lại [58].

Trong quá trình truy xuất dữ liệu, hệ thống lưu trữ sử dụng các địa chỉ vật lý và logic. Địa chỉ vật lý chỉ vị trí thực tế của dữ liệu trên ổ đĩa, trong khi địa chỉ logic giúp ứng dụng truy cập dữ liệu một cách trừu tượng hóa mà không cần quan tâm đến vị trí vật lý. Khi truy cập file, hệ thống sẽ ánh xạ các địa chỉ logic đến vị trí vật lý để đảm bảo dữ liệu được truy xuất đúng cách [58].

Một số hệ thống file còn hỗ trợ siêu dữ liệu, cung cấp thông tin mô tả về dữ liệu lưu trữ, giúp tăng hiệu quả quản lý và truy xuất dữ liệu. Hơn nữa, hệ thống lưu trữ có thể được tối ưu hóa dựa trên mục đích sử dụng, đảm bảo rằng dữ liệu được truy cập nhanh chóng cho các hoạt động trực tuyến hoặc các nhiệm vụ quan trọng. Các nguyên tắc này giúp duy trì tính ổn định và hiệu quả của hệ thống lưu trữ file, đáp ứng nhu cầu sử dụng đa dạng của người dùng và ứng dụng [58].

#### **2.3.2. Truy nhập File**

Truy nhập file là một phương pháp giúp đơn giản hóa quá trình truy cập dữ liệu bằng cách trừu tượng hóa các thành phần ở cấp khồi, cho phép dữ liệu được truy cập dễ dàng qua tên file thay vì phải biết vị trí vật lý cụ thể của dữ liệu. Sự trừu tượng hóa này giúp người dùng và ứng dụng dễ dàng thao tác với các file mà không cần quan tâm đến các chi tiết phức tạp của hệ thống lưu trữ bên dưới. Các phần mềm và giải pháp truy cập file thường bao gồm các hệ thống file có mục đích chung hoặc các hệ thống file chuyên biệt, đáp ứng cho các yêu cầu lưu trữ và truy xuất dữ liệu cụ thể của từng ứng dụng [59].



Hình 2.4: Ví dụ truy nhập File

Trong hệ thống lưu trữ hiện đại, lưu trữ đối tượng nổi lên như một phương pháp hiệu quả cho việc quản lý dữ liệu lớn và phân tán. Lưu trữ đối tượng lưu trữ dữ liệu dưới dạng các đối tượng độc lập, trong đó mỗi đối tượng chứa toàn bộ dữ liệu cùng với các thông tin mô tả về dữ liệu đó. Mỗi đối tượng được định danh bởi một ứng dụng hoặc thực thể gắn liền với dữ liệu, giúp định danh duy nhất cho từng đối tượng trong hệ thống. Không giống như hệ thống file truyền thống hoặc cơ sở dữ liệu, lưu trữ đối tượng không phụ thuộc vào bất kỳ cấu trúc tổ chức nào như thư mục hoặc bảng, mà quản lý dữ liệu dựa trên các đối tượng, tạo ra sự linh hoạt và hiệu quả trong quản lý dữ liệu phân tán và khối lượng lớn.

## 2.4. Hệ thống lưu trữ phân tán và tính nhất quán của bộ nhớ NFS và AFS

### 2.4.1. Hệ thống lưu trữ phân tán

Hệ thống lưu trữ phân tán là một mô hình lưu trữ trong đó dữ liệu được phân phối và lưu trữ trên nhiều nút máy chủ khác nhau, có thể nằm ở các vị trí địa lý khác nhau. Mục tiêu của hệ thống này là đảm bảo khả năng mở rộng, tính khả dụng cao và độ bền của dữ liệu, đặc biệt là khi xử lý khối lượng dữ liệu lớn hoặc khi dữ liệu cần phải được truy xuất từ nhiều địa điểm khác nhau [60]. Trong một hệ thống lưu trữ phân tán, dữ liệu không được lưu trữ ở một vị trí duy nhất mà được phân chia và sao chép qua nhiều nút khác nhau, giúp tối ưu hóa hiệu suất và giảm tải cho mỗi nút riêng lẻ.

Một trong những đặc điểm quan trọng của hệ thống lưu trữ phân tán là việc sử dụng kỹ thuật phân mảnh và sao chép dữ liệu. Phân mảnh dữ liệu giúp chia nhỏ thông tin thành các phần (mảnh), mỗi phần sẽ được lưu trữ ở các máy chủ khác nhau, giúp

giảm bớt tải cho một máy chủ duy nhất và tối ưu hóa quá trình xử lý dữ liệu. Sao chép dữ liệu, trong khi đó, giúp tăng tính sẵn sàng và độ tin cậy của hệ thống. Khi dữ liệu được sao chép tại nhiều nút khác nhau, hệ thống có thể đảm bảo rằng, ngay cả khi một nút gặp sự cố, dữ liệu vẫn có thể truy cập được từ các nút còn lại [61].

Tuy nhiên, một trong những thách thức lớn của hệ thống lưu trữ phân tán là đảm bảo tính nhất quán và đồng bộ dữ liệu giữa các nút. Khi dữ liệu được thay đổi ở một nút, hệ thống phải đảm bảo rằng sự thay đổi này được cập nhật đồng bộ ở tất cả các nút sao chép khác để tránh việc có dữ liệu không nhất quán. Các giao thức như Paxos hoặc Raft thường được sử dụng để quản lý việc đồng bộ hóa dữ liệu và duy trì tính nhất quán trong toàn hệ thống [61].

Một ưu điểm quan trọng của hệ thống lưu trữ phân tán là khả năng mở rộng dễ dàng. Khi nhu cầu về lưu trữ dữ liệu tăng lên, các nút mới có thể được thêm vào hệ thống mà không làm gián đoạn hoạt động của các nút đã có. Điều này giúp hệ thống có thể tiếp tục xử lý một lượng dữ liệu ngày càng lớn mà không gặp phải vấn đề về hiệu suất [61].

#### **2.4.2. Bộ nhớ NFS**

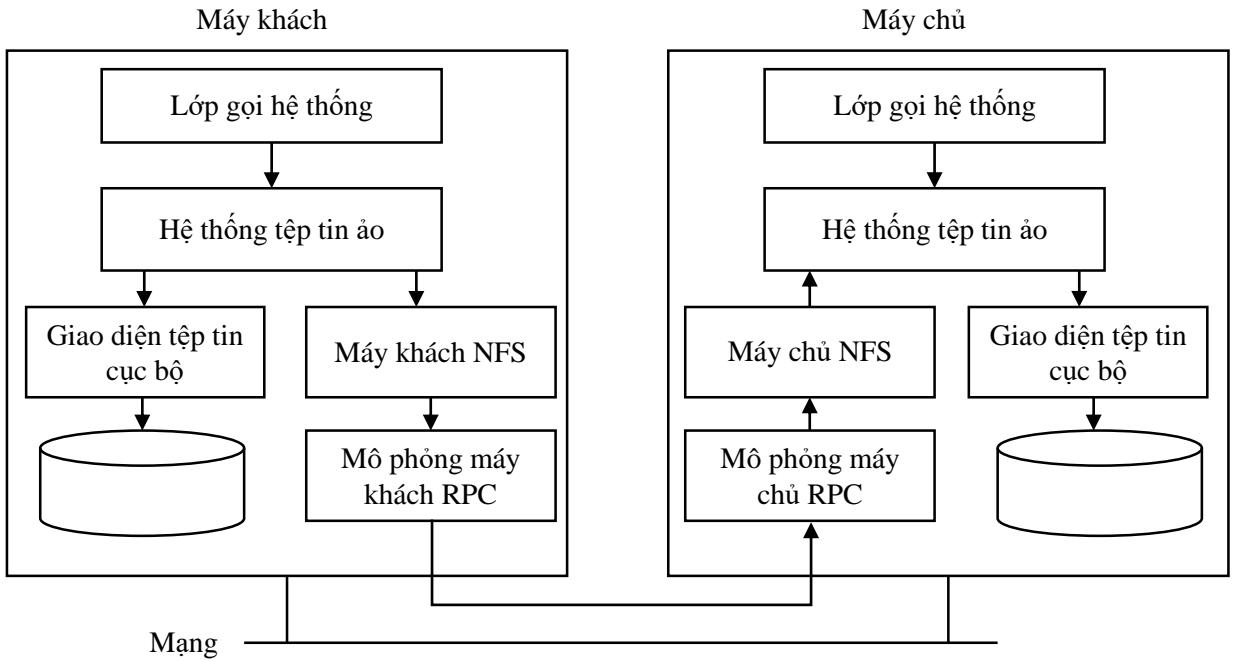
##### *2.4.2.1. Khái niệm*

NFS là một giao thức chia sẻ tệp tin qua mạng, cho phép các máy tính trong một mạng máy tính có thể truy cập và sử dụng tệp tin trên các máy tính khác giống như khi truy cập tệp tin cục bộ. Được phát triển bởi Sun Microsystems vào những năm 1980, NFS sử dụng mô hình khách - chủ, trong đó máy chủ lưu trữ tệp tin và chia sẻ chúng, còn máy khách có thể truy cập các tệp tin đó qua mạng [62].

NFS hoạt động dựa trên giao thức TCP/IP, cho phép các hệ thống khác nhau, thường là hệ điều hành Unix hoặc Linux, có thể chia sẻ tệp tin giữa chúng mà không cần phải sao chép dữ liệu vào ổ đĩa cục bộ của máy khách. Hệ thống này giúp giảm thiểu sự trùng lặp dữ liệu và tạo điều kiện thuận lợi cho việc chia sẻ tài nguyên trong môi trường mạng [62].

##### *2.4.2.2. Kiến trúc NFS*

Kiến trúc của NFS dựa trên mô hình máy khách – chủ, trong đó các máy tính có thể truy cập tệp tin lưu trữ trên các máy chủ qua mạng. Kiến trúc này được xây dựng dựa trên giao thức TCP/IP.



*Hình 2.5: Kiến trúc NFS*

Máy chủ NFS chịu trách nhiệm lưu trữ các tệp tin và chia sẻ chúng với các máy khách. Các thư mục cần chia sẻ được "xuất khẩu" từ máy chủ, và quyền truy cập vào các thư mục này được quản lý dựa trên các quy tắc đã được thiết lập sẵn. Các máy khách sẽ sử dụng các lệnh hoặc API NFS để yêu cầu máy chủ cung cấp tệp tin, đồng thời xử lý dữ liệu như thể nó là tệp tin cục bộ [63].

NFS sử dụng giao thức RPC để thực hiện các yêu cầu từ máy khách tới máy chủ, giúp việc giao tiếp trở nên minh bạch mà không cần quan tâm đến chi tiết phần cứng hay hệ điều hành của mỗi bên. Khi một máy khách yêu cầu thao tác với tệp tin, như đọc hoặc ghi, một yêu cầu RPC được gửi tới máy chủ, và máy chủ xử lý yêu cầu đó rồi trả về kết quả cho máy khách.

Trên máy chủ, có tiến trình NFS chịu trách nhiệm xử lý các yêu cầu từ máy khách. Máy khách không trực tiếp tương tác với hệ thống tệp của máy chủ mà thông qua dịch vụ NFS daemon. Các thư mục được chia sẻ sẽ được cấu hình rõ ràng và cấp quyền cho các máy khách cụ thể. Máy khách có thể sử dụng các thư mục này như thể chúng là thư mục cục bộ của mình [63].

Để cải thiện hiệu suất, NFS cũng cho phép lưu trữ tạm dữ liệu từ các tệp tin đã truy cập. Điều này giúp giảm tải mạng và tăng tốc độ truy xuất dữ liệu khi các tệp tin đã được yêu cầu trước đó. Tuy nhiên, việc đồng bộ dữ liệu giữa máy chủ và máy khách khi sử dụng cache cần được quản lý cẩn thận để tránh sự không nhất quán.

#### *2.4.2.3. Tính nhất quán*

Tính nhất quán của bộ nhớ trong NFS là vấn đề quan trọng khi nhiều máy khách cùng truy cập và thao tác trên cùng một tệp tin, đặc biệt trong các môi trường mạng phân tán. NFS sử dụng cơ chế lưu trữ tạm để cải thiện hiệu suất, nhưng điều này có thể gây ra sự không đồng bộ giữa các máy khách khi một máy khách thay đổi tệp tin trong khi các máy khách khác đang giữ bản sao trong bộ nhớ cache. Khi các thay đổi trên tệp tin không được đồng bộ kịp thời, các máy khách có thể làm việc với dữ liệu lỗi thời, dẫn đến sự không nhất quán.

Để giải quyết vấn đề này, NFS sử dụng một số cơ chế đồng bộ như attribute caching và file locking. File locking giúp ngăn chặn các máy khách sửa đổi tệp tin đồng thời, từ đó giảm nguy cơ xung đột và mất dữ liệu. Tuy nhiên, cơ chế khóa tệp này không phải lúc nào cũng được triển khai, và điều này có thể làm giảm hiệu quả trong việc đảm bảo tính nhất quán. Bên cạnh đó, NFSv4 và các phiên bản mới hơn đã cải thiện tính nhất quán bằng cách sử dụng các cơ chế như delegation và callback, cho phép máy khách nhận thông báo ngay lập tức về các thay đổi trên tệp tin từ máy chủ, giúp giảm tình trạng các máy khách làm việc với dữ liệu cũ.

#### *2.4.2.4. Ưu điểm và nhược điểm*

NFS có nhiều ưu điểm như: Dễ triển khai và cấu hình, NFS rất dễ triển khai trong các môi trường sử dụng hệ điều hành Unix/Linux, với cấu hình đơn giản và không yêu cầu phần mềm hay phần cứng phức tạp. Việc chia sẻ tệp tin chỉ cần một số cấu hình cơ bản trên máy chủ và máy khách. Tính tương thích cao, NFS hỗ trợ nhiều hệ điều hành khác nhau, chủ yếu là Unix/Linux, nhưng cũng có thể hoạt động trên các hệ điều hành khác như Windows và macOS thông qua các phần mềm bổ sung. Điều này giúp nó trở thành lựa chọn phổ biến trong các môi trường đa hệ điều hành.

Hiệu suất tốt với bộ nhớ cache, NFS hỗ trợ cơ chế lưu trữ tạm trên các máy khách, giúp giảm độ trễ và tăng hiệu suất khi truy cập tệp tin. Điều này đặc biệt hữu ích trong các hệ thống yêu cầu truy cập tệp tin nhiều lần. Chi phí thấp, NFS là một giao thức mã nguồn mở và miễn phí, giúp tiết kiệm chi phí phần mềm so với các giải pháp chia sẻ tệp tin khác, đặc biệt trong các hệ thống lớn với nhiều máy chủ và máy khách. Quản lý tập trung, các tệp tin được lưu trữ trên máy chủ, điều này giúp đơn giản hóa việc sao lưu, bảo mật và quản lý dữ liệu, vì tất cả dữ liệu đều được tập trung ở một nơi [62].

Bên cạnh những ưu điểm thì NFS cũng có những nhược điểm như: Vấn đề về bảo mật, NFS có thể gặp vấn đề về bảo mật, đặc biệt là trong các môi trường không tin cậy. Mặc dù có thể sử dụng các cơ chế bảo mật như Kerberos, việc cấu hình và triển khai

bảo mật đúng cách không phải lúc nào cũng đơn giản. Nếu không được bảo mật đúng cách, các máy khách có thể dễ dàng truy cập vào dữ liệu trên máy chủ. Khả năng đồng bộ kém, do cơ chế caching và thiếu cơ chế đồng bộ chặt chẽ, NFS có thể gặp vấn đề về tính nhất quán của dữ liệu khi nhiều máy khách cùng truy cập và thay đổi tệp tin cùng lúc. Điều này có thể dẫn đến sự không đồng bộ giữa các máy khách và máy chủ, gây ra các lỗi hoặc xung đột dữ liệu.

Hiệu suất giảm trong môi trường mạng không ổn định, trong các mạng có độ trễ cao hoặc băng thông thấp, hiệu suất của NFS có thể bị ảnh hưởng. Việc truyền tải dữ liệu qua mạng có thể làm giảm tốc độ truy cập tệp tin, đặc biệt là đối với các hệ thống yêu cầu truy cập tệp tin tốc độ cao. Quản lý khóa tệp tin phức tạp, mặc dù NFS hỗ trợ cơ chế khóa tệp tin (file locking) để tránh các vấn đề xung đột khi nhiều máy khách cùng chỉnh sửa một tệp tin, nhưng việc triển khai và quản lý khóa tệp tin có thể trở nên phức tạp và không luôn luôn hiệu quả trong môi trường có nhiều máy khách [62].

### 2.4.3. Bộ nhớ AFS

#### 2.4.3.1. Khái niệm

AFS là một hệ thống tệp phân tán được phát triển tại Đại học Carnegie Mellon vào những năm 1980, nhằm mục đích cung cấp khả năng chia sẻ tệp tin hiệu quả trong các mạng lớn và phân tán. Hệ thống này cho phép các máy tính trong mạng truy cập và thao tác với các tệp tin lưu trữ trên các máy chủ từ xa, nhưng người dùng không cần phải quan tâm đến vị trí vật lý của dữ liệu, vì tệp tin được quản lý và truy cập như thể chúng là tệp tin cục bộ [64].

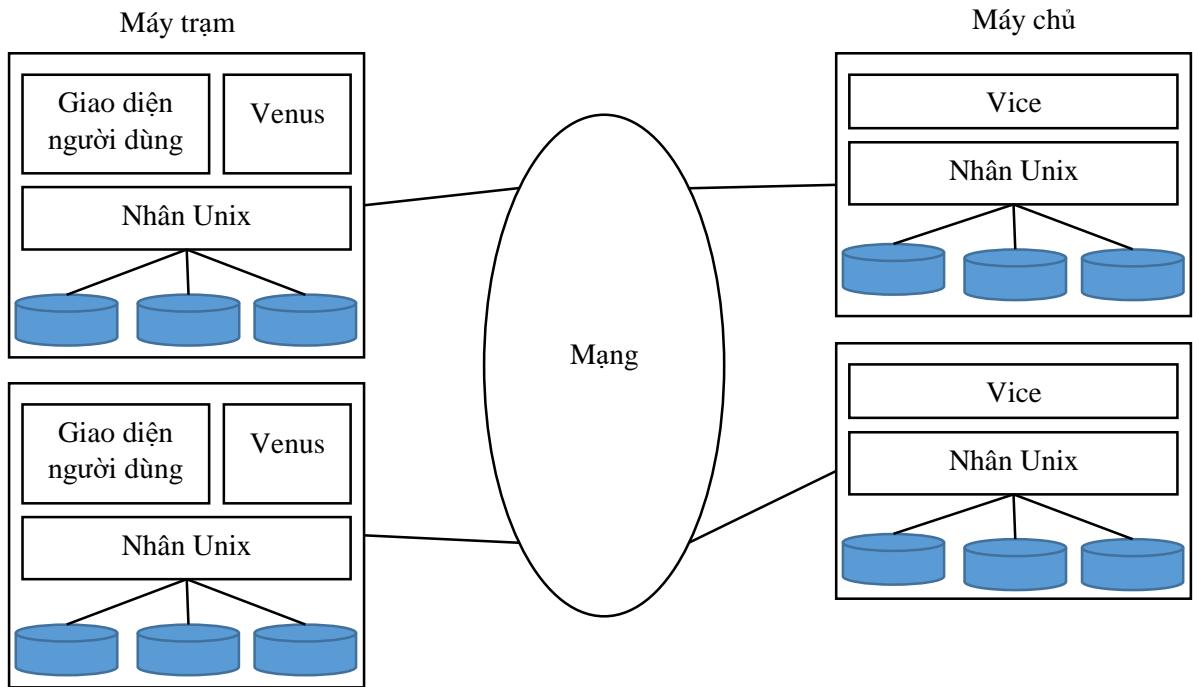
Một trong những đặc điểm nổi bật của AFS là cơ chế cache mạnh mẽ, giúp lưu trữ tệp tin vào bộ nhớ cache của máy khách khi lần đầu tiên truy cập, từ đó tăng tốc độ truy xuất và giảm tải cho máy chủ. Hệ thống cũng hỗ trợ khả năng quản lý quyền truy cập chi tiết thông qua cơ chế ACLs (Access Control Lists), cho phép kiểm soát quyền của người dùng hoặc nhóm người dùng đối với các tệp tin và thư mục trong hệ thống.

AFS được thiết kế với khả năng mở rộng cao, có thể xử lý một lượng lớn máy khách và máy chủ, và cung cấp tính linh hoạt trong việc phân phối và đồng bộ tệp tin giữa các máy chủ. Với tính năng bảo mật và sao lưu tốt, AFS trở thành một giải pháp đáng tin cậy cho việc chia sẻ và quản lý tệp tin trong các tổ chức, viện nghiên cứu, hoặc các công ty có quy mô lớn [65].

#### 2.4.3.2. Kiến trúc AFS

Kiến trúc của AFS được thiết kế để cung cấp một hệ thống tệp phân tán mạnh mẽ, hỗ trợ việc chia sẻ tệp tin trong các mạng lớn và phân tán. Kiến trúc này bao gồm nhiều

thành phần quan trọng, mỗi thành phần có vai trò riêng trong việc quản lý và truy xuất tệp tin qua mạng. AFS sử dụng mô hình máy khách – chủ, trong đó các máy khách yêu cầu và truy cập tệp tin từ các máy chủ tệp. Tệp tin trong AFS không chỉ được lưu trữ trên một máy chủ duy nhất mà có thể được phân phối trên nhiều máy chủ khác nhau trong mạng, giúp tối ưu hóa hiệu suất và khả năng mở rộng [64] [65].



*Hình 2.6: Kiến trúc AFS*

Hệ thống được xây dựng theo mô hình máy khách – máy chủ với hai thành phần chính là Venus (máy khách) và Vice (máy chủ). Vice là thành phần máy chủ, chịu trách nhiệm lưu trữ và quản lý tệp tin trong các volume, mỗi volume là một đơn vị lưu trữ logic chứa các thư mục và tệp tin, có thể được phân phối và sao chép giữa nhiều máy chủ để tăng tính sẵn sàng và phục hồi dữ liệu. Máy chủ Vice xử lý các yêu cầu từ máy khách và đảm bảo cung cấp dữ liệu đến các máy khách có kết nối [65].

Ở phía máy khách, Venus là một phần mềm trên máy khách AFS có nhiệm vụ kết nối với Vice để yêu cầu và nhận tệp tin từ xa. Venus thực hiện việc lưu trữ tạm thời các tệp tin đã truy cập vào bộ nhớ cache cục bộ, giúp tăng tốc độ truy xuất cho các lần sử dụng sau mà không cần yêu cầu liên tục từ máy chủ, nhờ đó giảm tải cho hệ thống máy chủ. Khi một máy khách yêu cầu tệp tin, Venus kiểm tra bộ nhớ cache trước tiên, và nếu tệp tin chưa có, nó sẽ gửi yêu cầu đến Vice để tải về [65].

#### *2.4.3.3. Tính nhất quán*

Tính nhất quán trong AFS liên quan đến việc đảm bảo rằng tất cả các máy khách luôn nhìn thấy cùng một phiên bản của dữ liệu khi truy cập vào cùng một tệp tin, bất kể tệp tin đó đã được chỉnh sửa ở đâu trong hệ thống. AFS sử dụng cơ chế lưu trữ tạm thời tại máy khách để cải thiện hiệu suất, cho phép máy khách truy cập tệp tin từ bộ nhớ cục bộ thay vì gửi yêu cầu đến máy chủ mỗi lần truy cập. Khi một máy khách tải tệp tin từ máy chủ, tệp tin sẽ được lưu vào bộ nhớ cache cục bộ của máy khách đó và có thể được sử dụng lại mà không cần phải yêu cầu từ máy chủ cho đến khi tệp tin bị thay đổi [64].

Để duy trì tính nhất quán, AFS sử dụng một cơ chế thông báo các máy khách về thay đổi. Khi một máy khách chỉnh sửa tệp tin và lưu lại, hệ thống sẽ cập nhật thay đổi lên máy chủ, và máy chủ sẽ gửi một thông báo đến các máy khách khác, yêu cầu họ làm mới hoặc vô hiệu hóa bản sao trong bộ nhớ tạm thời. Phiên bản đầu tiên của AFS chỉ thực hiện đồng bộ hóa khi tệp tin được mở hoặc đóng, vì vậy các thay đổi chỉ được truyền đến máy chủ sau khi máy khách đóng tệp tin đó, điều này có thể tạo ra một độ trễ nhất định [64].

#### *2.4.3.4. Ưu điểm và nhược điểm*

Về ưu điểm, AFS hỗ trợ chia sẻ tệp tin trên mạng một cách hiệu quả nhờ vào cơ chế phân phối tệp tin dưới dạng các volume, giúp hệ thống dễ dàng mở rộng và quản lý. Cơ chế cache tại máy khách cho phép cải thiện hiệu suất truy cập, giúp giảm tải cho máy chủ và tăng tốc độ truy xuất tệp tin khi máy khách có thể làm việc với bản sao cục bộ của tệp tin. Ngoài ra, AFS hỗ trợ tính bảo mật mạnh mẽ thông qua hệ thống xác thực người dùng và danh sách kiểm soát truy cập, cho phép quản trị viên thiết lập quyền truy cập chi tiết cho từng người dùng hoặc nhóm người dùng. Một ưu điểm khác là khả năng sao chép và phân phối volume giữa các máy chủ, giúp tăng tính sẵn sàng và độ tin cậy, đảm bảo dữ liệu luôn có thể truy cập ngay cả khi một số máy chủ gặp sự cố [64] [65].

Tuy nhiên, AFS cũng có những nhược điểm. Đầu tiên, tính nhất quán của AFS chủ yếu là nhất quán cuối cùng, tức là dữ liệu sẽ được đồng bộ sau một thời gian nhất định, nhưng không phải lúc nào cũng phản ánh ngay lập tức các thay đổi trên mọi máy khách. Điều này có thể gây ra sự không nhất quán tạm thời khi nhiều người dùng cùng chỉnh sửa tệp tin trong thời gian ngắn. Cơ chế thông báo gọi lại để cập nhật các tệp tin cache cũng có thể phức tạp trong các môi trường mạng lớn với nhiều máy khách, dễ gây ra lỗi hoặc độ trễ trong việc cập nhật. Ngoài ra, AFS yêu cầu cấu hình phức tạp và đòi hỏi khả năng quản lý cao từ phía quản trị viên, đặc biệt trong các hệ thống lớn, và có thể gây khó khăn cho những tổ chức chưa quen với mô hình hệ thống tệp phân tán [65].

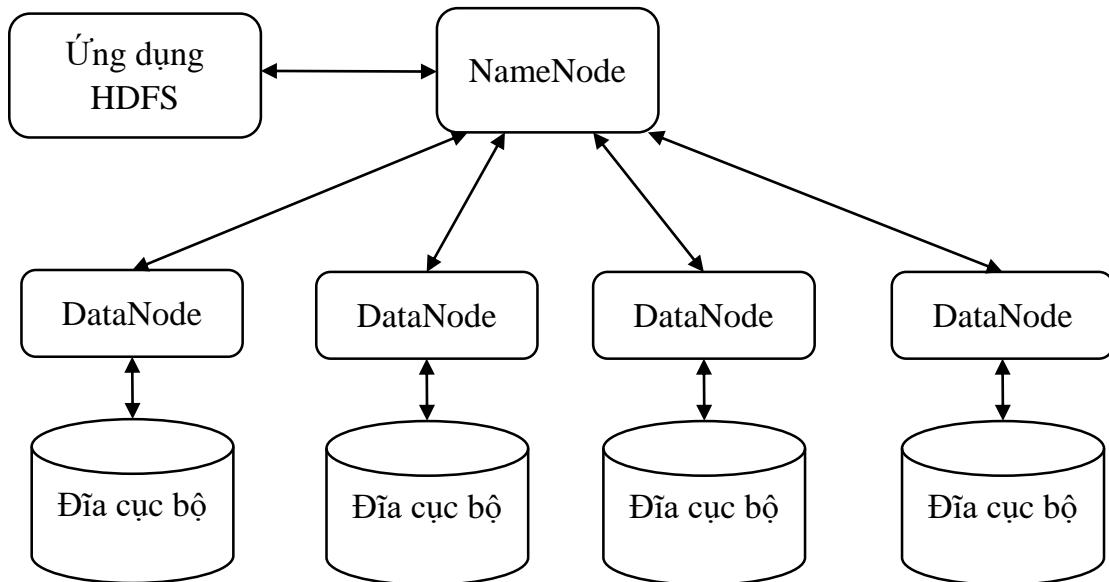
## 2.5. Hệ thống lưu trữ HDFS & GFS

Hệ thống lưu trữ HDFS (Hadoop Distributed File System) và GFS (Google File System) là hai kiến trúc lưu trữ phân tán phổ biến được thiết kế để quản lý và xử lý dữ liệu lớn trong môi trường phân tán. Cả hai hệ thống này đều được phát triển với mục tiêu cung cấp khả năng lưu trữ và xử lý dữ liệu với quy mô lớn, khả năng chịu lỗi và tính sẵn sàng cao.

### 2.5.1. Hệ thống tệp phân tán Hadoop

HDFS là hệ thống lưu trữ phân tán của Apache Hadoop, được thiết kế để quản lý các tập dữ liệu lớn trên các cụm máy tính với nhiều máy chủ [66]. HDFS phân chia một tập dữ liệu thành các khối dữ liệu nhỏ và phân phối chúng qua các nút khác nhau trong cụm. Các khối dữ liệu này được sao chép ở nhiều nút để đảm bảo tính sẵn sàng và khả năng chịu lỗi.

Khi một nút gặp sự cố, HDFS có thể sử dụng các bản sao trên các nút khác để duy trì tính toàn vẹn và tính liên tục của dữ liệu. HDFS bao gồm hai thành phần chính: NameNode và DataNode. NameNode giữ vai trò điều phối và quản lý thông tin về vị trí của các khối dữ liệu, trong khi DataNode thực hiện việc lưu trữ dữ liệu thực tế. HDFS phù hợp cho các ứng dụng phân tích dữ liệu lớn, nơi dữ liệu thường được ghi một lần và đọc nhiều lần [67].



Hình 2.7: Kiến trúc HDFS

Với các ưu điểm như khả năng mở rộng cao, HDFS có thể dễ dàng mở rộng bằng cách thêm các nút mới vào cụm mà không làm gián đoạn hoạt động, giúp nó thích hợp cho việc xử lý khối lượng dữ liệu lớn. Tính khả dụng và chịu lỗi tốt, HDFS thực hiện

sao chép dữ liệu trên nhiều DataNode (thường là 3 bản sao) để bảo vệ dữ liệu khỏi mất mát khi một hoặc nhiều máy chủ bị lỗi. Thiết kế tối ưu cho dữ liệu lớn, HDFS được thiết kế cho các ứng dụng cần xử lý dữ liệu quy mô lớn, trong đó các tệp dữ liệu có kích thước lớn hơn 100 MB là tiêu chuẩn.

Việc đọc dữ liệu lớn giúp giảm độ trễ khi truy xuất các khối dữ liệu từ nhiều DataNode. Khả năng tích hợp mạng mẽ, HDFS là một phần của hệ sinh thái Hadoop nên nó có khả năng tích hợp dễ dàng với các công cụ xử lý dữ liệu lớn như MapReduce, Hive, Spark và nhiều công cụ phân tích khác [66] [67].

Bên cạnh những ưu điểm nổi trội của HDFS thì cũng có những nhược điểm. Tốc độ xử lý dữ liệu nhỏ kém, HDFS không được tối ưu hóa cho các tệp nhỏ vì việc quản lý các tệp nhỏ gây ra nhiều tải cho NameNode, làm giảm hiệu suất của hệ thống. Phụ thuộc vào NameNode duy nhất, mặc dù NameNode được bảo vệ bằng cách sao lưu, HDFS vẫn có một điểm dễ bị lỗi do NameNode quản lý tất cả metadata về vị trí của dữ liệu. Nếu NameNode gặp sự cố, hệ thống sẽ bị gián đoạn cho đến khi nó được khôi phục.

Độ trễ cao cho các tác vụ cập nhật và ghi nhiều lần, HDFS không phù hợp cho các ứng dụng cần ghi dữ liệu thường xuyên và cập nhật theo thời gian thực vì dữ liệu trong HDFS chủ yếu được ghi một lần và đọc nhiều lần. Thiếu tính năng bảo mật phức tạp, HDFS không có sẵn các tính năng bảo mật phức tạp như mã hóa tích hợp. Một số tính năng này có thể được triển khai qua Kerberos hoặc qua các cấu hình bổ sung, nhưng điều này phức tạp hơn so với các hệ thống khác [66] [67].

### Cài đặt HDFS

HDFS sẽ được cài đặt trên môi trường WSL2 (Windows Subsystem for Linux 2). Bước đầu tiên để cài đặt HDFS phải đảm bảo đã cài đặt môi trường Java (ở đây sử dụng Java 21). Nếu máy chưa có Java có thể cài đặt theo các bước sau, mở Ubuntu và nhập câu lệnh:

Code:

```
sudo apt install openjdk-21-jdk -y
```

Sau khi nhập lệnh và chạy sẽ hiện thị kết quả như sau:

Kết quả:

```
The following additional packages will be installed: ...
```

```
0 upgraded, 23 newly installed, 0 to remove, and 0 not upgraded.
```

Để kiểm tra Java đã được cài đặt chưa, nhập câu lệnh:

Code:

```
java --version
```

Câu lệnh này sẽ hiện thị kết quả như sau:

Kết quả:

```
openjdk 21.0.5 2024-10-15
```

```
OpenJDK Runtime Environment (build 21.0.5+11-Ubuntu-1ubuntu124.04)
```

```
OpenJDK 64-Bit Server VM (build 21.0.5+11-Ubuntu-1ubuntu124.04, mixed mode, sharing)
```

Sau khi đã cài đặt Java thành công, cần gán đường dẫn Java vào JAVA\_HOME. Sau đó kiểm tra đường dẫn đã đúng hay chưa với câu lệnh:

Code:

```
dirname $(dirname $(readlink -f $(which java)))
```

Kết quả của câu lệnh trên sẽ là:

Kết quả:

```
/usr/lib/jvm/java-21-openjdk-amd64
```

Sau khi đã thiết lập môi trường Java thành công, bước thứ hai là tạo user trong Ubuntu để chạy tất cả thành phần Hadoop như đăng nhập vào giao diện Web của Hadoop. Để tạo user mới, hãy thực hiện theo các bước sau đây:

Code:

```
sudo adduser hadoop
```

Khi chạy câu lệnh này, hãy nhập mật khẩu cho tài khoản hadoop, với các trường dữ liệu có thể nhấn Enter để bỏ qua. Sau khi câu lệnh chạy thành công sẽ có kết quả:

Kết quả:

```
info: Adding new user `hadoop' to supplemental / extra groups `users' ...
```

```
info: Adding user `hadoop' to group `users' ...
```

Sau khi tạo thành công user hadoop, cần chuyển qua tài khoản hadoop để tiếp tục cài đặt HDFS với câu lệnh:

Code:

```
su -- hadoop
```

Bước kế tiếp cần tạo cặp khóa SSH để cấu hình quyền truy cập SSH với câu lệnh:

Code:

```
ssh-keygen -t rsa
```

Sau đó nhấn Enter để chọn mặc định đường dẫn:

Kết quả:

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
```

Ké tiếp copy khóa công khai vừa mới tạo vào file khóa ủy quyền và thiết lập quyền phù hợp với câu lệnh:

Code:

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Sau khi chạy câu lệnh trên tiếp tục với câu lệnh:

Code:

```
chmod 640 ~/.ssh/authorized_keys
```

Sau khi setup SSH thành công, bước tiếp theo sẽ cài đặt Hadoop

Code:

```
wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz
```

Kết quả thành công của câu lệnh trên:

Kết quả:

```
'hadoop-3.3.6.tar.gz' saved [730107476/730107476]
```

Di chuyển tệp hadoop vừa mới tải tới thư mục \$HOME và giải nén:

Code:

```
mv hadoop-3.3.6.tar.gz $HOME/
```

```
cd $HOME/
```

```
tar -xvzf hadoop-3.3.6.tar.gz
```

Mở file .profile và cấu hình thư viện cần thiết với câu lệnh:

Code:

```
nano .profile
```

Copy các lệnh sau đây vào thư mục .profile

Code:

```
export JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64  
export HADOOP_HOME=$HOME/hadoop-3.3.6  
export HADOOP_INSTALL=$HADOOP_HOME  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export HADOOP_YARN_HOME=$HADOOP_HOME  
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native  
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin  
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

Nhấn Ctrl+X và Y để lưu lại và load lại file .profile với câu lệnh:

Code:

```
source ~/.profile
```

Bước kế tiếp cần cấu hình biến JAVA\_HOME trước đó vào file hadoop-env.sh

Code:

```
nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

Tìm dòng # export JAVA\_HOME= và thêm lệnh sau vào đó:

Code:

```
export JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64
```

Nhấn Ctrl+X và Y để lưu lại. Như vậy là hoàn thành bước cấu hình môi trường Hadoop. Bước tiếp theo cần cấu hình DataNode và NameNode

Code:

```
mkdir -p ~/hadoopdata/hdfs/namenode  
mkdir -p ~/hadoopdata/hdfs/datanode
```

Chỉnh sửa tệp core-site.xml để cập nhập địa chỉ localhost:

Code:

```
nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

Nhập code sau vào:

Code:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

Ctrl+X và Y để lưu tệp core-site.xml. Tiếp theo chỉnh sửa tệp hdfs-site.xml:

Code:

```
nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

Cập nhật tệp hdfs-site.xml với đoạn code sau:

Code:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
  </property>
</configuration>
```

Lưu lại và tiếp tục cập nhật tệp mapred-site.xml:

Code:

```
nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

Cập nhật với đoạn code sau:

Code:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

Lưu lại và tiếp tục cập nhật tệp yarn-site.xml:

Code:

```
nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

Cập nhật với đoạn code sau:

Code:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

Lưu lại và bước cuối cùng là chạy Hadoop với câu lệnh:

Code:

```
hdfs namenode -format
```

Sau khi chạy câu lệnh thành công sẽ có kết quả:

Kết quả:

```
Storage directory /home/hadoop/hadoopdata/hdfs/namenode has been successfully
formatted.
```

Cuối cùng chạy lệnh sau đây để khởi động Hadoop:

Code:

```
start-all.sh
```

Kết quả của câu lệnh trên:

Kết quả:

Starting namenodes on [localhost]

Starting datanodes

Starting secondary namenodes [Admin]

Starting resourcemanager

Starting nodemanagers

Chạy URL <http://localhost:9870> để khởi động giao diện Web của Hadoop.

The screenshot shows a browser window with the URL [localhost:9870/dfshealth.html#tab-overview](http://localhost:9870/dfshealth.html#tab-overview). The page has a green header bar with tabs: Hadoop (selected), Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, Utilities. Below the header is a table with cluster details:

Started:	Wed Nov 13 20:30:16 +0700 2024
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 15:22:00 +0700 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-876ea07d-a3d7-4af0-861a-63bf969389a7
Block Pool ID:	BP-489875978-127.0.1.1-1731504587976

Below the table is a 'Summary' section with the note 'Security is off.'

Hình 2.8: Cài đặt thành công Hadoop

Theo hình 2.8, giao diện của Hadoop phần Overview mô tả thông tin tổng quan về NameNode. Trường Started mô tả thời điểm NameNode được khởi động, trường version mô tả phiên bản Hadoop đang chạy ở đây là bản 3.3.6. Trường Compiled là thông tin thời điểm Hadoop được biên dịch và cluster ID là mã định danh của cụm Hadoop. Phần Summary hiện thị trạng thái tổng quan và các thông tin bảo mật,

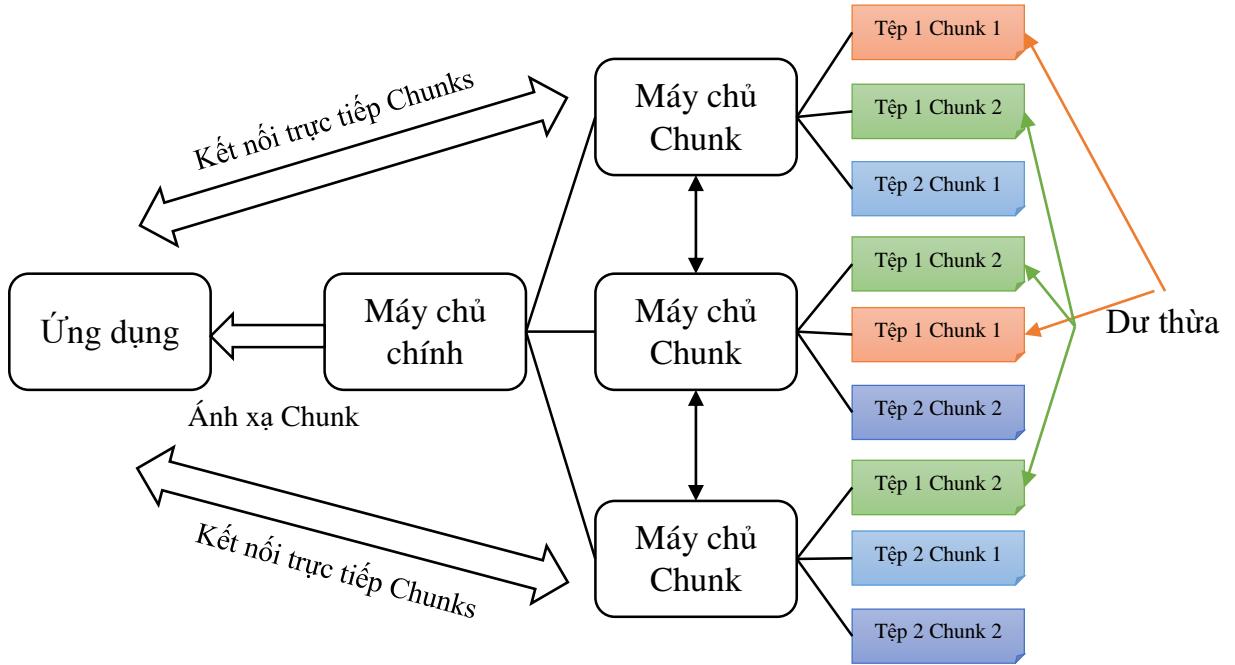
Trên thanh Menu cung cấp các liên kết như Datanodes là danh sách các DataNode trong cụm, kèm theo trạng thái dung lượng và block quản lý. Datanode Volume Failures, hiển thị các lỗi liên quan đến ổ đĩa của các DataNode. Snapshot, quản lý ảnh chụp nhanh của dữ liệu trong HDFS. Startup Progress, hiển thị tiến trình khởi động của NameNode. Utilities, công cụ hỗ trợ bao gồm xem logs và cấu hình.

### 2.5.2. Hệ thống tệp Google

GFS là hệ thống lưu trữ phân tán được phát triển bởi Google, dành cho các ứng dụng yêu cầu lưu trữ và xử lý dữ liệu lớn với khả năng mở rộng và khả năng chịu lỗi cao [68]. Tương tự HDFS, GFS phân chia các tập dữ liệu thành các khối nhỏ hơn, gọi là "chunks," và lưu trữ chúng trên các máy chủ khác nhau trong một cụm [69]. GFS sử

dụng Master Server để quản lý thông tin về các chunks, và Chunk Servers để lưu trữ các chunks dữ liệu.

Hệ thống này cũng đảm bảo rằng các chunks được sao chép trên nhiều máy chủ để duy trì tính sẵn sàng khi có sự cố [69]. Một điểm khác biệt của GFS là hệ thống này được tối ưu hóa để xử lý các ứng dụng cần lưu trữ và phân tích dữ liệu với khối lượng lớn trong thời gian thực, chẳng hạn như công cụ tìm kiếm của Google.



Hình 2.9: Kiến trúc GFS

Ưu điểm của GFS là tối ưu hóa cho dữ liệu lớn và tần suất ghi cao, GFS được thiết kế để lưu trữ và xử lý khối lượng dữ liệu rất lớn, đặc biệt là trong môi trường ứng dụng đòi hỏi ghi dữ liệu thường xuyên. Đây là một ưu điểm quan trọng trong các hệ thống xử lý dữ liệu của Google, chẳng hạn như công cụ tìm kiếm và quảng cáo. Tính năng kiểm soát lỗi tự động, GFS liên tục theo dõi các lỗi phần cứng và tự động di chuyển các chunks dữ liệu khi phát hiện lỗi, giúp đảm bảo rằng dữ liệu luôn sẵn sàng.

Phân tán metadata để giảm tải cho Master Server, Master Server của GFS chỉ lưu trữ metadata về vị trí chunks, trong khi chính dữ liệu lại được lưu trên các Chunk Server. Điều này giúp giảm tải cho Master Server, cho phép nó quản lý hàng ngàn chunks một cách hiệu quả. Khả năng tùy chỉnh linh hoạt, GFS cung cấp khả năng tùy chỉnh dễ dàng cho các tác vụ ghi đồng thời và kiểm soát việc sao lưu dữ liệu, cho phép người dùng tối ưu hóa hệ thống cho từng loại ứng dụng cụ thể [68] [69].

Bên cạnh những ưu điểm GFS cũng có những nhược điểm như phức tạp trong việc xử lý dữ liệu ngoài Google, GFS được tối ưu hóa và điều chỉnh đặc biệt cho các hệ thống

của Google, làm cho nó khó được áp dụng rộng rãi bên ngoài môi trường này. Vì vậy, nếu một tổ chức muốn sử dụng hệ thống tương tự, cần có sự phát triển hoặc điều chỉnh lớn. Tính nhất quán phức tạp, khi ghi đồng thời vào các chunk trong GFS, dữ liệu có thể gặp phải vấn đề không nhất quán tạm thời.

Mặc dù Google có các giao thức riêng để xử lý vấn đề này, nó vẫn có thể gây ra các lỗi không mong muốn khi sử dụng hệ thống. Giống như NameNode trong HDFS, Master Server trong GFS là một điểm yếu duy nhất trong hệ thống. Nếu Master Server bị lỗi, hệ thống sẽ ngừng hoạt động cho đến khi được khôi phục. Không phù hợp với dữ liệu nhỏ, GFS hoạt động kém hiệu quả với các tệp nhỏ, tương tự như HDFS, vì việc quản lý metadata của các tệp nhỏ tiêu tốn tài nguyên và làm giảm hiệu suất [68] [69].

## 2.6. Cơ sở dữ liệu NoSQL

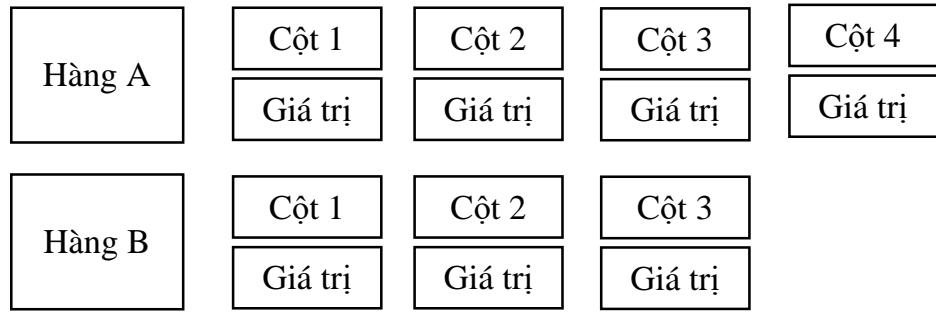
### 2.6.1. Khái niệm

Cơ sở dữ liệu NoSQL là một loại cơ sở dữ liệu không sử dụng mô hình quan hệ truyền thống, mang đến khả năng lưu trữ và quản lý dữ liệu linh hoạt, không cần cấu trúc cứng nhắc như trong cơ sở dữ liệu quan hệ. Đặc điểm của NoSQL là hỗ trợ khả năng mở rộng ngang, cho phép tăng khả năng lưu trữ và xử lý bằng cách thêm nhiều máy chủ khi dữ liệu phát triển. Nhờ vào tính linh hoạt này, NoSQL rất phù hợp với các ứng dụng hiện đại có yêu cầu về hiệu suất cao và khối lượng dữ liệu lớn, như mạng xã hội, ứng dụng IoT, và dịch vụ thương mại điện tử [70] [71].

Cơ sở dữ liệu NoSQL cũng hỗ trợ lưu trữ dữ liệu phi cấu trúc hoặc bán cấu trúc một cách hiệu quả, giúp cho việc quản lý dữ liệu không nhất thiết phải theo một lược đồ cố định. Điều này giúp NoSQL thích hợp với những ứng dụng có dữ liệu biến đổi liên tục, nơi mà việc thay đổi cấu trúc dữ liệu không thể thực hiện dễ dàng với các hệ thống quan hệ truyền thống. Tuy nhiên, vì đặc điểm thiết kế tập trung vào hiệu suất và khả năng mở rộng, NoSQL có thể không đạt được tính nhất quán dữ liệu cao như SQL và có thể không phù hợp với các ứng dụng đòi hỏi tính chính xác cao về giao dịch [71].

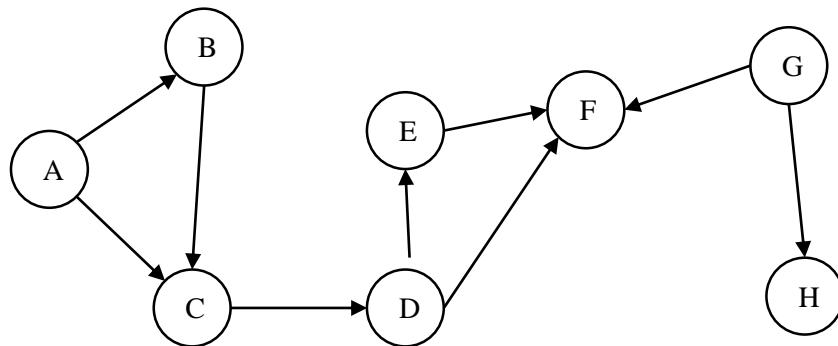
### 2.6.2. Các dạng lưu trữ

Không giống như các hệ thống cơ sở dữ liệu quan hệ truyền thống, NoSQL không yêu cầu cấu trúc dữ liệu cố định và không sử dụng các bảng hoặc quan hệ để tổ chức dữ liệu. Thay vào đó, NoSQL bao gồm nhiều loại hình lưu trữ khác nhau, như cơ sở dữ liệu tài liệu, cơ sở dữ liệu dạng cặp khóa-giá trị, cơ sở dữ liệu đồ thị, và cơ sở dữ liệu dạng cột, mỗi loại phù hợp với các loại dữ liệu và nhu cầu xử lý khác nhau.



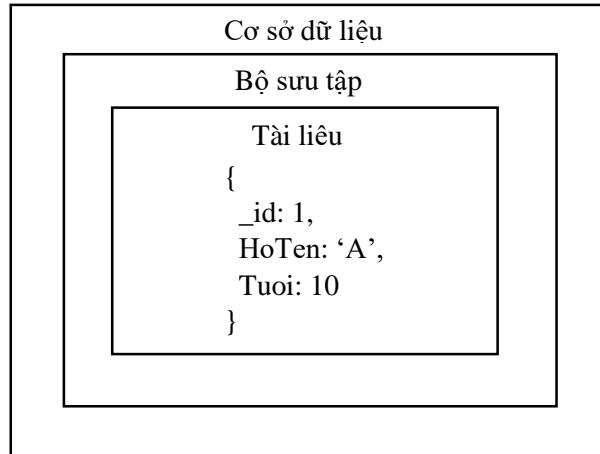
*Hình 2.10: Cơ sở dữ liệu dạng cột*

Cơ sở dữ liệu dạng cột lưu trữ dữ liệu dưới dạng các nhóm cột, mỗi nhóm có thể lưu trên nhiều máy chủ khác nhau, giúp cho việc mở rộng dữ liệu lớn một cách dễ dàng. Dữ liệu được lưu trong từng nhóm cột riêng biệt, cho phép truy vấn nhanh theo cột, rất phù hợp cho các ứng dụng xử lý khối lượng lớn dữ liệu, như Apache Cassandra và HBase. Dạng lưu trữ này thường được sử dụng khi dữ liệu có cấu trúc lặp lại, cần truy xuất nhanh theo cột và mở rộng dễ dàng, nhưng không phù hợp cho dữ liệu phức tạp [72].



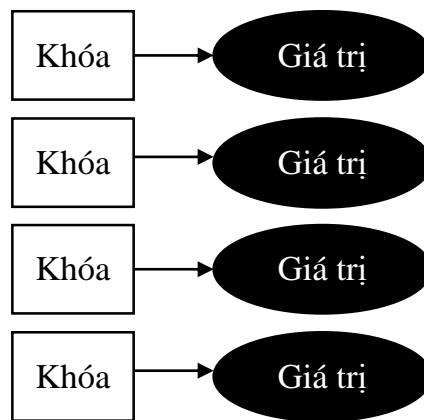
*Hình 2.11: Cơ sở dữ liệu dạng đồ thị*

Cơ sở dữ liệu dạng đồ thị lưu trữ dữ liệu theo các nút và cạnh để mô tả các mối quan hệ phức tạp giữa các dữ liệu, rất hữu ích cho các ứng dụng có cấu trúc liên kết và kết nối phức tạp, chẳng hạn như mạng xã hội, hệ thống khuyến nghị, và các hệ thống phân tích mối quan hệ. Các cơ sở dữ liệu đồ thị như Neo4j, Amazon Neptune và OrientDB tối ưu cho các truy vấn quan hệ phức tạp và biểu diễn trực quan các mối liên hệ, nhưng khả năng mở rộng ngang có thể bị giới hạn và không phù hợp với các ứng dụng có dữ liệu phi cấu trúc không cần mối quan hệ [71] [72].



*Hình 2.12: Cơ sở dữ liệu dạng tài liệu*

Cơ sở dữ liệu dạng tài liệu lưu trữ dữ liệu dưới dạng tài liệu riêng biệt, thường là các định dạng JSON hoặc BSON. Mỗi tài liệu là một đơn vị lưu trữ độc lập với cấu trúc linh hoạt, cho phép chứa các trường khác nhau và dễ dàng thay đổi mà không cần chỉnh sửa toàn bộ hệ thống, rất phù hợp để lưu trữ dữ liệu phi cấu trúc hoặc bán cấu trúc. Điều này giúp các cơ sở dữ liệu tài liệu như MongoDB và Couchbase linh hoạt, có thể mở rộng ngang dễ dàng, phù hợp cho những ứng dụng với dữ liệu không có cấu trúc cố định [72].



*Hình 2.13: Cơ sở dữ liệu dạng cặp khóa - giá trị*

Cơ sở dữ liệu dạng cặp khóa-giá trị lưu trữ dữ liệu dưới dạng các cặp khóa và giá trị, với khóa là định danh duy nhất để truy xuất dữ liệu một cách nhanh chóng. Giá trị có thể là bất kỳ kiểu dữ liệu nào, từ chuỗi đơn giản đến các đối tượng phức tạp. Các cơ sở dữ liệu khóa-giá trị như Redis, DynamoDB và Riak rất lý tưởng cho các ứng dụng đòi hỏi tốc độ truy xuất cao và lưu trữ nhanh như bộ nhớ đệm, nhưng không hiệu quả khi truy vấn các yêu cầu phức tạp do thiếu khả năng lập chỉ mục theo giá trị [71] [72].

### **2.6.3. Ưu điểm và nhược điểm**

Ưu điểm của cơ sở dữ liệu NoSQL bao gồm khả năng mở rộng ngang, NoSQL cho phép mở rộng dễ dàng bằng cách thêm nhiều máy chủ thay vì tăng dung lượng của một máy chủ duy nhất, điều này phù hợp cho các ứng dụng hiện đại, nơi nhu cầu về lưu trữ và xử lý dữ liệu có thể tăng nhanh chóng. Linh hoạt về cấu trúc dữ liệu, NoSQL có khả năng xử lý dữ liệu phi cấu trúc và bán cấu trúc mà không yêu cầu lược đồ cố định, cho phép thay đổi cấu trúc dữ liệu mà không cần phải chỉnh sửa cơ sở dữ liệu toàn diện.

Hiệu suất truy xuất cao, nhiều loại cơ sở dữ liệu NoSQL, đặc biệt là các dạng cặp khóa-giá trị và tài liệu, được thiết kế để tối ưu hóa tốc độ truy xuất và xử lý dữ liệu, làm cho chúng phù hợp cho các ứng dụng đòi hỏi hiệu suất cao như bộ nhớ đệm và xử lý dữ liệu thời gian thực. Phù hợp với dữ liệu lớn, nhờ khả năng mở rộng ngang và khả năng lưu trữ dữ liệu phi cấu trúc, NoSQL là lựa chọn tốt cho các ứng dụng xử lý dữ liệu lớn, chẳng hạn như phân tích dữ liệu lớn, IoT, và các mạng xã hội [70].

Bên cạnh các ưu điểm, cơ sở dữ liệu NoSQL cũng có những nhược điểm. Thiếu tính nhất quán mạnh, NoSQL thường bỏ tính nhất quán để đảm bảo khả năng mở rộng và hiệu suất, nhiều hệ thống NoSQL sử dụng mô hình "cuối cùng nhất quán" thay vì tính nhất quán tức thì như các cơ sở dữ liệu quan hệ, do đó có thể không phù hợp với các ứng dụng yêu cầu tính chính xác cao về giao dịch. Giới hạn về hỗ trợ truy vấn phức tạp, NoSQL thiếu tính linh hoạt trong việc thực hiện các truy vấn phức tạp.

Thiếu tiêu chuẩn hóa, không giống SQL với chuẩn ngôn ngữ truy vấn thống nhất, NoSQL có nhiều dạng lưu trữ và mỗi loại sử dụng một cách tiếp cận và giao diện truy vấn khác nhau, gây khó khăn cho việc chuyển đổi giữa các hệ thống. Tính bảo mật và công cụ quản lý hạn chế hơn, vì các hệ thống NoSQL còn khá mới so với SQL, một số hệ thống NoSQL có thể không hỗ trợ tốt các tiêu chuẩn bảo mật cao cấp và các công cụ quản lý phức tạp, điều này có thể là thách thức cho các doanh nghiệp đòi hỏi bảo mật cao [71].

## **2.7. Dữ liệu lớn và điện toán đám mây**

Dữ liệu lớn (Big Data) và điện toán đám mây (Cloud Computing) là hai công nghệ quan trọng trong thời đại số hiện nay, chúng có sự kết hợp mạnh mẽ giúp xử lý và lưu trữ lượng dữ liệu khổng lồ mà các doanh nghiệp và tổ chức phải đối mặt hàng ngày.

### **2.7.1. Dữ liệu lớn**

Big Data là thuật ngữ chỉ các tập dữ liệu có kích thước và độ phức tạp vượt quá khả năng xử lý của các công cụ quản lý dữ liệu truyền thống [73]. Dữ liệu này có thể đến từ nhiều nguồn khác nhau, như mạng xã hội, thiết bị di động, cảm biến IoT, giao

dịch trực tuyến và các hệ thống máy tính. Đặc điểm nổi bật của Big Data là 3V: Volume (Khối lượng): Dữ liệu lớn với kích thước khổng lồ, từ hàng terabyte đến petabyte. Velocity (Tốc độ): Dữ liệu được tạo ra và cập nhật nhanh chóng, đòi hỏi xử lý kịp thời. Variety (Độ đa dạng): Dữ liệu đến từ nhiều nguồn với các định dạng khác nhau, bao gồm dữ liệu cấu trúc, bán cấu trúc và phi cấu trúc (như văn bản, hình ảnh, video) [74].

BigData với các ưu điểm như: Phân tích dự đoán và đưa ra quyết định chính xác, Big Data cho phép các doanh nghiệp phân tích dữ liệu từ nhiều nguồn khác nhau để đưa ra những quyết định chính xác hơn. Các công cụ phân tích có thể dự đoán xu hướng tương lai, hành vi của khách hàng, và các cơ hội kinh doanh mới. Tối ưu hóa hoạt động kinh doanh, thông qua phân tích Big Data, doanh nghiệp có thể tối ưu hóa quy trình, giảm thiểu lãng phí và cải thiện hiệu quả hoạt động.

Khả năng phát hiện các mối nguy hiểm và sự cố sớm, Big Data có thể giúp phát hiện các vấn đề tiềm ẩn trong các hệ thống (như máy móc, mạng lưới hay dữ liệu) trước khi chúng trở thành sự cố lớn, từ đó giảm thiểu thời gian chết và chi phí sửa chữa. Khả năng xử lý dữ liệu lớn và đa dạng, Big Data có khả năng xử lý không chỉ dữ liệu cấu trúc (structured data) mà còn dữ liệu phi cấu trúc (unstructured data) như văn bản, hình ảnh, video, và âm thanh, giúp mở rộng khả năng phân tích cho các doanh nghiệp [74].

Bên cạnh ưu điểm nổi bật thì BigData cũng có những nhược điểm. Chi phí đầu tư và duy trì cao, việc thu thập, lưu trữ và xử lý Big Data đòi hỏi đầu tư lớn vào hạ tầng phần cứng và phần mềm, bao gồm các công nghệ tiên tiến như hệ thống phân tán, cơ sở dữ liệu NoSQL, và các công cụ phân tích Big Data (như Hadoop, Spark). Ngoài ra, việc duy trì và vận hành hệ thống cũng tốn kém và phức tạp. Yêu cầu kỹ năng chuyên môn cao, việc phân tích và xử lý Big Data đòi hỏi các chuyên gia có kỹ năng cao về dữ liệu, phân tích và công nghệ.

Việc thiếu nhân lực có chuyên môn sẽ khiến doanh nghiệp gặp khó khăn trong việc khai thác giá trị từ dữ liệu lớn. Tính chính xác và độ tin cậy của dữ liệu, dữ liệu thu thập từ nhiều nguồn khác nhau có thể không đồng nhất hoặc không chính xác. Nếu không xử lý đúng cách, dữ liệu có thể dẫn đến các phân tích sai lệch hoặc đưa ra các quyết định không chính xác [74].

### **2.7.2. Mối quan hệ**

Big Data và điện toán đám mây có một mối quan hệ tương hỗ rất mạnh mẽ. Điện toán đám mây cung cấp các nền tảng và tài nguyên cần thiết để xử lý và lưu trữ các tập dữ liệu khổng lồ của Big Data. Những tính năng linh hoạt và mở rộng của điện toán đám

mây giúp các tổ chức có thể tăng hoặc giảm tài nguyên khi cần thiết để xử lý lượng dữ liệu thay đổi liên tục mà không cần phải đầu tư vào phần cứng đắt đỏ [75].

Cụ thể, khi xử lý Big Data, các tổ chức có thể sử dụng IaaS từ các nhà cung cấp đám mây để có máy chủ ảo và lưu trữ dữ liệu, hoặc PaaS để phát triển các ứng dụng phân tích dữ liệu lớn. Các dịch vụ SaaS có thể cung cấp các công cụ phân tích và báo cáo giúp người dùng hiểu và trích xuất giá trị từ dữ liệu.

### **2.7.3. Công cụ**

#### *2.7.3.1. Amazon Web Service (AWS)*

Amazon EMR (Elastic MapReduce) là một dịch vụ của AWS giúp xử lý lượng dữ liệu lớn bằng các công cụ phân tích như Hadoop, Spark. EMR cho phép xử lý dữ liệu trong môi trường đám mây, tiết kiệm chi phí và thời gian, từ đó giúp các tổ chức phân tích dữ liệu nhanh chóng và hiệu quả.

Amazon Redshift là dịch vụ cơ sở dữ liệu kho dữ liệu (data warehouse) hoàn toàn được quản lý trên đám mây. Nó giúp lưu trữ và phân tích khối lượng dữ liệu khổng lồ với hiệu suất cao, cho phép các công ty thực hiện các phân tích dữ liệu lớn và các báo cáo phân tích mạnh mẽ. AWS Glue là dịch vụ tích hợp và làm sạch dữ liệu, giúp doanh nghiệp dễ dàng chuẩn hóa và tải dữ liệu vào kho dữ liệu. Đây là công cụ giúp xử lý Big Data một cách linh hoạt và dễ dàng tích hợp với các dịch vụ AWS khác.

#### *2.7.3.2. Google Cloud Platform (GCP)*

Google BigQuery là một dịch vụ phân tích dữ liệu lớn được Google cung cấp, cho phép người dùng lưu trữ và phân tích các tập dữ liệu lớn với tốc độ nhanh và không cần phải quản lý hạ tầng. BigQuery hỗ trợ các phân tích thời gian thực với các công cụ AI tích hợp.

Google Cloud Dataproc là dịch vụ của GCP giúp chạy các cụm Hadoop và Spark trên đám mây, giúp xử lý các khối lượng dữ liệu lớn với chi phí hợp lý. Dịch vụ này tối ưu hóa quá trình xử lý và phân tích dữ liệu bằng cách tận dụng các công cụ mã nguồn mở. Google Cloud Dataflow giúp xử lý và phân tích dữ liệu theo dòng hoặc theo lô. Dataflow hỗ trợ các ứng dụng phân tích dữ liệu và máy học để thực hiện các phân tích dữ liệu lớn theo thời gian thực.

#### *2.7.3.3. Microsoft Azure*

Azure HDInsight là một dịch vụ của Microsoft Azure cung cấp nền tảng quản lý cho các công cụ Big Data như Hadoop, Spark, Hive, và HBase, giúp doanh nghiệp xử

lý các tập dữ liệu lớn trên đám mây. HDInsight hỗ trợ nhiều công cụ phân tích dữ liệu, từ phân tích dữ liệu đến mô hình hóa dữ liệu.

Azure Synapse Analytics (trước đây là Azure SQL Data Warehouse) là dịch vụ kho dữ liệu tích hợp giúp tổ chức phân tích và tổng hợp dữ liệu lớn từ các nguồn khác nhau. Nó hỗ trợ phân tích dữ liệu nhanh chóng với hiệu suất cao và tích hợp các công cụ AI.

#### 2.7.3.4. IBM Cloud

IBM Cloud Pak for Data là một nền tảng đám mây giúp doanh nghiệp thu thập, quản lý và phân tích dữ liệu lớn. Nó tích hợp nhiều công cụ Big Data như IBM Watson Studio, IBM SPSS Modeler và các công cụ phân tích khác, giúp tối ưu hóa các quy trình phân tích và đưa ra các quyết định dựa trên dữ liệu. IBM Watson Studio cung cấp công cụ và nền tảng để phân tích dữ liệu lớn và xây dựng các mô hình học máy, hỗ trợ Big Data trong các phân tích sâu và dự đoán xu hướng.

### 2.8. Khái quát cơ sở dữ liệu và tác động tới điện toán đám mây

Điện toán đám mây mang lại nhiều lợi ích cho người tiêu dùng dịch vụ nhờ khả năng chia sẻ tài nguyên và cơ sở hạ tầng hiệu quả, tuy nhiên cũng tồn tại những lo ngại về bảo mật do tính chất chia sẻ lớn của nền tảng này, dẫn đến các rủi ro tiềm ẩn. Một hệ thống bảo mật dựa trên đám mây cần phải đáp ứng đầy đủ các yêu cầu cơ bản của một hệ thống thông tin, bao gồm bảo đảm tính bảo mật, tính toàn vẹn, tính sẵn có của thông tin, cùng với việc quản lý danh tính, xác thực và ủy quyền người dùng [76].

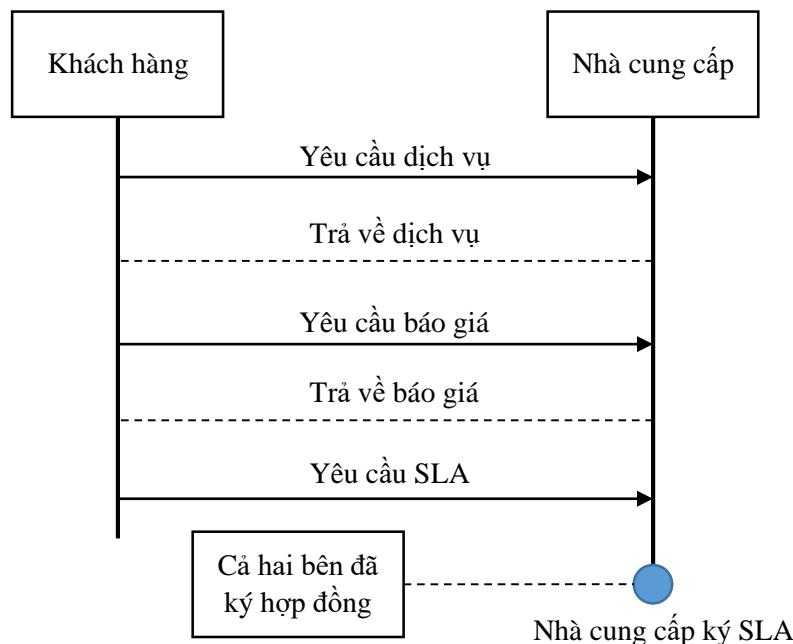
#### 2.8.1. Nguyên lý bảo mật chung

Việc triển khai cơ chế quản lý danh tính và kiểm soát truy cập trong doanh nghiệp là yếu tố thiết yếu để bảo vệ cả hệ thống máy tính và dữ liệu mà nó lưu trữ [77]. Bảo mật được xem như một khía cạnh quan trọng của kiến trúc điện toán đám mây, bao trùm toàn bộ các lớp trong mô hình tham chiếu [76]. Nguyên tắc bảo mật trong đám mây bao gồm việc đảm bảo an ninh từ tầng vật lý của cơ sở hạ tầng đến các ứng dụng, nhằm duy trì tính toàn vẹn, sẵn có và bảo mật của thông tin [77].

Nhà cung cấp dịch vụ đám mây phải đảm bảo an ninh cho cơ sở hạ tầng của chính mình, cũng như bảo vệ dữ liệu và ứng dụng của khách hàng. Người tiêu dùng dịch vụ cần phải xác minh và đảm bảo rằng nhà cung cấp đã áp dụng đầy đủ các biện pháp bảo mật nhằm bảo vệ an toàn cho các dịch vụ mà họ sử dụng. Đồng thời, nhà cung cấp cần thiết lập một mối quan hệ tin cậy với người tiêu dùng, đảm bảo rằng cả hai bên đều hiểu rõ trách nhiệm của mình trong việc quản lý và bảo vệ bảo mật [78].

### 2.8.2. Thỏa thuận ở mức dịch vụ SLA

SLA (Service Level Agreement) là thỏa thuận ở mức dịch vụ, một hợp đồng hoặc tài liệu giữa nhà cung cấp dịch vụ và khách hàng, xác định các dịch vụ cụ thể mà nhà cung cấp cam kết cung cấp, mức độ chất lượng dịch vụ, và các tiêu chuẩn mà dịch vụ cần đáp ứng. Mục tiêu của SLA là thiết lập các kỳ vọng rõ ràng và minh bạch giữa các bên về các tiêu chí dịch vụ như độ sẵn sàng, hiệu suất, bảo mật và các cam kết khác [79].



Hình 2.14: Quy trình thỏa thuận mức dịch vụ

Theo hình 2.13, sơ đồ minh họa quy trình giao dịch giữa khách hàng và nhà cung cấp trong việc cung cấp dịch vụ. Quá trình bắt đầu bằng việc khách hàng gửi yêu cầu dịch vụ đến nhà cung cấp. Nhà cung cấp sau đó sẽ hoàn thiện mẫu yêu cầu này và gửi lại cho khách hàng để xác nhận. Sau đó, khách hàng sẽ yêu cầu một báo giá chi tiết cho dịch vụ và nhà cung cấp sẽ cung cấp báo giá.

Sau khi thống nhất về giá cả, khách hàng sẽ yêu cầu một thỏa thuận về mức độ dịch vụ (SLA) để đảm bảo chất lượng dịch vụ. Khi cả hai bên đã đồng ý với các điều khoản trong SLA, nhà cung cấp sẽ chính thức ký kết SLA và hợp đồng cung cấp dịch vụ. Cuối cùng, cả khách hàng và nhà cung cấp sẽ ký kết hợp đồng chính thức, chính thức hóa mối quan hệ hợp tác.

Thỏa thuận mức dịch vụ (SLA) đóng vai trò quan trọng trong việc thiết lập mối quan hệ tin cậy giữa nhà cung cấp dịch vụ và người tiêu dùng. SLA định rõ các cam kết của nhà cung cấp về khả năng dịch vụ mà họ sẽ cung cấp, đồng thời nêu rõ các yêu cầu

và mong đợi của người tiêu dùng. Tài liệu SLA cần phải bao gồm các vấn đề bảo mật, đảm bảo rằng các cam kết về bảo mật của nhà cung cấp được rõ ràng và minh bạch.

Trong trường hợp của dịch vụ đám mây, SLA giữa nhà cung cấp dịch vụ đám mây và người tiêu dùng phải chi tiết hóa các khả năng bảo mật của giải pháp và các tiêu chuẩn bảo mật mà nhà cung cấp cam kết duy trì để bảo vệ dữ liệu và ứng dụng của người tiêu dùng [80].

### **2.8.3. Đe dọa và tính rủi ro**

Đe dọa là một sự kiện có thể gây hại cho hệ thống, làm giảm độ tin cậy và ảnh hưởng đến tính bảo mật, tính khả dụng, cũng như tính toàn vẹn của thông tin lưu trữ trong hệ thống. Các mối đe dọa này có thể bao gồm nhiều hình thức tấn công khác nhau, chẳng hạn như nghe trộm, gian lận, trộm cắp, phá hoại và các tấn công từ bên ngoài. Rủi ro là khả năng một mối đe dọa khai thác các lỗ hổng trong hệ thống và gây hại cho hệ thống. Rủi ro xảy ra khi mối đe dọa và lỗ hổng hệ thống chồng chéo lên nhau, biến mối đe dọa thành hiện thực [81].

Nghe trộm xảy ra khi các gói dữ liệu được bắt trong quá trình truyền tải mạng để tìm kiếm thông tin nhạy cảm. Gian lận thường liên quan đến các giao dịch nguy hiểm hoặc việc thay đổi dữ liệu để thu lợi bất hợp pháp. Trộm cắp là hành vi ăn cắp bí mật thương mại hoặc dữ liệu để thu lợi, hoặc tiết lộ thông tin trái phép gây thiệt hại.

Phá hoại có thể thực hiện qua các hành động như phá vỡ tính toàn vẹn của dữ liệu, trì hoãn sản xuất, hoặc tấn công từ chối dịch vụ (DDoS) [82]. Tấn công bên ngoài thường liên quan đến việc chèn mã độc hại hoặc virus vào hệ thống hoặc ứng dụng, tạo ra mối đe dọa đối với tính bảo mật của hệ thống. Tất cả những hành động này đều có thể gây tổn hại nghiêm trọng và làm suy yếu hoạt động của hệ thống [81].

### **2.8.4. Mối đe dọa với điện toán đám mây**

Mối đe dọa đối với điện toán đám mây có thể xuất hiện ở nhiều cấp độ khác nhau. Đầu tiên, cơ sở hạ tầng có thể đối mặt với các mối đe dọa bảo mật ở các cấp độ như ứng dụng, máy chủ và mạng [81]. Thứ hai, thông tin của người dùng cũng là mục tiêu của các mối đe dọa, đe dọa sự bảo mật và tính toàn vẹn của dữ liệu [82]. Cuối cùng, việc kiểm soát truy cập là yếu tố quan trọng, đòi hỏi phải có các biện pháp xác thực và quản lý ủy quyền phù hợp để bảo vệ các ứng dụng khỏi sự truy cập trái phép [81] [82].

#### **2.8.4.1. Bảo mật về cơ sở hạ tầng**

Bảo mật trong cơ sở hạ tầng điện toán đám mây đóng vai trò quan trọng trong việc đảm bảo an toàn cho dữ liệu và các tài nguyên hệ thống. Một trong những yếu tố cốt lõi là kiểm soát chặt chẽ quyền truy cập vào các tài nguyên vật lý hỗ trợ cơ sở hạ tầng đám

mây. Bảo mật được triển khai qua ba cấp độ chính: cấp độ mạng, cấp độ máy chủ và cấp độ dịch vụ [83].

Ở cấp độ mạng, mục tiêu là đảm bảo tính bảo mật, tính toàn vẹn và tính sẵn sàng của dữ liệu. Điều này được thực hiện thông qua việc áp dụng các kỹ thuật như mã hóa và chữ ký điện tử. Các biện pháp này thường được triển khai tại chỗ hoặc tại cơ sở của nhà cung cấp, tùy thuộc vào tiềm lực và sự thiết kế của các kiến trúc sư hạ tầng [83].

Tại cấp độ máy chủ, trọng tâm là bảo vệ các máy chủ chuyên dụng dành riêng cho hoạt động điện toán đám mây khỏi những mối đe dọa tiềm ẩn. Với hệ thống đám mây kết nối hàng trăm nút điện toán, các biện pháp bảo mật ở cấp độ này càng trở nên phức tạp và cần thiết để duy trì tính ổn định và hiệu quả của toàn bộ hệ thống [84].

Ở cấp độ dịch vụ, yêu cầu bảo mật thay đổi tùy thuộc vào mô hình dịch vụ được cung cấp. Trong mô hình IaaS người dùng chịu trách nhiệm chính trong việc quản lý và bảo vệ các máy chủ ảo của mình, làm việc song song với nhà cung cấp. Với PaaS, nhiệm vụ bảo mật tập trung vào nền tảng và các ứng dụng mà người dùng triển khai. Trong khi đó, ở SaaS, nhà cung cấp hoàn toàn đảm nhận trách nhiệm bảo mật cho toàn bộ bộ ứng dụng mà họ cung cấp đến người dùng [85].

#### 2.8.4.2. An toàn và bảo mật của hệ thống vật lý

An toàn và bảo mật của hệ thống vật lý là nền tảng quan trọng trong việc bảo vệ cơ sở hạ tầng công nghệ, đặc biệt là các hệ thống điện toán đám mây và trung tâm dữ liệu. Để đảm bảo tính sẵn sàng và toàn vẹn của hệ thống, cần áp dụng một loạt các biện pháp bảo mật vật lý và kỹ thuật [79]. Một số biện pháp được sử dụng phổ biến như hệ thống cung cấp điện liên tục, hệ thống làm mát, quản lý nghiêm ngặt khu vực xung quanh máy chủ, trang bị các thiết bị mạng cần thiết, camera giám sát an ninh 24/7.

Hệ thống cung cấp điện liên tục đóng vai trò không thể thiếu, đảm bảo các thiết bị và máy chủ không bị gián đoạn trong trường hợp mất điện. Ngoài ra, các biện pháp an toàn chống cháy, như hệ thống phát hiện khói, báo cháy và chữa cháy tự động, cần được triển khai để giảm thiểu thiệt hại do hỏa hoạn hoặc thiên tai. Bên cạnh đó việc duy trì điều kiện môi trường lý tưởng cho thiết bị công nghệ cũng là ưu tiên hàng đầu.

Hệ thống làm mát và thông gió thích hợp không chỉ giúp kiểm soát nhiệt độ và độ ẩm mà còn kéo dài tuổi thọ và đảm bảo hiệu suất của thiết bị. Quyền truy cập vật lý vào khu vực máy chủ và các thiết bị liên quan phải được quản lý nghiêm ngặt. Chỉ những nhân viên được ủy quyền mới được phép vào các khu vực này, với các biện pháp kiểm soát bổ sung như hệ thống nhận diện sinh trắc học, thẻ từ hoặc mã PIN. Điều này giúp giảm nguy cơ truy cập trái phép hoặc phá hoại.

Ngoài ra, cần mở rộng các biện pháp bảo vệ vật lý này đến các thiết bị mạng như bộ định tuyến, thiết bị chuyển mạch, và cáp kết nối. Tất cả các thiết bị và đường truyền phải được bảo vệ trong các tủ hoặc hộp chứa an toàn, tránh việc bị tháo gỡ hoặc truy cập trái phép. Để nâng cao mức độ an toàn, các tổ chức có thể triển khai hệ thống giám sát bằng camera an ninh, kết hợp với đội ngũ bảo vệ hoạt động 24/7. Các quy trình kiểm tra định kỳ về an ninh vật lý cần được thực hiện, nhằm phát hiện và xử lý kịp thời các lỗ hổng bảo mật hoặc nguy cơ tiềm ẩn.

## 2.9. Mã hóa dữ liệu đám mây

### 2.9.1. Bảo mật dữ liệu

Mã hóa dữ liệu trong điện toán đám mây là một yếu tố cốt lõi để đảm bảo an toàn và bảo mật dữ liệu, đồng thời tăng cường tính toàn vẹn và tính sẵn có của hệ thống. Quá trình bảo mật dữ liệu bắt đầu với việc xác định những dữ liệu cần mã hóa, đặc biệt khi dữ liệu được truyền qua các môi trường khác nhau như mạng công cộng, mạng riêng hoặc giữa các thiết bị cục bộ [86].

Trong mạng công cộng, việc mã hóa đóng vai trò quan trọng trong việc bảo vệ dữ liệu khi truyền qua internet, nơi nguy cơ bị đánh cắp hoặc tấn công là rất cao. Đối với mạng riêng, các biện pháp bảo mật được thực hiện để duy trì sự an toàn và tính riêng tư trong quá trình truyền dữ liệu. Tương tự, khi dữ liệu được truyền giữa các thiết bị cục bộ như máy tính, thiết bị lưu trữ và thiết bị ngoại vi, mã hóa đảm bảo rằng chỉ những thiết bị được ủy quyền mới có thể truy cập thông tin [85].

Các công nghệ hỗ trợ quá trình mã hóa và bảo mật dữ liệu bao gồm nhiều lĩnh vực như mạng, cơ sở dữ liệu, hệ điều hành, ảo hóa, lập lịch tài nguyên, quản lý giao dịch, cân bằng tải, kiểm soát luồng, và quản lý bộ nhớ. Việc thực thi chính sách bảo mật hiệu quả cũng là yếu tố quan trọng trong quá trình chia sẻ dữ liệu, đảm bảo rằng chỉ những người dùng hoặc hệ thống được phép mới có thể truy cập [87].

Truyền dữ liệu, dù là dữ liệu số hay tương tự, đòi hỏi các biện pháp bảo mật mạnh mẽ để bảo vệ thông tin khi nó di chuyển qua các phương tiện truyền thông đến các thiết bị hoặc mạng. Bảo mật ảo hóa cũng được triển khai để bảo vệ các môi trường ảo hóa, đảm bảo rằng các hệ thống hạ tầng và tài nguyên ảo không bị xâm phạm. An ninh mạng đóng vai trò hỗ trợ toàn diện, đảm bảo rằng tất cả lưu lượng truy cập và tài sản của tổ chức đều được bảo vệ trước các mối đe dọa tiềm tàng.

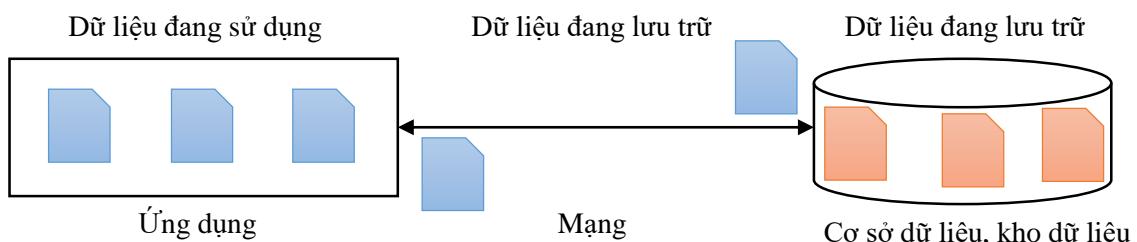
Trong mô hình lưu trữ đám mây, dữ liệu được tổ chức và lưu trữ trong các nhóm logic, giúp tối ưu hóa việc quản lý và truy cập thông tin. Tính sẵn có của dữ liệu, một

yếu tố quan trọng được các nhà cung cấp dịch vụ lưu trữ và bộ nhớ quan tâm, đảm bảo rằng dữ liệu luôn có thể truy cập bất cứ khi nào cần thiết.

### 2.9.2. Kỹ thuật mã hóa đám mây

#### 2.9.2.1. Mã hóa đám mây

Mã hóa đám mây là quá trình chuyển đổi dữ liệu của người dùng thành văn bản mã trước khi lưu trữ trên các dịch vụ đám mây. Kỹ thuật này đảm bảo rằng ngay cả khi dữ liệu bị truy cập trái phép, nội dung thực tế vẫn được bảo vệ. Mã hóa đám mây thường được áp dụng cho dữ liệu đang lưu trữ (Data at Rest), dữ liệu đang truyền tải (Data in Transit), dữ liệu đang sử dụng (Data in Use) [86].



Hình 2.15: Mã hóa dữ liệu

Dữ liệu đang lưu trữ (Data at Rest) là dữ liệu không di chuyển hoặc không đang được truyền tải qua mạng mà được lưu trữ ở trạng thái tĩnh trên các thiết bị lưu trữ [86]. Đây là dạng dữ liệu phổ biến nhất và bao gồm: tệp tin lưu trên ổ cứng, USB, hoặc các thiết bị vật lý; dữ liệu trong cơ sở dữ liệu hoặc hệ thống sao lưu; dữ liệu lưu trữ trên đám mây [87]. Đặc điểm loại này là dữ liệu thường được lưu trữ trong thời gian dài; không hoạt động cho đến khi được truy cập, chỉnh sửa hoặc truyền tải; dễ bị tấn công vật lý hoặc qua việc xâm phạm hệ thống lưu trữ [86] [87].

Trong quá trình truyền tải dữ liệu qua mạng, việc đảm bảo an toàn và bảo mật thông tin là một yếu tố cực kỳ quan trọng. Để thực hiện điều này, các giao thức bảo mật như TLS (Transport Layer Security) được sử dụng để bảo vệ dữ liệu khỏi các cuộc tấn công và xâm nhập không mong muốn. TLS là giao thức mã hóa giúp đảm bảo rằng các thông tin truyền tải giữa các bên giao tiếp (ví dụ, giữa máy chủ và trình duyệt) không thể bị nghe lén hay thay đổi trong quá trình truyền [86] [87].

Dữ liệu đang sử dụng là dữ liệu đang được xử lý, truy cập hoặc thao tác bởi các ứng dụng, hệ thống, hoặc người dùng. Khác với dữ liệu đang lưu trữ (Data at Rest) hoặc dữ liệu đang truyền tải (Data in Transit), dữ liệu đang sử dụng tồn tại ở trạng thái động, thường được xử lý trong bộ nhớ (RAM), CPU, hoặc GPU. Đặc điểm của loại này là thường chỉ tồn tại trong thời gian ngắn, trong lúc hệ thống xử lý dữ liệu; dữ liệu dễ bị

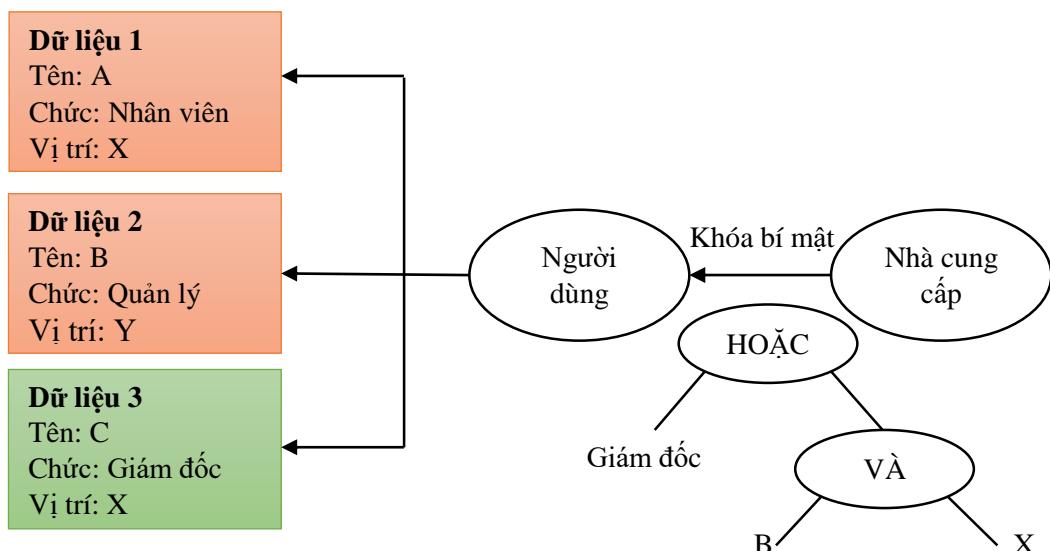
tấn công bởi mã độc, phần mềm gián điệp; khi dữ liệu đang được sử dụng, nó thường được giải mã để xử lý, dẫn đến việc giảm một số lớp bảo vệ [86] [87].

#### 2.9.2.2. Mã hóa dựa trên thuộc tính

Mã hóa dựa trên thuộc tính (Attribute-Based Encryption) là một dạng mã hóa khóa công khai tiên tiến, trong đó việc truy cập và giải mã dữ liệu phụ thuộc vào một tập hợp các thuộc tính [88]. Đây là một kỹ thuật quan trọng trong bảo mật dữ liệu, đặc biệt trong các hệ thống phân tán như điện toán đám mây, nơi dữ liệu cần được chia sẻ có kiểm soát giữa nhiều người dùng [89].

Trong ABE, quyền truy cập dữ liệu không được xác định dựa trên danh tính cụ thể như trong mã hóa truyền thống, mà dựa trên các thuộc tính của người dùng. Nếu thuộc tính của người dùng phù hợp với chính sách giải mã của dữ liệu, họ có thể giải mã và truy cập thông tin [90].

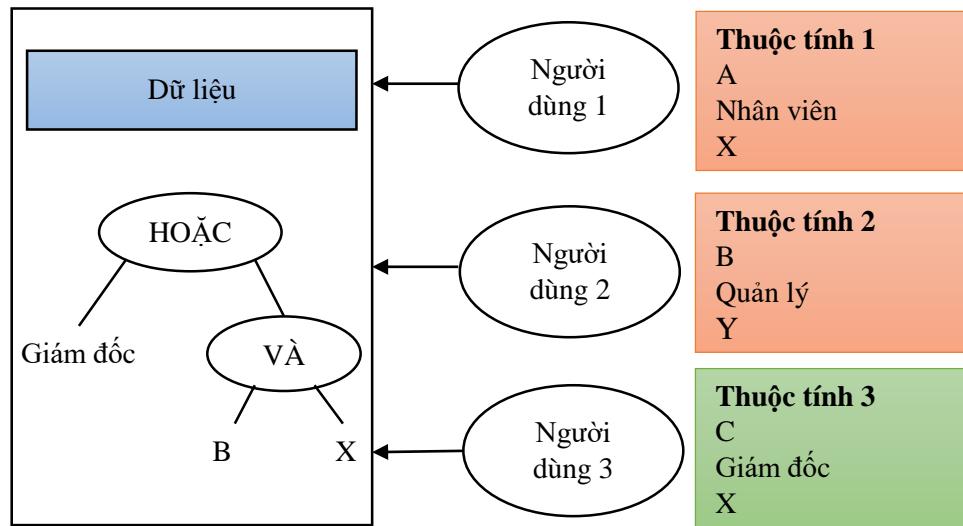
Trong ABE có hai biến thể chính là mã hóa dựa trên thuộc tính chính sách khóa (Key-Policy Attribute-Based Encryption) và mã hóa dựa trên thuộc tính chính sách mã hóa (Attribute-Based Encryption) [90].



Hình 2.16: Mã hóa dựa trên thuộc tính chính sách khóa

Mã hóa dựa trên thuộc tính chính sách khóa hoạt động theo cơ chế trong đó chính sách truy cập được mã hóa vào khóa bí mật của người dùng, còn các thuộc tính được gắn trực tiếp vào dữ liệu. Khi dữ liệu được mã hóa, nó kèm theo một tập thuộc tính cụ thể. Khóa bí mật của người dùng, do nhà cung cấp dịch vụ tạo ra, chứa một cấu trúc truy cập định nghĩa các điều kiện mà dữ liệu có thể được giải mã.

Nếu thuộc tính của dữ liệu khớp với chính sách trong khóa bí mật của người dùng, việc giải mã sẽ thành công. Trong hình 2.15, dữ liệu 1 và dữ liệu 2 không thể giải mã được vì thuộc tính của chúng không thỏa mãn chính sách của người dùng. Tuy nhiên dữ liệu 3 có thể được giải mã vì có thuộc tính ‘Giám đốc’.



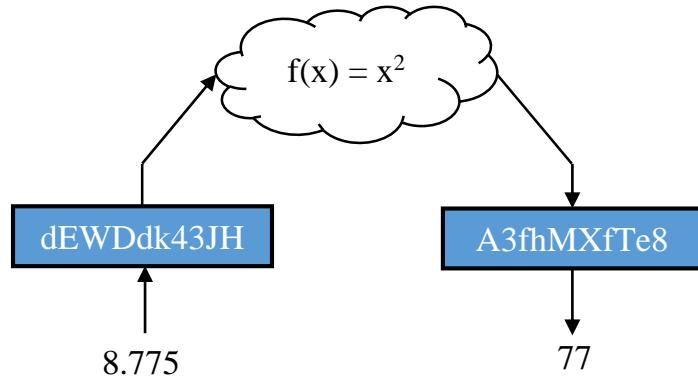
Hình 2.17: Mã hóa dựa trên thuộc tính chính sách mã hóa

Trong mô hình mã hóa dựa trên thuộc tính chính sách mã hóa, khi dữ liệu được mã hóa, người mã hóa định nghĩa một cấu trúc truy cập để xác định những điều kiện mà một người dùng cần thỏa mãn để giải mã dữ liệu. Ví dụ, trong hình 2.16, dữ liệu được mã hóa với cấu trúc truy cập là "Giám đốc HOẶC (B AND X)". Mỗi người dùng được cung cấp một khóa bí mật chứa các thuộc tính của họ. Nếu các thuộc tính này khớp với chính sách truy cập gắn trên dữ liệu, người dùng sẽ giải mã thành công.

### 2.9.3. Bảo mật với cơ sở hạ tầng

#### 2.9.3.1. Mã hóa đồng hình hoàn toàn

Mã hóa đồng hình hoàn toàn (Fully Homomorphic Encryption) là một kỹ thuật mã hóa cho phép thực hiện các phép toán trên dữ liệu đã được mã hóa mà không cần giải mã. Kết quả của các phép toán trên dữ liệu mã hóa, khi được giải mã, sẽ tương đương với kết quả của các phép toán đó trên dữ liệu gốc [91].



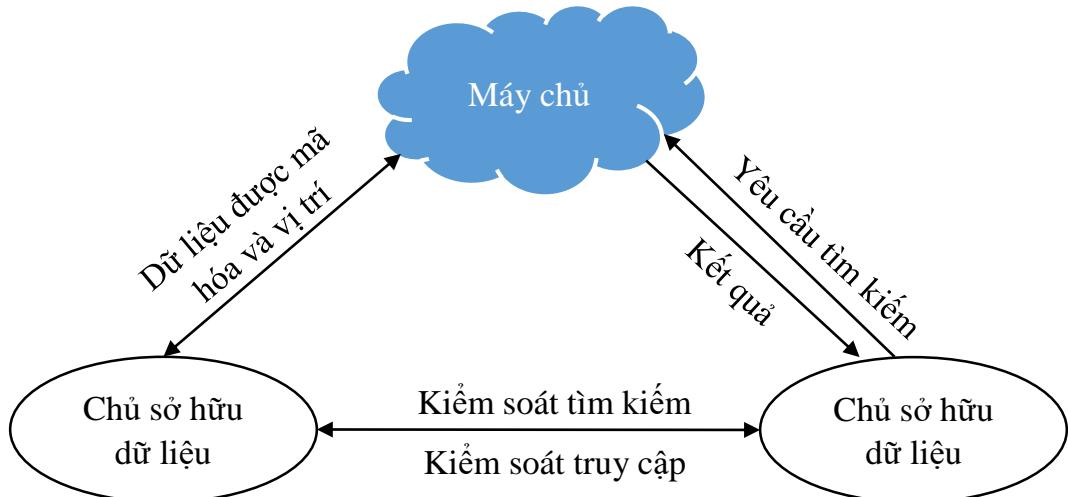
Hình 2.18: Mã hóa đồng hình hoàn toàn

Đặc điểm nổi bật của FHE là tính bảo mật, ở đây dữ liệu luôn được mã hóa trong suốt quá trình xử lý, đảm bảo được tính riêng tư và bảo mật ngay cả khi dữ liệu được xử lý trên các hệ thống không đáng tin cậy. FHE hỗ trợ cả hai loại phép toán cơ bản là cộng và nhân, cho phép xây dựng các ứng dụng phức tạp mà vẫn bảo mật dữ liệu [91] [92].

Tuy nhiên các phép toán thường rất tốn tài nguyên và thời gian để xử lý do tính chất phức tạp của thuật toán, điều này dẫn đến hiệu suất xử lý thấp. Bên cạnh đó, FHE vẫn còn đang nghiên cứu và cải tiến để phù hợp với các ứng dụng quy mô lớn [92].

#### 2.9.3.2. Mã hóa có thể tìm kiếm

Mã hóa có thể tìm kiếm (Searchable Encryption) là một phương pháp mã hóa cho phép người dùng thực hiện các truy vấn tìm kiếm trên dữ liệu đã được mã hóa mà không cần giải mã toàn bộ. Điều này đảm bảo rằng dữ liệu vẫn được bảo mật trong quá trình truy xuất thông tin [93].



Hình 2.19: Mã hóa có thể tìm kiếm

SE được chia thành hai loại: mã hóa tìm kiếm dựa trên khóa bí mật (Symmetric Searchable Encryption) sử dụng chung một khóa bí mật cho cả mã hóa và tìm kiếm, loại

mã hóa này hiệu quả và nhanh chóng, phù hợp với các ứng dụng tin cậy giữa các bên như tổ chức nội bộ. Loại thứ hai là mã hóa có thể tìm kiếm dựa trên khóa công khai (Public-Key Seachable Encryption), một bên sử dụng khóa công khai để mã hóa dữ liệu và một bên khác sử dụng khóa bí mật để thực hiện tìm kiếm và giải mã, phù hợp cho các môi trường không tin cậy với nhiều bên tham gia [94].

Đặc điểm của SE là tìm kiếm an toàn, người tìm kiếm có thể thực hiện truy vấn trên dữ liệu mã hóa mà không làm lộ nội dung của cả dữ liệu lẫn truy vấn. Hiệu quả trong việc tìm kiếm từ khóa hoặc các truy vấn phức tạp trên dữ liệu lớn [93].

## 2.10. Bảo mật hệ thống điều hành

Bảo mật trong hệ điều hành là một yếu tố cốt lõi nhằm đảm bảo rằng nhiều ứng dụng có thể chia sẻ tài nguyên phần cứng trong một hệ thống vật lý một cách an toàn, tuân thủ theo các chính sách được định nghĩa. Điều này không chỉ bao gồm việc bảo vệ tài nguyên hệ thống mà còn cần chặn các cuộc tấn công nguy hiểm nhằm vào ứng dụng. Một trong những thách thức lớn là khả năng dữ liệu được đưa vào hệ thống có thể chứa mã độc, như trường hợp của ứng dụng Java hoặc dữ liệu được trình duyệt tải từ các trang web độc hại [95].

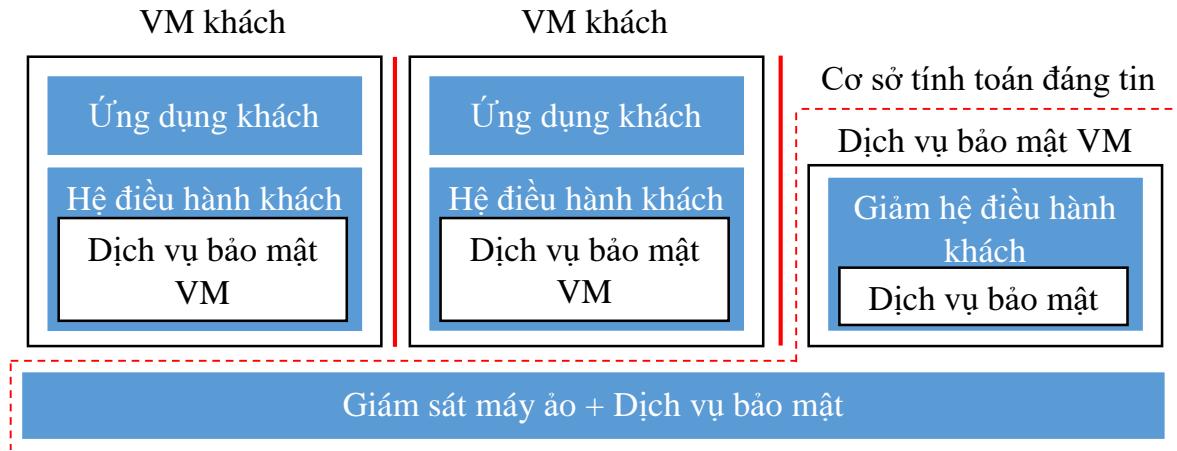
Để đạt được tính bảo mật, các hệ thống con chịu trách nhiệm thực hiện các chức năng liên quan đến bảo mật phải đảm bảo khả năng chống giả mạo và không thể bị vượt qua. Đây là điều kiện cần nhưng không phải điều kiện đủ, bởi hệ điều hành cần đảm bảo rằng mỗi ứng dụng bị giới hạn trong một miền bảo mật duy nhất để tránh các rủi ro tiềm ẩn. Một cách tiếp cận hiệu quả là phân rã các cơ chế bảo mật phức tạp thành các thành phần đơn giản hơn với vai trò được xác định rõ ràng, chẳng hạn như tách riêng cơ chế thực thi và cơ chế quyết định trong kiểm soát truy cập không gian ứng dụng [96].

Tuy nhiên, hiện nay hệ điều hành vẫn tồn tại những hạn chế trong việc cung cấp các cơ chế mạnh mẽ để ứng dụng xác thực lẫn nhau. Ngoài ra, việc thiếu một đường dẫn tin cậy giữa người dùng và ứng dụng tạo ra nhiều thách thức trong việc bảo mật, đặc biệt là trong môi trường máy tính phân tán. Những điểm yếu này đòi hỏi các giải pháp bảo mật toàn diện hơn để đảm bảo tính an toàn cho hệ thống và các ứng dụng hoạt động trên đó [95] [96].

### 2.10.1. Bảo mật máy ảo

Bảo mật cho máy ảo đóng vai trò quan trọng trong việc đảm bảo an toàn dữ liệu và hoạt động của hệ thống ảo hóa [97]. Công nghệ máy ảo (VM) được đánh giá cao nhờ khả năng cung cấp sự cách ly chặt chẽ hơn giữa các máy ảo, vượt trội so với cách ly quy

trình trong hệ điều hành truyền thống. Tuy nhiên, việc triển khai máy ảo cũng đặt ra nhiều thách thức đáng kể [84].



*Hình 2.20: Kiến trúc dịch vụ bảo mật ảo*

Một trong những vấn đề chính là chi phí phần cứng cao hơn, vì các hệ thống ảo thường yêu cầu tài nguyên lớn như chu kỳ CPU, bộ nhớ, đĩa và băng thông mạng [98]. Đồng thời, việc phát triển bộ hypervisor và sửa đổi hệ điều hành chủ trong trường hợp ảo hóa cũng tạo thêm áp lực tài chính và kỹ thuật. Chi phí vận hành tăng lên do hypervisor liên quan đến các hoạt động đặc quyền, làm giảm hiệu suất tổng thể của hệ thống [99].

Bên cạnh đó, việc phân bổ tài nguyên không hợp lý hoặc thiếu tài nguyên có thể dẫn đến tình trạng từ chối dịch vụ cho một số máy ảo, gây gián đoạn hoạt động [97]. Các cuộc tấn công qua kênh bên máy ảo cũng là một nguy cơ lớn, khi kẻ tấn công khai thác thông tin từ các hoạt động như thời gian xử lý hoặc băng thông để xâm nhập hệ thống. Ngoài ra, sự tồn tại của máy ảo giả mạo hoặc không an toàn trong môi trường triển khai có thể làm gia tăng rủi ro bảo mật, cùng với nguy cơ hình ảnh máy ảo bị giả mạo hoặc chứa mã độc trong kho lưu trữ hình ảnh [98].

Những thách thức này nhấn mạnh tầm quan trọng của việc thiết kế các chiến lược bảo mật mạnh mẽ cho hệ thống ảo hóa. Điều này bao gồm không chỉ việc cải thiện khả năng cách ly và bảo vệ dữ liệu, mà còn tăng cường các biện pháp giám sát và xác thực để ngăn chặn sự xâm nhập và đảm bảo an toàn cho toàn bộ môi trường ảo hóa [98].

### 2.10.2. Mối đe dọa bảo mật

Hệ điều hành quản lý, thường đóng vai trò như một phần trung gian hỗ trợ hypervisor, có thể trở thành nguồn gốc của nhiều mối đe dọa bảo mật đối với các hệ thống máy ảo [100]. Hypervisor dựa vào hệ điều hành quản lý để thực hiện các nhiệm vụ quan trọng, chẳng hạn như tạo máy ảo mới và quản lý luồng dữ liệu giữa máy ảo

khách và các thiết bị lưu trữ hoặc giao diện mạng [101]. Ngoài ra, hệ điều hành quản lý còn hỗ trợ các công cụ quản trị, quá trình di chuyển trực tiếp, trình điều khiển thiết bị, và trình giả lập thiết bị, khiến nó trở thành một mắt xích nhạy cảm trong chuỗi bảo mật.

Một số quy trình liên quan đến việc quản lý tài nguyên máy ảo có thể bị khai thác nếu không được bảo vệ đúng cách. Khi phân bổ bộ nhớ trong không gian địa chỉ Dom0, hệ điều hành quản lý phải tải hạt nhân của hệ điều hành khách từ bộ nhớ thứ cấp. Tiếp đó, nó phải phân bổ bộ nhớ cho máy ảo mới và sử dụng cơ chế ánh xạ nước ngoài để tải hạt nhân vào máy ảo, thiết lập các bảng trang ban đầu, và sau đó giải phóng ánh xạ ngoại để kích hoạt các thanh ghi CPU ảo và khởi chạy máy ảo mới [98].

Những hoạt động này, nếu không được kiểm soát, có thể dẫn đến các lỗ hổng bảo mật. Để triển khai một hệ thống thời gian chạy an toàn, cần phải giám sát và kiểm soát chặt chẽ các siêu cuộc gọi được sử dụng trong quá trình giao tiếp giữa một Dom0 không đáng tin cậy và một DomU mà hệ thống cần bảo vệ [99]. Điều này nhằm đảm bảo tính riêng tư và toàn vẹn của CPU máy ảo cũng như bộ nhớ ảo của VM. Đồng thời, cần duy trì sự mới mẻ của CPU ảo và bộ nhớ của máy ảo thông qua các cơ chế bổ sung, chẳng hạn như thêm mã băm hoặc số phiên bản. Những biện pháp này không chỉ cải thiện tính toàn vẹn mà còn bảo vệ hệ thống khỏi các cuộc tấn công tiềm năng trong môi trường ảo hóa [100].

## 2.11. Bảo mật cho giải pháp ảo hóa

Giải pháp ảo hóa mang lại nhiều lợi ích cho bảo mật nhưng cũng đi kèm với những thách thức và ảnh hưởng đến hiệu suất. Một trong những tác động tiêu cực là chi phí bổ sung làm giảm hiệu suất, đồng thời đòi hỏi các hệ thống phần cứng mạnh mẽ hơn để có thể vận hành nhiều máy ảo một cách hiệu quả. Trong mô hình IaaS, ảo hóa cung cấp một nền tảng linh hoạt, cho phép người dùng chọn và tải lên các ảnh máy ảo phù hợp với môi trường ứng dụng cục bộ, sau đó triển khai và chạy ứng dụng trên đám mây.

Ngoài ra, ảo hóa còn tăng cường độ tin cậy hệ thống. Toàn bộ hệ điều hành cùng với các ứng dụng có thể được sao chép và đưa vào chế độ chờ nóng, đảm bảo rằng trong trường hợp xảy ra sự cố, hệ thống có thể nhanh chóng được khôi phục. Điều này cũng hỗ trợ các chính sách quản lý tài nguyên một cách hiệu quả, như di chuyển các máy ảo sang các máy chủ khác để cân bằng tải trong hệ thống [102].

Khả năng phát hiện xâm nhập cũng được cải thiện trong môi trường ảo hóa. Các bản sao máy ảo có thể tìm kiếm các mẫu hoạt động bất thường và phát hiện kịp thời các dấu hiệu xâm nhập. Tuy nhiên, để đảm bảo tính bảo mật, cần triển khai các cơ chế bảo vệ nhật ký và chống xâm nhập hiệu quả, vì khi các tính năng phát hiện xâm nhập được

triển khai ở cấp hệ điều hành, chúng có thể bị vô hiệu hóa hoặc nhật ký bị kẻ tấn công chỉnh sửa.

Cuối cùng, ảo hóa mang lại sự linh hoạt vượt trội trong việc bảo trì và kiểm tra phần mềm. Toàn bộ quá trình này trở nên dễ dàng và hiệu quả hơn nhờ vào khả năng tách biệt và quản lý các máy ảo độc lập trong hệ thống, giúp đảm bảo tính bảo mật và hiệu quả hoạt động tổng thể [102].

### **2.11.1. Mối đe dọa**

Các mối đe dọa bảo mật trong môi trường ảo hóa bắt nguồn từ nhiều yếu tố khác nhau, phản ánh sự phức tạp và các thách thức trong việc bảo vệ các hệ thống ảo hóa. Một trong những đặc điểm quan trọng của ảo hóa là khả năng chia sẻ tài nguyên trên một máy chủ vật lý duy nhất, giúp tối ưu hóa chi phí. Tuy nhiên, điều này đồng thời tạo ra nguy cơ khi nhiều máy ảo chạy trên cùng một bộ tài nguyên vật lý, dẫn đến khả năng xảy ra xung đột hoặc lỗ hổng bảo mật [99].

Trình ảo hóa, đóng vai trò như trung tâm điều phối các tài nguyên và quản lý máy ảo, là một mục tiêu hấp dẫn cho các cuộc tấn công. Bảo mật và tính ổn định của toàn bộ môi trường ảo hóa phụ thuộc rất nhiều vào khả năng tự bảo vệ của trình ảo hóa trước các mối đe dọa [103]. Tuy nhiên, chính việc thêm lớp trừu tượng của ảo hóa cũng làm gia tăng sự phức tạp trong cấu hình và quản lý hệ thống, mở rộng không gian tiềm ẩn các lỗi cấu hình và lỗ hổng bảo mật [103].

Một vấn đề đáng lo ngại khác là sự phân cấp đặc quyền trong hệ thống. Các cuộc tấn công leo thang đặc quyền có thể xảy ra khi người dùng hoặc ứng dụng vượt qua giới hạn quyền hạn được cấp, thường do các lỗi thiết kế hoặc lỗ hổng trong hệ thống. Điều này đặc biệt nghiêm trọng vì nó cho phép kẻ tấn công truy cập trái phép vào nhiều tài nguyên hoặc chức năng quan trọng [102].

Một thách thức lớn khác là việc hợp nhất các vùng tin cậy khác nhau lên cùng một hệ thống vật lý mà không có sự tách biệt rõ ràng. Khối lượng công việc từ các khu vực với cấp độ tin cậy khác nhau có thể dẫn đến sự xung đột và gia tăng nguy cơ xâm nhập nếu không có cơ chế phù hợp. Những mối đe dọa này nhấn mạnh sự cần thiết phải xây dựng các chiến lược bảo mật toàn diện và hiệu quả trong môi trường ảo hóa [103].

### **2.11.2. Khuyến nghị**

Để đảm bảo an toàn trong môi trường ảo hóa, việc triển khai các khuyến nghị bảo mật đóng vai trò then chốt trong việc giảm thiểu rủi ro. Trước hết, làm cứng máy ảo là một biện pháp thiết yếu, vì trong hệ thống ảo hóa máy chủ, người dùng chỉ có thể truy cập gián tiếp vào tài nguyên thông qua các máy ảo. Việc này đòi hỏi cấu hình máy ảo

sao cho giảm thiểu các lỗ hổng và đảm bảo rằng chúng được bảo vệ trước các cuộc tấn công tiềm ẩn [98].

Làm cứng trình ảo hóa cũng là một bước quan trọng, bởi trình ảo hóa là thành phần trung tâm trong môi trường ảo hóa, chịu trách nhiệm điều phối giao tiếp giữa các máy ảo và tài nguyên vật lý [98]. Nếu trình ảo hóa bị tấn công, toàn bộ hệ thống ảo hóa có thể bị xâm phạm. Hạn chế quyền truy cập vật lý vào máy chủ là một biện pháp cơ bản nhưng không kém phần quan trọng [103]. Một lỗ hổng vật lý trên máy chủ có thể gây nguy hiểm cho toàn bộ môi trường ảo hóa, do đó cần có các chính sách kiểm soát truy cập nghiêm ngặt đối với phần cứng máy chủ.

Việc giới hạn mỗi máy ảo thực hiện một chức năng chính duy nhất cũng là một giải pháp để tăng cường bảo mật. Dù các máy ảo có khả năng xử lý nhiều nhiệm vụ, việc phân chia quy trình chính giữa các máy ảo giúp giảm thiểu rủi ro khi một máy ảo bị xâm phạm sẽ không ảnh hưởng đến các chức năng khác [100]. Sử dụng các cơ chế truyền thông bảo mật là điều cần thiết để bảo vệ dữ liệu trong quá trình truyền tải giữa các hệ thống. Điều này đảm bảo rằng thông tin được mã hóa và an toàn trước sự can thiệp từ bên ngoài.

## Tiêu kết chương

Truy nhập và bảo mật dữ liệu là những thành phần cốt lõi, đóng vai trò quyết định trong việc đảm bảo hiệu quả hoạt động của các điện toán đám mây. Chương này đã trình bày chi tiết các mô hình lưu trữ phổ biến như lưu trữ ô, lưu trữ nhật ký và các hệ thống lưu trữ phân tán, cùng những công nghệ tiên tiến như HDFS, GFS và cơ sở dữ liệu NoSQL, bên cạnh đó các biện pháp bảo mật ảo hóa trong điện toán đám mây.

Tóm lại, lưu trữ và bảo mật dữ liệu trên đám mây không chỉ là giải pháp công nghệ hiện đại mà còn là yếu tố nền tảng giúp doanh nghiệp và tổ chức khai thác tối đa tiềm năng của điện toán đám mây. Để đạt được hiệu quả tối ưu, việc triển khai và quản lý hệ thống lưu trữ cần phải được thực hiện một cách chiến lược, đồng thời xử lý tốt các vấn đề liên quan đến bảo mật và hiệu suất, nhằm đảm bảo tính ổn định và hiệu quả lâu dài.

## CHƯƠNG 3. TRIỂN KHAI WEBSITE BÁN GAMING GEAR TRÊN AWS

Trong chương này sẽ giới thiệu một số nền tảng điện toán đám mây phổ biến và triển khai website bán Gaming Gear trên nền tảng Amazon Web Service.

### 3.1. Amazon Web Service

#### 3.1.1. Lịch sử ra đời

Amazon Web Services (AWS) là một trong những nền tảng điện toán đám mây lớn nhất và thành công nhất trên thế giới. Được sáng lập và phát triển bởi Amazon.com, AWS cung cấp các dịch vụ đám mây tiên tiến, từ cơ sở hạ tầng tính toán (IaaS) cho đến các nền tảng phát triển ứng dụng và trí tuệ nhân tạo. Lịch sử ra đời và phát triển của AWS không chỉ phản ánh sự sáng tạo của Amazon mà còn đánh dấu một bước ngoặt trong ngành công nghiệp điện toán đám mây [104].

Khởi nguồn của AWS từ những năm 2002, khi Amazon nhận thấy nhu cầu ngày càng tăng về các dịch vụ điện toán, đặc biệt là trong việc xây dựng hạ tầng IT cho các công ty khởi nghiệp và các doanh nghiệp lớn. Trước đó, Amazon chỉ đơn giản là một công ty bán lẻ trực tuyến, nhưng người sáng lập Jeff Bezos nhận thấy rằng Amazon đã sở hữu một hệ thống cơ sở hạ tầng máy chủ rất mạnh mẽ, mà nó có thể chia sẻ với các công ty khác, tạo thành một dịch vụ điện toán đám mây [104] [105].

AWS chính thức ra mắt vào tháng 3 năm 2006 với các dịch vụ cơ bản như Amazon S3 (Simple Storage Service) và EC2 (Elastic Compute Cloud), đánh dấu một bước đột phá trong ngành điện toán đám mây. Những dịch vụ này cho phép doanh nghiệp thuê tài nguyên máy chủ từ Amazon mà không cần đầu tư vào cơ sở hạ tầng đắt đỏ. Với S3, người dùng có thể lưu trữ và quản lý dữ liệu quy mô lớn, trong khi EC2 cung cấp khả năng tính toán linh hoạt, cho phép thuê máy chủ với sức mạnh tính toán theo nhu cầu thực tế mà không phải đầu tư vào phần cứng [104].

Từ 2006 đến 2010, AWS không ngừng mở rộng các dịch vụ, bổ sung nhiều tính năng mới như Amazon RDS (Relational Database Service), Amazon SQS (Simple Queue Service) và các dịch vụ CDN như CloudFront. Mỗi dịch vụ được ra mắt đều mở rộng khả năng của AWS, không chỉ cung cấp dịch vụ lưu trữ và tính toán, mà còn bao gồm các lĩnh vực như cơ sở dữ liệu, phân phối nội dung và mạng, đáp ứng nhu cầu ngày càng đa dạng của các doanh nghiệp và tổ chức [104].

#### 3.1.2. Khái niệm

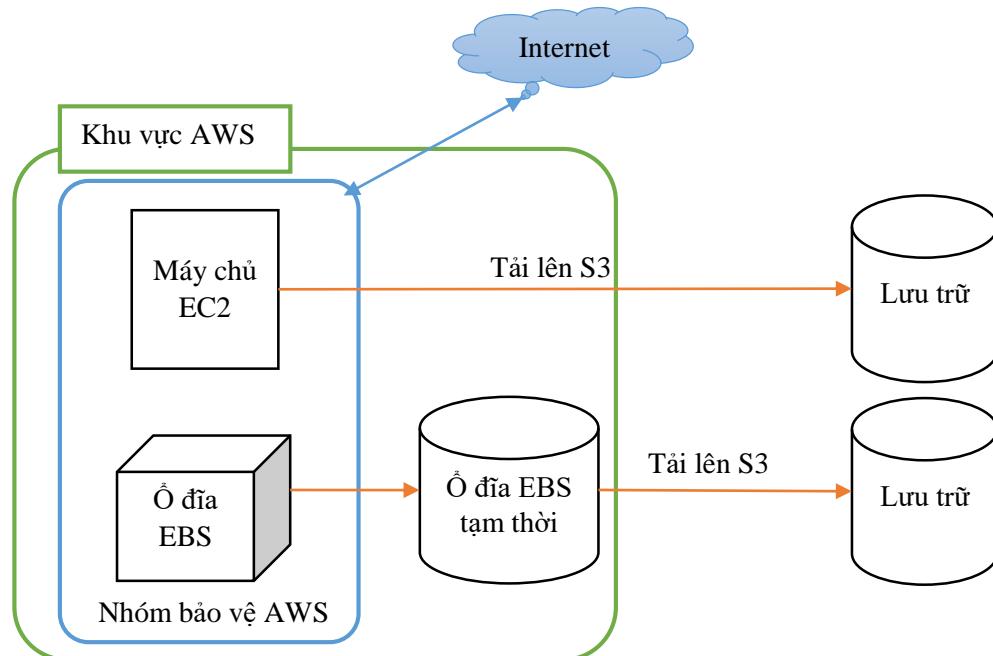
AWS là nền tảng điện toán đám mây do Amazon phát triển, ra mắt vào năm 2006, cung cấp các dịch vụ như tính toán, lưu trữ, cơ sở dữ liệu, trí tuệ nhân tạo và nhiều lĩnh vực khác. AWS giúp cá nhân và doanh nghiệp triển khai, quản lý và mở rộng ứng dụng

một cách hiệu quả, linh hoạt mà không cần phải sở hữu hạ tầng vật lý. AWS hoạt động theo mô hình thanh toán theo mức sử dụng (pay-as-you-go), giúp giảm chi phí ban đầu và tối ưu hóa hiệu quả vận hành. AWS không chỉ cung cấp dịch vụ hạ tầng mà còn hỗ trợ các công cụ phát triển, tự động hóa và bảo mật mạnh mẽ [106].

Với các trung tâm dữ liệu đặt tại nhiều khu vực trên thế giới, AWS đảm bảo độ tin cậy cao và độ trễ thấp. Nền tảng này là lựa chọn lý tưởng cho các công ty khởi nghiệp, doanh nghiệp vừa và nhỏ, cũng như các doanh nghiệp lớn. AWS cho phép các doanh nghiệp tùy chọn dịch vụ phù hợp với nhu cầu mà không cần đầu tư lớn vào cơ sở hạ tầng ban đầu [105].

### 3.1.3. Dịch vụ

Amazon Web Services (AWS) là một trong những nền tảng đám mây phổ biến nhất hiện nay, cung cấp một loạt các dịch vụ từ tính toán, lưu trữ, cơ sở dữ liệu, đến trí tuệ nhân tạo (AI), phân tích dữ liệu và bảo mật. AWS giúp các doanh nghiệp triển khai các ứng dụng và dịch vụ với tính linh hoạt cao, khả năng mở rộng, và hiệu quả chi phí.



Hình 3.1: Kiến trúc cơ bản của AWS

AWS cung cấp các dịch vụ tính toán mạnh mẽ như Amazon EC2 (Elastic Compute Cloud), cho phép người dùng thuê máy chủ ảo và tùy chỉnh theo nhu cầu sử dụng. EC2 hỗ trợ nhiều loại instance với các cấu hình khác nhau, giúp tối ưu hóa hiệu suất và chi phí cho từng ứng dụng. Bên cạnh đó, AWS Lambda cung cấp dịch vụ tính toán không máy chủ, giúp nhà phát triển chỉ cần tập trung vào việc viết mã mà không phải lo lắng

về việc quản lý hạ tầng. Các dịch vụ khác như Amazon ECS và Amazon EKS hỗ trợ triển khai các ứng dụng containerized với Docker và Kubernetes [104].

AWS cung cấp các dịch vụ lưu trữ mạnh mẽ và linh hoạt, bao gồm Amazon S3 (Simple Storage Service), giúp lưu trữ dữ liệu ở quy mô lớn với khả năng mở rộng không giới hạn và độ bền cao. S3 thường được sử dụng cho các mục đích như sao lưu, lưu trữ dữ liệu tĩnh và phân phối nội dung. Amazon EBS (Elastic Block Store) cung cấp lưu trữ block cho các máy chủ EC2, hỗ trợ các ứng dụng yêu cầu hiệu suất truy xuất nhanh chóng. Đối với các doanh nghiệp cần lưu trữ dữ liệu lâu dài, Amazon Glacier là dịch vụ lý tưởng với chi phí thấp nhưng thời gian truy xuất dữ liệu chậm [104] [105].

AWS cung cấp các dịch vụ cơ sở dữ liệu mạnh mẽ, từ các hệ quản trị cơ sở dữ liệu quan hệ đến NoSQL. Amazon RDS (Relational Database Service) giúp quản lý các cơ sở dữ liệu quan hệ như MySQL, PostgreSQL và SQL Server, tự động hóa các tác vụ như sao lưu và vá lỗi. Đối với các ứng dụng cần xử lý dữ liệu phi cấu trúc hoặc khối lượng lớn, Amazon DynamoDB là dịch vụ cơ sở dữ liệu NoSQL lý tưởng với độ trễ thấp và khả năng mở rộng tự động. Amazon Aurora, một dịch vụ cơ sở dữ liệu quan hệ với hiệu suất cao, cũng được AWS cung cấp để đáp ứng các nhu cầu về quy mô lớn [104].

AWS cung cấp các dịch vụ phân tích mạnh mẽ giúp doanh nghiệp xử lý và phân tích dữ liệu lớn. Amazon EMR (Elastic MapReduce) hỗ trợ các framework như Hadoop và Spark, giúp xử lý khối lượng công việc phân tán một cách nhanh chóng và tiết kiệm chi phí. Amazon Redshift là một dịch vụ kho dữ liệu mạnh mẽ, giúp phân tích dữ liệu lớn với hiệu suất cao và khả năng tích hợp với các công cụ phân tích khác [105].

Bảo mật là một yếu tố quan trọng khi sử dụng các dịch vụ đám mây, và AWS cung cấp nhiều công cụ để bảo vệ dữ liệu và ứng dụng của người dùng. Identity and Access Management (IAM) giúp kiểm soát quyền truy cập vào tài nguyên AWS, trong khi AWS Shield bảo vệ các ứng dụng khỏi các cuộc tấn công từ chối dịch vụ (DDoS). AWS CloudTrail giúp giám sát và ghi lại các hoạt động trong tài khoản AWS, giúp phát hiện các hành vi bất thường và đảm bảo tính bảo mật cho hệ thống. Các công cụ bảo mật này giúp doanh nghiệp duy trì an toàn cho dữ liệu và tuân thủ các quy định về bảo mật [104].

### 3.2. Microsoft Azure

#### 3.2.1. Lịch sử ra đời

Microsoft Azure, còn được biết đến là Windows Azure trước khi đổi tên vào năm 2014, là nền tảng đám mây toàn diện của Microsoft. Sự ra đời và phát triển của Azure đánh dấu một cột mốc quan trọng trong chiến lược điện toán đám mây của Microsoft, đưa công ty này vào cuộc đua với các đối thủ lớn như AWS và GCP [107].

Ý tưởng về Microsoft Azure bắt đầu hình thành vào giữa những năm 2000 khi Microsoft nhận thấy sự tiềm năng to lớn của điện toán đám mây trong tương lai. Lúc bấy giờ, phần lớn các công ty công nghệ lớn đều chú trọng vào việc cung cấp các dịch vụ dựa trên đám mây, nhưng Microsoft vẫn chưa có một nền tảng mạnh mẽ để cạnh tranh. Trái ngược với các đối thủ như Amazon, vốn đã có AWS ra mắt từ năm 2006, Microsoft bắt đầu phát triển một nền tảng đám mây riêng biệt, với mục tiêu cung cấp các dịch vụ lưu trữ, tính toán và phân tích dữ liệu [107].

Microsoft công bố Windows Azure vào tháng 10 năm 2008 tại một sự kiện ở San Francisco. Mặc dù ban đầu chỉ cung cấp dịch vụ lưu trữ và tính toán cơ bản, Azure nhanh chóng chứng minh sức mạnh với hệ thống mạnh mẽ, được hỗ trợ bởi các máy chủ và trung tâm dữ liệu của Microsoft. Vào tháng 2 năm 2010, Azure ra mắt phiên bản đầu tiên (CTP - Community Technology Preview), cho phép các nhà phát triển bắt đầu triển khai ứng dụng trên nền tảng đám mây. Sự ra mắt này đánh dấu sự gia nhập của Microsoft vào thị trường đám mây, mở ra những cơ hội phát triển và đổi mới lớn cho Azure [107].

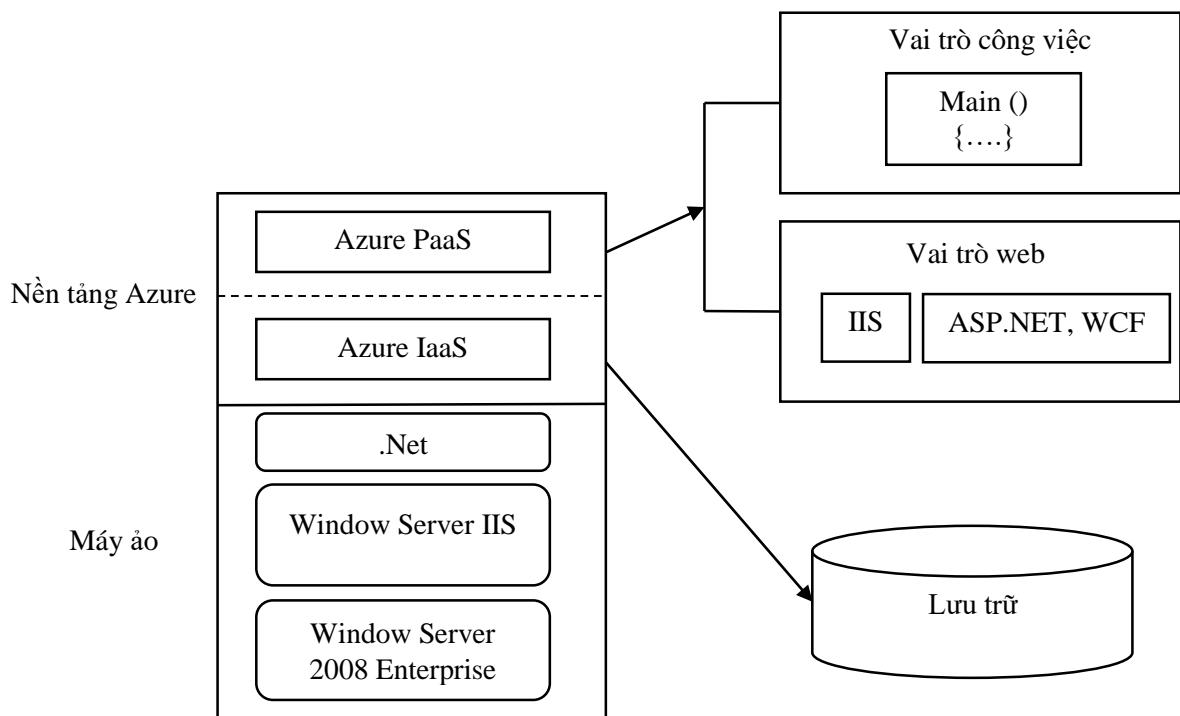
### **3.2.2. Khái niệm**

Microsoft Azure, trước đây gọi là Windows Azure, là nền tảng điện toán đám mây do Microsoft phát triển và ra mắt vào năm 2010. Azure cung cấp các dịch vụ toàn diện để xây dựng, triển khai và quản lý ứng dụng cùng hạ tầng trên đám mây. Nền tảng này hoạt động dựa trên mô hình điện toán đám mây, cung cấp ba loại dịch vụ chính: IaaS, PaaS, và SaaS, cho phép doanh nghiệp tiếp cận tài nguyên công nghệ mà không cần đầu tư vào cơ sở hạ tầng vật lý và mở rộng nhanh chóng khi cần [108].

Một điểm mạnh của Microsoft Azure là khả năng tích hợp mạnh mẽ với các sản phẩm khác của Microsoft như Windows Server, SQL Server, Active Directory và Microsoft 365. Azure hỗ trợ nhiều ngôn ngữ lập trình và công cụ phát triển như .NET, Java, Python và Node.js, đáp ứng nhu cầu đa dạng của doanh nghiệp. Với cơ sở hạ tầng toàn cầu, Azure có hơn 60 vùng dữ liệu trên toàn thế giới, đảm bảo độ tin cậy cao, độ trễ thấp và tuân thủ các quy định địa phương [108].

### **3.2.3. Dịch vụ**

Azure cung cấp dịch vụ tính toán mạnh mẽ thông qua các máy chủ ảo (Virtual Machines - VMs). Các máy chủ này cho phép doanh nghiệp triển khai và quản lý các ứng dụng trên đám mây mà không cần đầu tư vào phần cứng vật lý. Với Azure, người dùng có thể chọn lựa các hệ điều hành, cấu hình phần cứng, và cấu hình mạng cho máy ảo của mình. Dịch vụ này cung cấp khả năng mở rộng linh hoạt, cho phép doanh nghiệp tăng hoặc giảm tài nguyên tính toán tùy theo nhu cầu [109].



Hình 3.2: Kiến trúc cơ bản của Microsoft Azure

Lưu trữ là một trong những yếu tố cốt lõi của các dịch vụ đám mây, và Azure cung cấp một loạt các giải pháp lưu trữ để đáp ứng nhu cầu đa dạng của người dùng. Các dịch vụ lưu trữ của Azure bao gồm Azure Blob Storage, Azure Disk Storage, và Azure File Storage. Từ lưu trữ dữ liệu không có cấu trúc như hình ảnh và video (Blob Storage), cho đến lưu trữ dữ liệu đĩa cho các máy ảo (Disk Storage), và cung cấp chia sẻ tệp tin thông qua giao thức SMB (File Storage) [110].

Azure cung cấp một loạt dịch vụ cơ sở dữ liệu để quản lý dữ liệu cho các ứng dụng đám mây, bao gồm Azure SQL Database, một dịch vụ cơ sở dữ liệu quan hệ được quản lý hoàn toàn, giúp giảm thiểu công việc quản lý cơ sở hạ tầng cho người dùng. Ngoài SQL, Azure còn cung cấp dịch vụ cơ sở dữ liệu NoSQL như Azure Cosmos DB, được tối ưu cho các ứng dụng yêu cầu khả năng mở rộng toàn cầu và tốc độ truy vấn nhanh chóng [109]. Các dịch vụ này giúp các tổ chức dễ dàng xây dựng các ứng dụng dựa trên dữ liệu mà không phải lo lắng về việc duy trì hoặc mở rộng cơ sở hạ tầng cơ sở dữ liệu.

Các dịch vụ mạng của Azure cho phép kết nối các ứng dụng và hệ thống trên đám mây một cách dễ dàng và an toàn. Azure Virtual Network cho phép người dùng thiết lập mạng riêng ảo, giúp kết nối các máy chủ ảo, các dịch vụ trên Azure và các hệ thống của doanh nghiệp một cách an toàn. Azure còn cung cấp dịch vụ Azure Load Balancer để phân phối lưu lượng truy cập đến các máy chủ đám mây, và Azure CDN (Content Delivery Network) để tăng tốc độ tải nội dung cho người dùng trên toàn cầu [109] [110].

Bảo mật luôn là yếu tố quan trọng trong các nền tảng đám mây, và Azure cung cấp nhiều công cụ bảo mật giúp người dùng bảo vệ các ứng dụng và dữ liệu của mình. Azure Security Center là một công cụ quản lý bảo mật toàn diện, giúp giám sát và bảo vệ môi trường đám mây. Bên cạnh đó, Azure Key Vault giúp quản lý các khóa mật mã và chứng chỉ an toàn, còn Azure Active Directory (Azure AD) cung cấp dịch vụ xác thực và quyền truy cập, giúp quản lý người dùng và bảo vệ các tài nguyên đám mây [110].

Microsoft Azure cung cấp các công cụ để hỗ trợ quá trình phát triển và triển khai ứng dụng, như Azure DevOps Services, một bộ công cụ hỗ trợ quy trình DevOps, từ lập kế hoạch đến triển khai ứng dụng. Azure cũng cung cấp Azure Monitor, giúp giám sát và thu thập dữ liệu từ các ứng dụng và hạ tầng trên Azure, và Azure Automation, giúp tự động hóa các tác vụ quản trị và triển khai, nâng cao hiệu suất và giảm thiểu lỗi [110].

### 3.3. Google App Engine

#### 3.3.1. Lịch sử ra đời

Google App Engine (GAE) ra đời như một bước đi tiên phong của Google trong lĩnh vực điện toán đám mây, nhằm đáp ứng nhu cầu ngày càng lớn của các nhà phát triển về một nền tảng triển khai ứng dụng dễ sử dụng, không cần quản lý hạ tầng. Được giới thiệu lần đầu vào tháng 4 năm 2008 tại sự kiện Google I/O, GAE ban đầu là một dự án thử nghiệm với sự hỗ trợ ngôn ngữ Python và chỉ cung cấp cho 10.000 nhà phát triển nhằm thu thập phản hồi. Đây là một trong những sản phẩm cốt lõi đầu tiên của Google Cloud Platform (GCP), đặt nền móng cho sự phát triển mạnh mẽ sau này.

Vào năm 2009, GAE chính thức ra mắt phiên bản sản xuất, bổ sung hỗ trợ ngôn ngữ Java, một ngôn ngữ phổ biến trong các ứng dụng doanh nghiệp. Điều này không chỉ mở rộng đối tượng người dùng của GAE mà còn giúp nền tảng này được các công ty lớn chấp nhận. Cùng với việc hỗ trợ Java, Google cũng triển khai các tính năng tự động hóa, bao gồm tự động mở rộng tài nguyên (auto-scaling), giảm thiểu gánh nặng quản lý cho các nhà phát triển và giúp họ tập trung vào việc phát triển ứng dụng thay vì lo lắng về cơ sở hạ tầng.

Đến năm 2011, GAE mở rộng khả năng của mình khi bổ sung ngôn ngữ lập trình Go, do Google phát triển, giúp nền tảng này xử lý hiệu suất cao và tính năng đồng thời, phù hợp với các ứng dụng khói lượng công việc lớn. Vào năm 2013, GAE được tích hợp vào hệ sinh thái GCP, kết nối dễ dàng với các dịch vụ mạnh mẽ như Google Cloud Storage, BigQuery và Google AI. Sự tích hợp này mang lại lợi ích to lớn cho nhà phát triển và giúp Google cạnh tranh trực tiếp với các đối thủ lớn như AWS và Microsoft Azure.

### **3.3.2. *Khái niệm***

Google App Engine (GAE) là nền tảng dịch vụ đám mây PaaS do Google phát triển, cho phép các nhà phát triển xây dựng và triển khai ứng dụng web và di động mà không cần quản lý máy chủ vật lý. Hệ thống tự động quản lý các yếu tố như cân bằng tải, mở rộng tài nguyên, giám sát hiệu suất và sao lưu dữ liệu. GAE hỗ trợ nhiều ngôn ngữ lập trình như Python, Java, Go, PHP, Node.js, Ruby và C# thông qua các runtime tích hợp hoặc tùy chỉnh [111].

Một điểm mạnh của GAE là khả năng tích hợp mạnh mẽ với các dịch vụ khác trong hệ sinh thái Google Cloud, như Google Cloud Storage, BigQuery và Google AI, giúp các doanh nghiệp phát triển các ứng dụng từ đơn giản đến phức tạp mà không phải lo lắng về khả năng mở rộng hay độ tin cậy [112].

### **3.3.3. *Dịch vụ***

Một trong những tính năng nổi bật của GAE chính là khả năng tự động hóa. Khi triển khai ứng dụng trên nền tảng này, các nhà phát triển không cần lo lắng về việc quản lý hạ tầng hoặc duy trì các máy chủ vật lý. GAE tự động xử lý mọi yếu tố liên quan đến việc mở rộng tài nguyên, cập nhật hệ thống và giám sát hiệu năng. Khi lưu lượng truy cập tăng lên, GAE có thể tự động phân bổ tài nguyên bổ sung để đảm bảo ứng dụng luôn hoạt động mượt mà và hiệu quả. Điều này giúp giảm thiểu gánh nặng quản lý hạ tầng cho các nhà phát triển, cho phép họ tập trung vào việc xây dựng và cải thiện ứng dụng mà không lo về các vấn đề kỹ thuật phức tạp [111] [112].

Khả năng mở rộng linh hoạt là một trong những yếu tố quan trọng giúp GAE trở thành một nền tảng đáng tin cậy cho các ứng dụng quy mô lớn. GAE cung cấp hai chế độ mở rộng tài nguyên: tự động và thủ công. Trong chế độ tự động, GAE sẽ tự động mở rộng hoặc thu nhỏ tài nguyên dựa trên lưu lượng thực tế, đảm bảo ứng dụng hoạt động hiệu quả trong mọi tình huống. Ngược lại, chế độ thủ công cho phép nhà phát triển kiểm soát số lượng tài nguyên cụ thể, phù hợp với nhu cầu của ứng dụng. Điều này đảm bảo rằng các ứng dụng trên GAE luôn có khả năng chịu tải tốt và độ tin cậy cao, đồng thời giảm thiểu chi phí vận hành [111] [112].

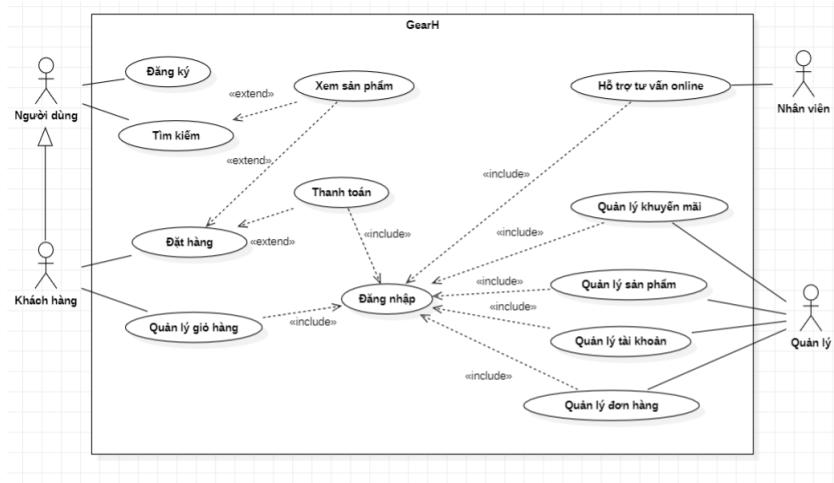
Bảo mật là yếu tố quan trọng không thể thiếu trong các nền tảng đám mây, và GAE cũng không phải ngoại lệ. Google cung cấp các tính năng bảo mật mạnh mẽ như mã hóa dữ liệu, bảo vệ chống lại tấn công DDoS và quản lý truy cập thông qua Google Identity and Access Management (IAM). Hệ thống IAM cho phép người dùng quản lý quyền truy cập vào các tài nguyên một cách chi tiết, đảm bảo rằng chỉ những người có quyền hạn mới có thể truy cập và sửa đổi dữ liệu quan trọng [111].

### 3.4. Phân tích và thiết kế hệ thống

Hệ thống Website bán Gaming Gear GearH là nền tảng chuyên cung cấp các sản phẩm phục vụ nhu cầu chơi game, từ bàn phím, chuột, tai nghe, màn hình đến ghế gaming và các phụ kiện khác. Các chức năng chính bao gồm trang chủ nơi hiện thị danh mục sản phẩm, công cụ tìm kiếm giúp tìm kiếm sản phẩm nhanh chóng.

Trang web cung cấp cho khách hàng xem chi tiết sản phẩm và đặt hàng sản phẩm yêu thích. Bên cạnh đó các chức năng của quản lý bao gồm quản lý khuyến mãi, quản lý sản phẩm, quản lý tài khoản, quản lý đơn hàng.

#### 3.4.1. Sơ đồ Use case



Hình 3.3: Sơ đồ Use case tổng quát

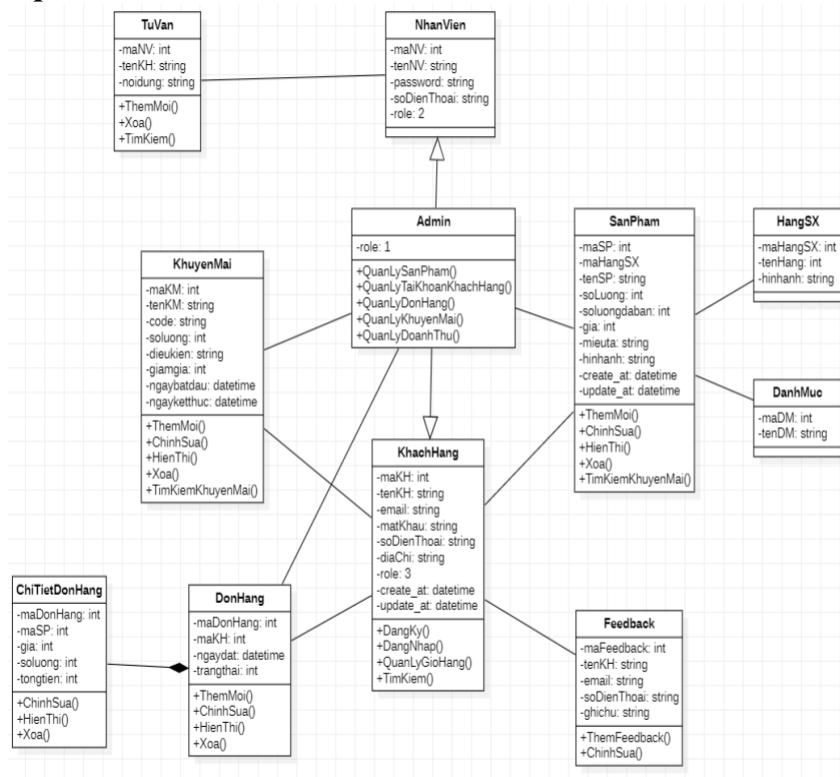
Theo hình 3.3 sơ đồ Use case tổng quát của hệ thống Website bán Gaming Gear mô tả 3 loại tác nhân chính bao gồm người dùng, khách hàng, nhân viên và quản lý, với các vai trò và chức năng cụ thể trong hệ thống. Người dùng, chưa đăng nhập hoặc chưa đăng ký, có thể thực hiện các thao tác cơ bản như tìm kiếm và xem sản phẩm. Khi đăng ký và đăng nhập, họ trở thành khách hàng và có thể truy cập thêm các chức năng như quản lý giờ hàng, đặt hàng và thanh toán. Việc đăng nhập là bước quan trọng, đóng vai trò nền tảng để khách hàng sử dụng các tính năng cá nhân hóa.

Quản lý đóng vai trò điều hành và hỗ trợ hệ thống, thực hiện các chức năng quản lý quan trọng như cập nhật thông tin sản phẩm, xử lý đơn hàng, tổ chức các chương trình khuyến mãi và quản lý tài khoản khách hàng. Đồng thời, nhân viên cũng hỗ trợ khách hàng thông qua tính năng tư vấn trực tuyến, tạo cầu nối trực tiếp để giải đáp thắc mắc hoặc hỗ trợ trong quá trình mua hàng.

Bảng 3.1: Ý nghĩa Use case

STT	Use case	Ý nghĩa	Actor
1	Đăng ký	Đăng ký tài khoản	Khách hàng
2	Đăng nhập	Đăng nhập vào hệ thống	Khách hàng, Admin
3	Xem sản phẩm	Xem chi tiết thông tin sản phẩm	Khách hàng
4	Tìm kiếm	Tìm kiếm sản phẩm theo tên, danh mục	Khách hàng
5	Đặt hàng	Đặt hàng trong giỏ hàng	Khách hàng
6	Thanh toán	Có thể chọn phương thức thanh toán	Khách hàng
7	Quản lý giỏ hàng	Thêm, xóa, sửa giỏ hàng trước khi thanh toán	Khách hàng
8	Quản lý sản phẩm	Thêm, xóa, sửa sản phẩm	Admin
9	Quản lý tài khoản	Thêm, xóa, sửa tài khoản	Admin
11	Quản lý đơn hàng	Quản lý đơn hàng của khách hàng đặt	Admin
12	Quản lý khuyến mãi	Thêm, xóa, sửa khuyến mãi	Admin
13	Hỗ trợ online	Hỗ trợ và hướng dẫn cho khách hàng	Nhân viên

### 3.4.2. Sơ đồ lớp



Hình 3.4: Sơ đồ lớp

Sơ đồ lớp của hệ thống bán hàng Gaming Gear thể hiện cấu trúc dữ liệu và mối quan hệ giữa các thành phần trong hệ thống. Trong hệ thống này, các lớp chính bao gồm Khách hàng, Admin, Nhân viên, Sản phẩm, Đơn hàng, Khuyến mãi, và một số lớp hỗ trợ như Tư vấn, Danh mục, Hàng sản xuất, Chi tiết đơn hàng, và Nhận xét.

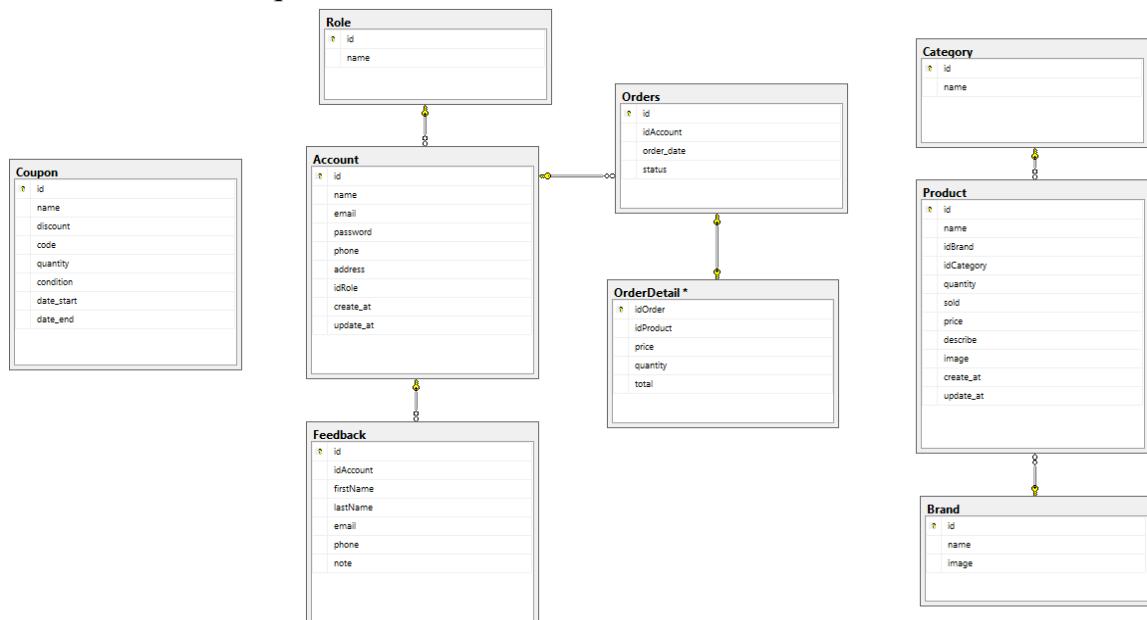
Lớp Admin kế thừa từ lớp Nhân viên, chịu trách nhiệm quản lý toàn bộ hệ thống. Admin có thể quản lý sản phẩm, danh mục, hàng sản xuất, khuyến mãi, đơn hàng, cũng như thống kê doanh thu. Mọi quan hệ giữa Admin và các lớp khác là mối quan hệ trực tiếp, thể hiện quyền điều hành toàn diện của Admin. Lớp Nhân viên có nhiệm vụ hỗ trợ tư vấn, xử lý thông tin khách hàng và các yêu cầu từ hệ thống. Lớp Tư vấn là một phần trong nhiệm vụ của nhân viên, cho phép quản lý các yêu cầu và phản hồi từ khách hàng.

Lớp Sản phẩm liên kết chặt chẽ với Danh mục và Hàng sản xuất, giúp phân loại và xác định nguồn gốc sản phẩm. Một sản phẩm thuộc một danh mục và một hàng sản xuất cụ thể, trong khi mỗi danh mục hoặc hàng sản xuất có thể chứa nhiều sản phẩm. Lớp Feedback lưu trữ phản hồi từ khách hàng về sản phẩm hoặc dịch vụ, cho phép hệ thống cải thiện chất lượng.

Lớp Khuyến mãi lưu trữ thông tin về các chương trình giảm giá, có thể áp dụng cho khách hàng, sản phẩm hoặc cả đơn hàng. Khuyến mãi được quản lý bởi Admin và liên kết với khách hàng thông qua các giao dịch cụ thể. Khách hàng có thể sử dụng các mã khuyến mãi để giảm giá đơn hàng.

### 3.4.3. Thiết kế cơ sở dữ liệu

#### 3.4.3.1. Lược đồ quan hệ thực thể



Hình 3.5: Lược đồ quan hệ thực thể

Bảng 3.2: Mô tả bảng cơ sở dữ liệu

STT	Tên	Mô tả
1	Role	Vai trò khi sử dụng hệ thống (admin, nhân viên, khách hàng)
2	Account	Thông tin tài khoản của người dùng
3	Brand	Danh sách hãng sản xuất
4	Category	Danh sách danh mục sản phẩm
5	Product	Danh sách thông tin sản phẩm
6	Orders	Thông tin tổng quan khi mua hàng
7	OrderDetail	Thông tin chi tiết của đơn hàng
8	Coupon	Danh sách mã khuyến mãi
9	Feedback	Danh sách phản hồi của khách hàng

### 3.4.3.2. Lược đồ chi tiết

Bảng 3.3: Bảng ROLE

Tên cột	Kiểu dữ liệu	Mô tả
idRole	INT	Khóa chính, tự động tăng
name	NVARCHAR (30)	admin   nhân viên   khách hàng

Theo bảng 3.3, Bảng Role đóng vai trò quản lý thông tin về vai trò của người dùng trong hệ thống. Bảng này có hai cột chính: idRole và name. Cột idRole có kiểu dữ liệu là số nguyên và được thiết lập là khóa chính với thuộc tính tự động tăng, điều này đảm bảo mỗi vai trò sẽ có một mã định danh duy nhất. Cột name có kiểu dữ liệu chuỗi, dùng để lưu trữ tên vai trò, chẳng hạn như "admin", "nhân viên", hoặc "khách hàng".

Hệ thống sử dụng bảng này để phân quyền và xác định chức năng mà từng người dùng có thể thực hiện. Bảng Role hoạt động như một cấu trúc cốt lõi để thiết lập và duy trì hệ thống phân quyền một cách rõ ràng và dễ mở rộng.

Bảng 3.4: Bảng ACCOUNT

Tên cột	Kiểu dữ liệu	Mô tả
idAccount	INT	Khóa chính, mã tài khoản, tự động tăng
name	NVARCHAR (30)	Tên tài khoản
email	NVARCHAR (100)	Email đăng nhập

password	NVARCHAR (255)	Mật khẩu
phone	NVARCHAR (20)	Số điện thoại
address	NVARCHAR(MAX)	Địa chỉ
idRole	INT	Mã vai trò của tài khoản
create_at	DATETIME	Thời gian tạo tài khoản
update_at	DATETIME	Thời gian cập nhật tài khoản

Theo bảng 3.4, bảng Account chịu trách nhiệm lưu trữ thông tin chi tiết về tài khoản của người dùng trong hệ thống. Bảng này chứa các cột quan trọng giúp quản lý thông tin cá nhân, thông tin đăng nhập, và vai trò của mỗi tài khoản. Cột idAccount là khóa chính, có kiểu dữ liệu số nguyên và được thiết lập tự động tăng, đảm bảo mỗi tài khoản có một mã định danh duy nhất. Với các cột name, email, password, phone, address với kiểu dữ liệu chuỗi chứa thông tin của người dùng.

Cột idRole là khóa ngoại liên kết với bảng Role, xác định vai trò của tài khoản. Điều này cho phép hệ thống phân quyền và xác định các chức năng cụ thể mà tài khoản được phép thực hiện. Cột create\_at, có kiểu dữ liệu thời gian, ghi nhận thời điểm tài khoản được tạo, trong khi cột update\_at lưu lại thời gian cập nhật cuối cùng thông tin của tài khoản.

Bảng 3.5: Bảng BRAND

Tên cột	Kiểu dữ liệu	Mô tả
idBrand	INT	Khóa chính, mã hàng SX, tự động tăng
name	NVARCHAR (255)	Tên hàng
idCategory	INT	Mã danh mục

Theo bảng 3.5, bảng Brand chịu trách nhiệm quản lý thông tin về các hãng sản xuất trong hệ thống. Bảng này đóng vai trò quan trọng trong việc tổ chức và phân loại các sản phẩm theo từng thương hiệu. Cột idBrand là khóa chính của bảng, có kiểu dữ liệu số nguyên và được thiết lập tự động tăng. Đảm bảo mỗi hãng sản xuất được gán một mã định danh duy nhất, giúp dễ dàng truy xuất và liên kết dữ liệu với các bảng khác.

Cột name, với kiểu dữ liệu chuỗi, lưu trữ tên của hãng sản xuất. Cột idCategory là khóa ngoại liên kết với bảng Category, xác định danh mục mà hãng sản xuất thuộc về. Thông qua mối liên hệ này, hệ thống có thể tổ chức các thương hiệu theo từng nhóm sản phẩm cụ thể, chẳng hạn như thiết bị ngoại vi, linh kiện máy tính hoặc ghé gaming.

Bảng 3.6: Bảng CATEGORY

Tên cột	Kiểu dữ liệu	Mô tả
idCategory	INT	Khóa chính, mã danh mục tự động tăng
name	NVARCHAR (100)	Tên danh mục, Mặc định: 'Chưa đặt tên'

Theo bảng 3.6, bảng Category được sử dụng để quản lý thông tin về các danh mục sản phẩm trong hệ thống, đóng vai trò quan trọng trong việc phân loại sản phẩm và tối ưu hóa việc tìm kiếm hoặc sắp xếp. Cột idCategory là khóa chính của bảng, có kiểu dữ liệu số nguyên và được thiết lập tự động tăng. Đảm bảo rằng mỗi danh mục đều có một mã định danh duy nhất, giúp liên kết với các bảng khác, chẳng hạn như bảng Product hoặc Brand.

Cột name có kiểu dữ liệu chuỗi, lưu trữ tên của danh mục sản phẩm. Giá trị mặc định của cột này là "Chưa đặt tên", được sử dụng khi danh mục chưa được đặt tên cụ thể. Giúp duy trì tính nhất quán trong dữ liệu và tránh trường hợp cột bị đẻ trống.

Bảng 3.7: Bảng PRODUCT

Tên cột	Kiểu dữ liệu	Mô tả
idProduct	INT	Khóa chính, mã sản phẩm, tự động tăng
name	NVARCHAR (30)	Tên sản phẩm
idBrand	INT	Mã hãng SX
idCategory	INT	Mã danh mục
quantity	INT	Số lượng sản phẩm còn hàng
sold	INT	Số lượng sản phẩm đã bán
price	INT	Giá sản phẩm
describe	NVARCHAR(MAX)	Mô tả sản phẩm
image	NVARCHAR (255)	Hình ảnh của sản phẩm
create_at	DATETIME	Thời gian tạo sản phẩm
update_at	DATETIME	Thời gian cập nhật sản phẩm

Theo bảng 3.7, bảng Product đóng vai trò quan trọng trong hệ thống, chịu trách nhiệm lưu trữ thông tin chi tiết về các sản phẩm được cung cấp. Cột idProduct là khóa chính của bảng, có kiểu dữ liệu số nguyên và được thiết lập tự động tăng. Mã sản phẩm

này là duy nhất, giúp xác định và truy xuất chính xác từng sản phẩm trong cơ sở dữ liệu. Cột name, với kiểu dữ liệu chuỗi, lưu trữ tên sản phẩm.

Cột quantity với kiểu số nguyên lưu trữ số lượng sản phẩm còn hàng, giúp theo dõi tình trạng tồn kho. Cột sold ghi nhận số lượng sản phẩm đã bán, hỗ trợ việc thống kê và phân tích kinh doanh. Cột price, lưu giá bán của sản phẩm, giúp hệ thống dễ dàng tính toán tổng giá trị đơn hàng và hiển thị giá cho khách hàng. Cột describe, chứa thông tin mô tả chi tiết về sản phẩm, như tính năng, thông số kỹ thuật hoặc đặc điểm nổi bật. Cột image, lưu đường dẫn hình ảnh của sản phẩm, giúp hiển thị sản phẩm một cách trực quan trên giao diện người dùng.

Bảng 3.8: Bảng ORDER

Tên cột	Kiểu dữ liệu	Mô tả
idOrder	INT	Khóa chính, mã hóa đơn, tự động tăng
idAccount	INT	Mã người dùng
order_date	DATETIME	Thời gian đặt hàng
status	INT	1: Chờ xác nhận   2: Xác nhận

Theo bảng 3.8, bảng Order chịu trách nhiệm lưu trữ thông tin chi tiết về các đơn đặt hàng trong hệ thống. Đây là bảng quan trọng để quản lý quy trình giao dịch từ phía khách hàng và theo dõi trạng thái của từng đơn hàng. Cột idOrder là khóa chính của bảng, có kiểu dữ liệu số nguyên và được thiết lập tự động tăng. Đảm bảo mỗi đơn hàng có một mã định danh duy nhất, giúp hệ thống dễ dàng quản lý và truy xuất thông tin từng đơn hàng.

Cột idAccount là khóa ngoại, liên kết với bảng Account, dùng để xác định tài khoản của người dùng đã thực hiện đặt hàng. Cột order\_date, lưu thời gian đặt hàng, ghi nhận thời điểm chính xác mà đơn hàng được tạo. Cột status, có kiểu dữ liệu số nguyên, được sử dụng để theo dõi trạng thái của đơn hàng. Hệ thống định nghĩa hai trạng thái chính: "Chờ xác nhận", khi đơn hàng mới được đặt và đang trong quá trình xử lý; "Xác nhận", khi đơn hàng đã được phê duyệt và sẵn sàng để tiếp tục các bước tiếp theo.

Bảng 3.9: Bảng ORDERDETAIL

Tên cột	Kiểu dữ liệu	Mô tả
idOrder	INT	Khóa chính, mã hóa đơn
idProduct	INT	Khóa chính, mã sản phẩm

quantity	INT	Số lượng
total	INT	Giá

Theo bảng 3.9, bảng OrderDetails lưu trữ thông tin chi tiết về các sản phẩm trong mỗi đơn hàng. Đây là bảng quan trọng để ghi nhận các sản phẩm cụ thể được đặt trong từng đơn hàng, giúp theo dõi và quản lý từng sản phẩm một cách chi tiết. Cột idOrder và idProduct cùng là khóa chính, dùng để xác định trong hóa đơn có những sản phẩm nào. Khi truy xuất dữ liệu hóa đơn, có thể xem được thông tin hóa đơn.

Cột quantity có kiểu dữ liệu số nguyên, lưu trữ số lượng của sản phẩm được đặt trong đơn hàng. Điều này giúp theo dõi số lượng từng sản phẩm mà khách hàng đã mua. Cột total có kiểu dữ liệu số nguyên, lưu trữ giá trị tổng cho sản phẩm trong đơn hàng, tính theo số lượng và giá của sản phẩm. Tổng giá này có thể dùng để tính toán tổng giá trị đơn hàng, hoặc hỗ trợ các báo cáo tài chính và thống kê doanh thu.

Bảng 3.10: Bảng COUPON

Tên cột	Kiểu dữ liệu	Mô tả
idCoupon	INT	Khóa chính, tự động tăng
name	NVARCHAR (100)	Tên khuyến mãi
discount	INT	Giảm giá
code	NVARCHAR (100)	Mã khuyến mãi
quantity	INT	Số lượng mã khuyến mãi
condition	NVARCHAR(MAX)	Điều kiện sử dụng mã giảm giá
date_start	DATETIME	Thời gian bắt đầu
date_end	DATETIME	Thời gian kết thúc

Theo bảng 3.10, bảng Coupon lưu trữ thông tin về các mã khuyến mãi trong hệ thống. Bảng này giúp quản lý và áp dụng các chương trình giảm giá cho khách hàng trong các giao dịch mua bán. Cột idCoupon là khóa chính của bảng được thiết lập tự động tăng, đảm bảo mỗi mã khuyến mãi sẽ có một mã định danh duy nhất. Cột name, lưu trữ tên của mã khuyến mãi. Cột discount, lưu trữ mức giảm giá mà mã khuyến mãi áp dụng. Giá trị này là tỷ lệ phần trăm, giúp hệ thống tính toán giảm giá cho hóa đơn.

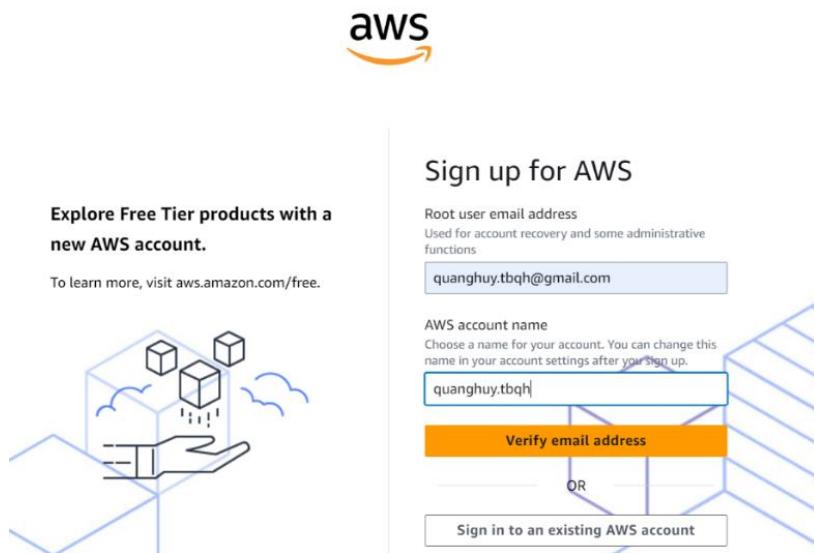
Cột code chứa mã khuyến mãi mà khách hàng sẽ nhập vào khi thanh toán. Cột quantity, lưu trữ số lượng mã khuyến mãi còn lại có thể được sử dụng. Điều này giúp theo dõi số lượng mã khuyến mãi còn hiệu lực, tránh việc mã đã hết lượt sử dụng. Cột

condition, mô tả các điều kiện sử dụng mã giảm giá. Điều này giúp khách hàng hiểu rõ yêu cầu và điều kiện để nhận giảm giá. Cột date\_start và date\_end, lưu trữ thời gian bắt đầu và kết thúc của mã khuyến mãi.

### 3.5. Triển khai và thực nghiệm

Website bán Gaming Gear được triển khai trên nền tảng AWS, sử dụng hai dịch vụ chính của AWS bao gồm AWS RDS để triển khai cơ sở dữ liệu quan hệ SQL và Elastic Beanstalk để triển khai website. Bên cạnh đó sử dụng một số dịch vụ hỗ trợ như IAM để quản lý người dùng truy cập AWS để xuất bản website trên Visual Studio và EC2 để triển khai máy ảo dùng để tải thư mục website lên IIS.

Bước đầu tiên để sử dụng và triển khai website trên nền tảng AWS là đăng ký tài khoản. Ở đây sẽ đăng ký tài khoản miễn phí trên AWS với loại tài khoản này có thể sử dụng miễn phí trong vòng 12 tháng và hơn 100 dịch vụ khác nhau trên AWS.



Hình 3.6: Trang đăng ký AWS:

Sau khi điền các thông tin cơ bản như email, và tên tài khoản sẽ có email xác thực gửi vào địa chỉ email đã điền. Để thực hiện tạo tài khoản với phiên bản trải nghiệm cần phải thiết lập tài khoản thanh toán với các loại thẻ thanh toán quốc tế.

The screenshot shows the 'Sign up for AWS' process. On the left, under 'Secure verification', there is a note about temporary hold of funds and a shield icon with a checkmark. On the right, under 'Billing Information', fields include 'Billing country' (set to Vietnam), 'Credit or Debit card number' (redacted), 'Expiration date' (September 2029), 'Security code' (redacted), 'Cardholder's name' (TRAN BUI QUANG HUY), and 'Billing address' (checkboxes for 'Use my contact address' and 'Use a new address', both redacted). A 'Verify and Continue (step 3 of 5)' button is at the bottom.

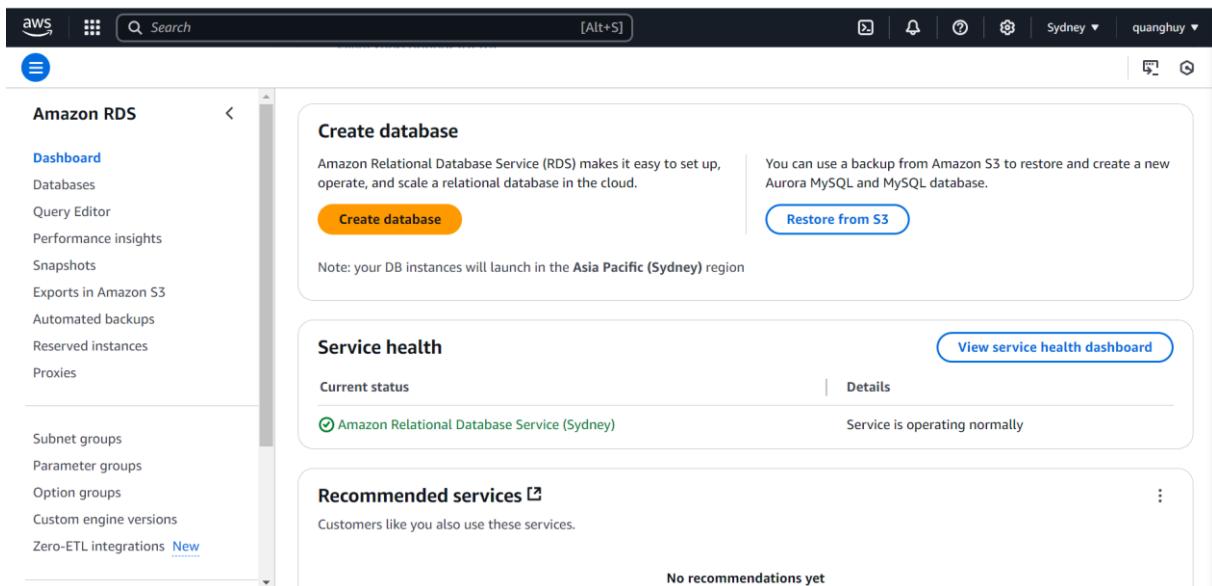
Hình 3.7: Trang tài khoản thanh toán

Sau khi điền thông tin thẻ thanh toán ở hình 3.7, để chọn phiên bản miễn phí cần chọn loại hỗ trợ. Chọn loại hỗ trợ miễn phí (basic support).

The screenshot shows the 'Select a support plan' section. It lists three options: 'Basic support - Free' (selected), 'Developer support - From \$29/month', and 'Business support - From \$100/month'. Each option has a description and an icon. Below the plans is a section for 'Enterprise level support' with a building icon and a note about \$15,000/month. At the bottom is a 'Complete sign up' button.

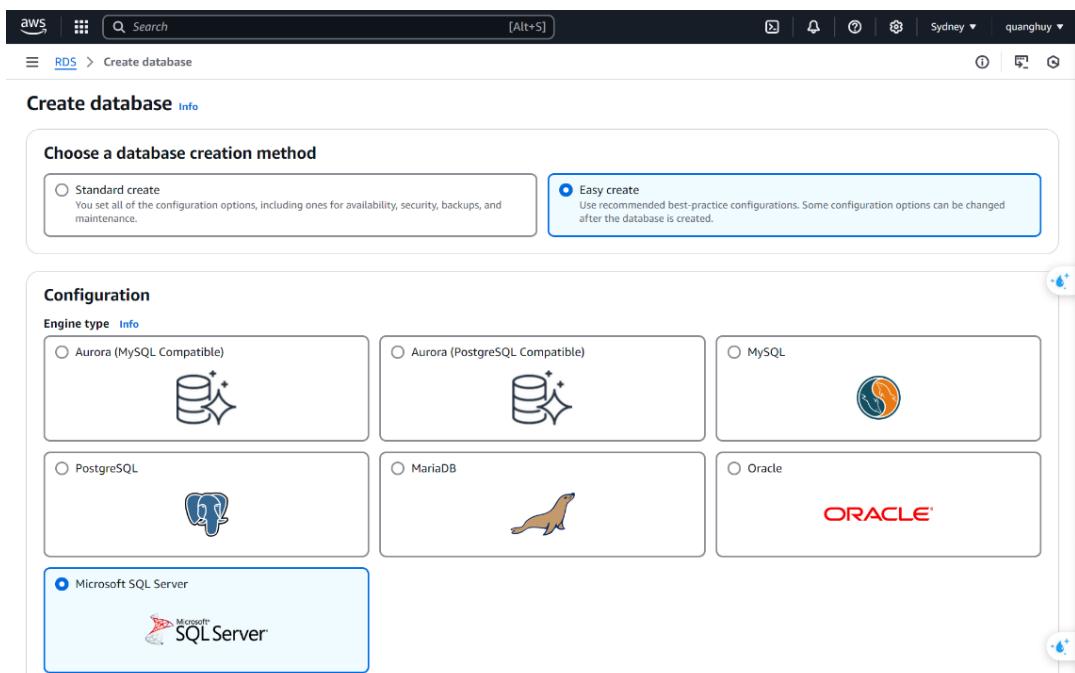
Hình 3.8: Trang chọn hỗ trợ AWS

Bước tạo tài khoản hoàn tất thì bước thứ 2 sẽ thiết lập AWS RDS để triển khai cơ sở dữ liệu quan hệ SQL. Đầu tiên chọn dịch vụ RDS để tiến hành thiết lập.



Hình 3.9: Dịch vụ Amazon RDS

Để tạo mới cơ sở dữ liệu chọn nút ‘Create Database’ để tạo mới.



Hình 3.10: Trang tạo cơ sở dữ liệu

Chọn phương thức tạo database ‘Easy create’, phương thức này giúp triển khai dễ dàng hơn không cần phải cấu hình quá phiền phức. Chọn loại cơ sở dữ liệu ‘Microsoft SQL Server’.

**DB instance size**

<input type="radio"/> Production db.r5.xlarge 4 vCPUs 32 GiB RAM 500 GiB 3.280 USD/hour	<input checked="" type="radio"/> Dev/Test db.m5.large 2 vCPUs 8 GiB RAM 100 GiB 1.068 USD/hour
--	---

**DB instance identifier**  
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.  
**GearHDB**

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

**Master username** [Info](#)  
Type a login ID for the master user of your DB instance.  
**admin**

1 to 16 alphanumeric characters. The first character must be a letter.

**Credentials management**  
You can use AWS Secrets Manager or manage your master user credentials.

<input checked="" type="radio"/> Managed in AWS Secrets Manager - most secure RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.	<input type="radio"/> Self managed Create your own password or have RDS create a password that you manage.
--	---

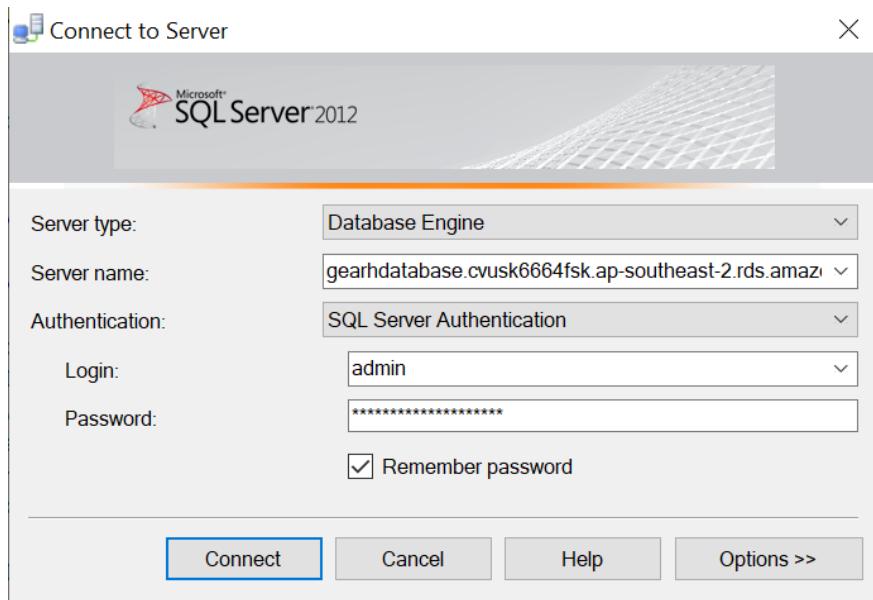
If you manage the master user credentials in AWS Secrets Manager, additional charges apply. See [AWS Secrets Manager pricing](#) Additionally, some RDS features aren't supported. See limitations [here](#).

**Select the encryption key** [Info](#)  
You can encrypt using the KMS key that Secrets Manager creates or a customer managed KMS key that you create.  
**aws/secretsmanager (default)**

[Add new key](#)

*Hình 3.11: Cấu hình cơ sở dữ liệu*

Theo hình 3.11, chọn kích thước cơ sở dữ liệu lưu trữ, vì lý do thử nghiệm nên sẽ chọn ‘Dev/Test’ với 2 vCPUs, 8 GiB RAM và 100 GiB. Tiếp theo đặt tên cho cơ sở dữ liệu và tên tài khoản và mật khẩu để cấu hình trên SQL Management Tool. Sau khi tạo cơ sở dữ liệu thành công sẽ cấu hình bằng tài khoản vừa mới tạo.



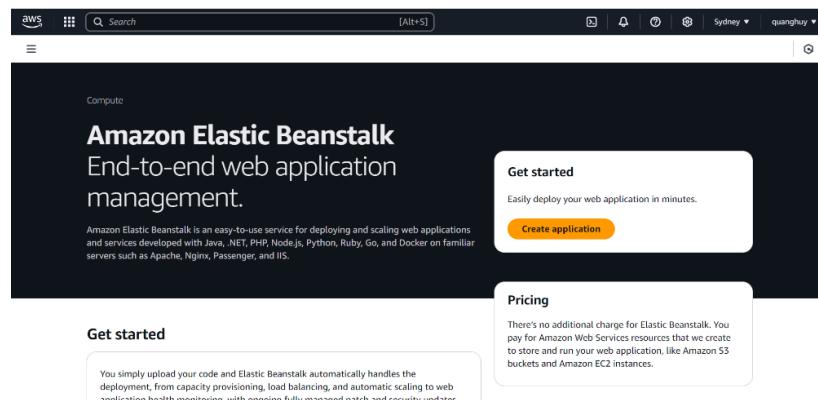
*Hình 3.12: Truy cập RDS trên SQL Server*

Để cấu hình cơ sở dữ liệu trên RDS sẽ thực hiện tạo bảng, truy vấn trên SQL Server Management Tool. Theo hình 3.12, ‘Server name’ là tên của máy chủ sau khi tạo thành công trên RDS và tài khoản, mật khẩu theo như đã đăng ký trước đó. Thiết lập, tạo bảng

trong máy chủ RDS. Kế tiếp sẽ cấu hình lại chuỗi kết nối trong dự án website.

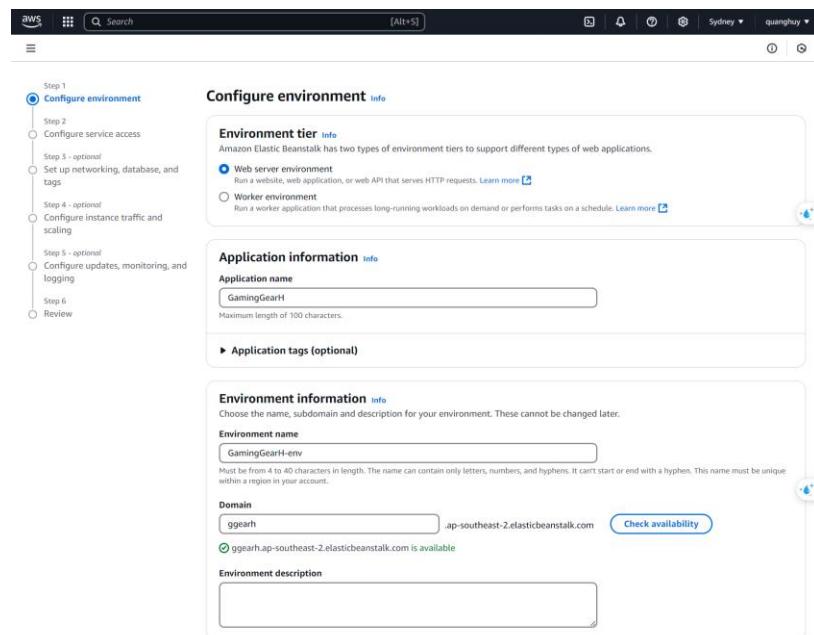
```
<connectionStrings>
    <add name="dbGearHDataContext"
        connectionString="Server=*****.cvusk6664fsk.ap-southeast-2.rds.amazonaws.com,*****;Database=GearH;Integrated Security=false;UserID=*****;Password=*****;MultipleActiveResultSets=True"
        providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Cấu hình chuỗi kết nối hoàn tất, chạy thử website trên localhost để kiểm tra xem RDS đã được kết nối thành công chưa. Sau khi thành công, bước thứ 3 sẽ triển khai website lên Elastic Beanstalk.



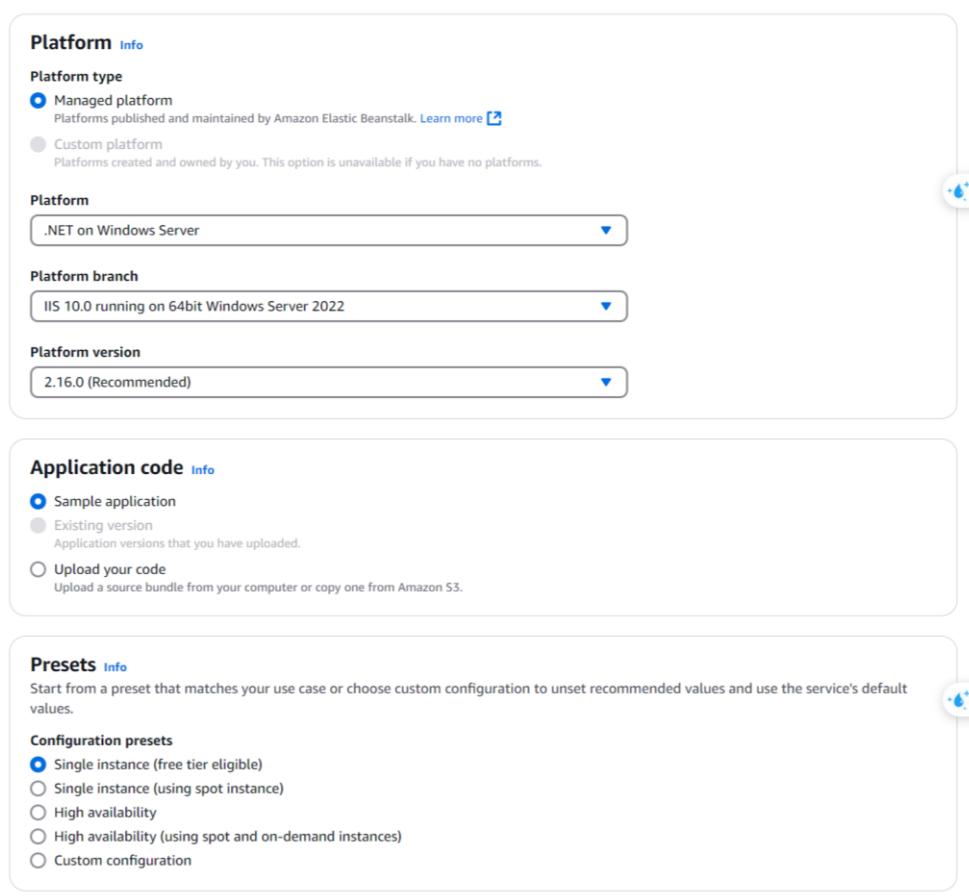
Hình 3.13: Trang Amazon Elastic Beanstalk

Truy cập dịch vụ Amazon Elastic Beanstalk và chọn ‘Create application’ để tạo mới phần mềm và môi trường để triển khai website.



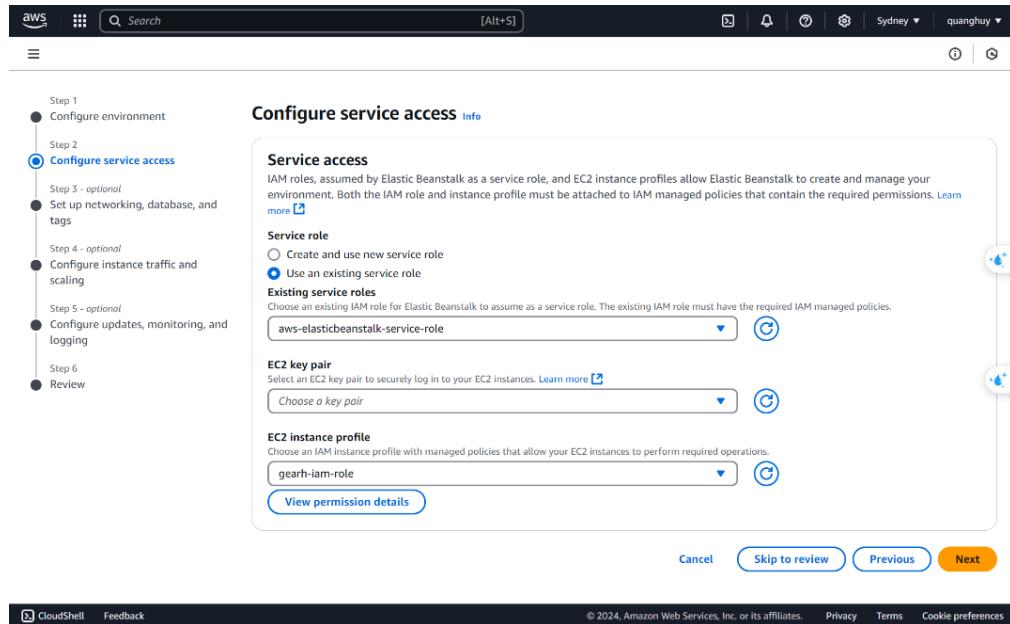
Hình 3.14: Cấu hình môi trường Elastic Beanstalk

Chọn loại môi trường để cấu hình mục đích sử dụng, ở đây chọn ‘Web server environment’ nhằm mục đích triển khai website. Đặt tên ứng dụng, tên mô trường và tên miền hợp lệ theo hình 3.14.



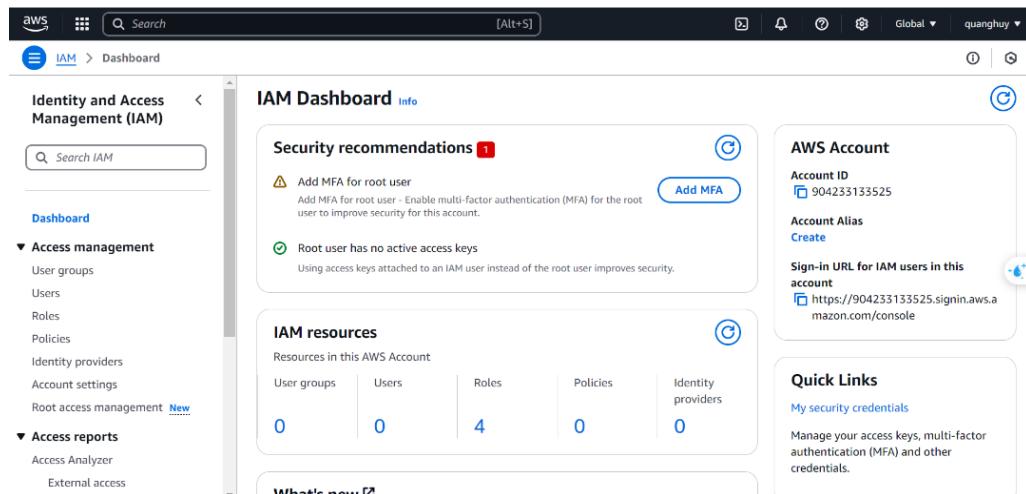
Hình 3.15: Chọn nền tảng triển khai Website

Tiếp tục chọn nền tảng triển khai Website, ở đây sẽ triển khai trên Windows Server 2022 với IIS 10.0 theo hình 3.25. Và chọn tạm thời sample code để kiểm thử Elastic hoạt động tốt rồi mới tải lên thư mục code.



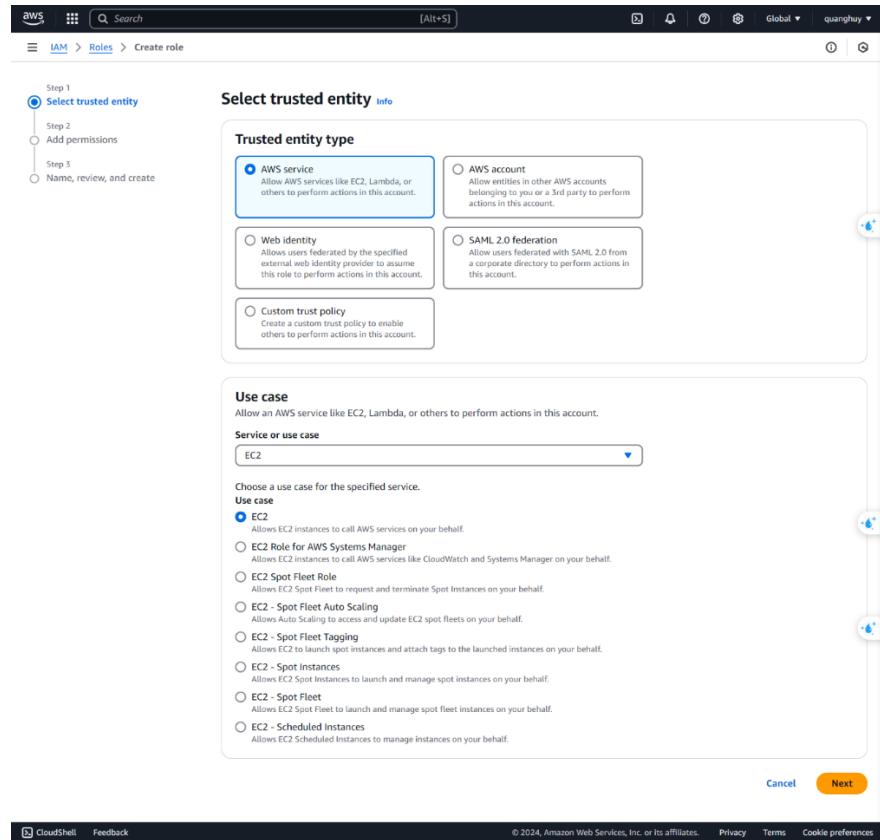
Hình 3.16: Cấu hình quyền truy cập dịch vụ

Để sử dụng được Elastic Beanstalk cần cấu hình sự truy cập của dịch vụ, ở trường ‘EC2 instance profile’ thiết lập các quyền của EC2 bằng AWS IAM.



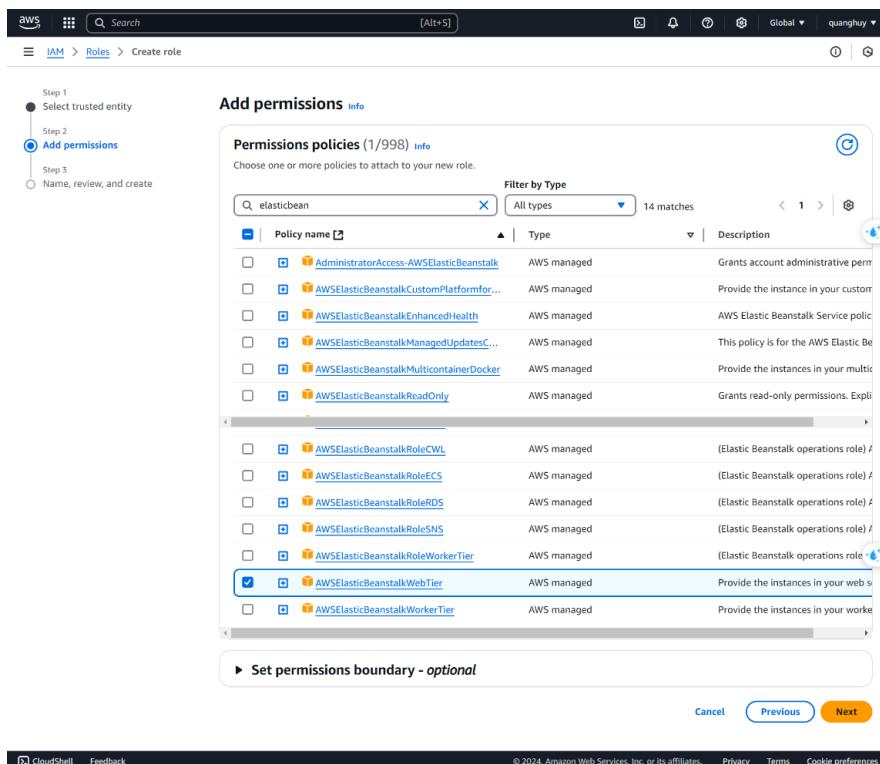
Hình 3.17: Trang Amazon IAM

Để có thể thêm quyền và trường ‘EC2 Instance profile’ cần truy cập AWS IAM để tạo quyền mới và thêm vào trường ‘EC2 instance profile’.



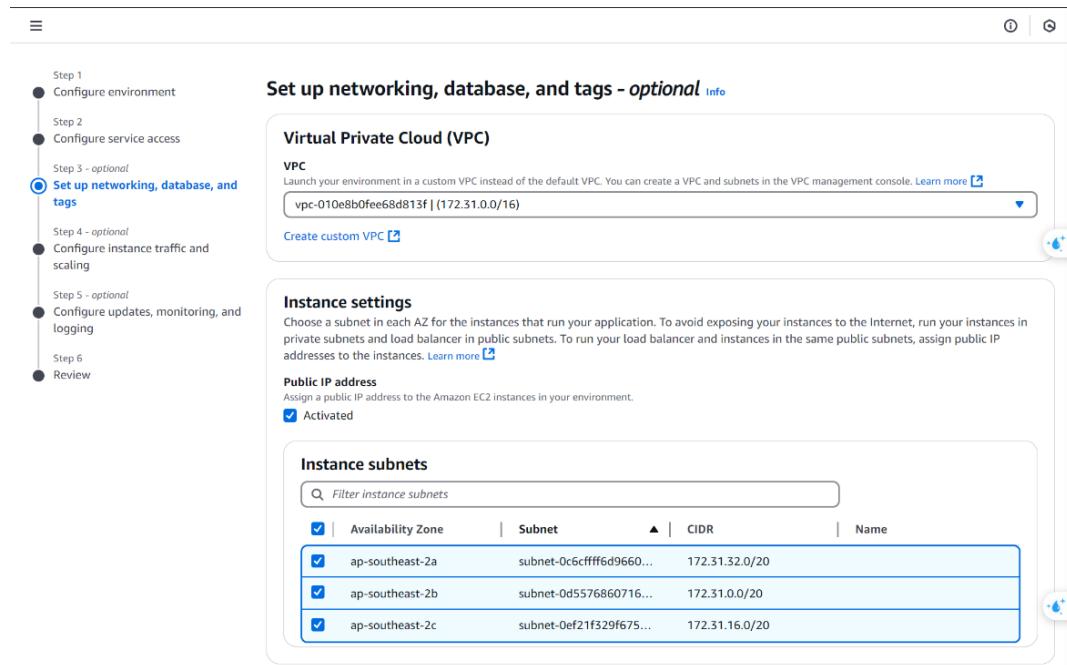
Hình 3.18: Cấu hình vai trò

Chọn các thuộc tính như hình 3.18 để tạo quyền trong EC2.



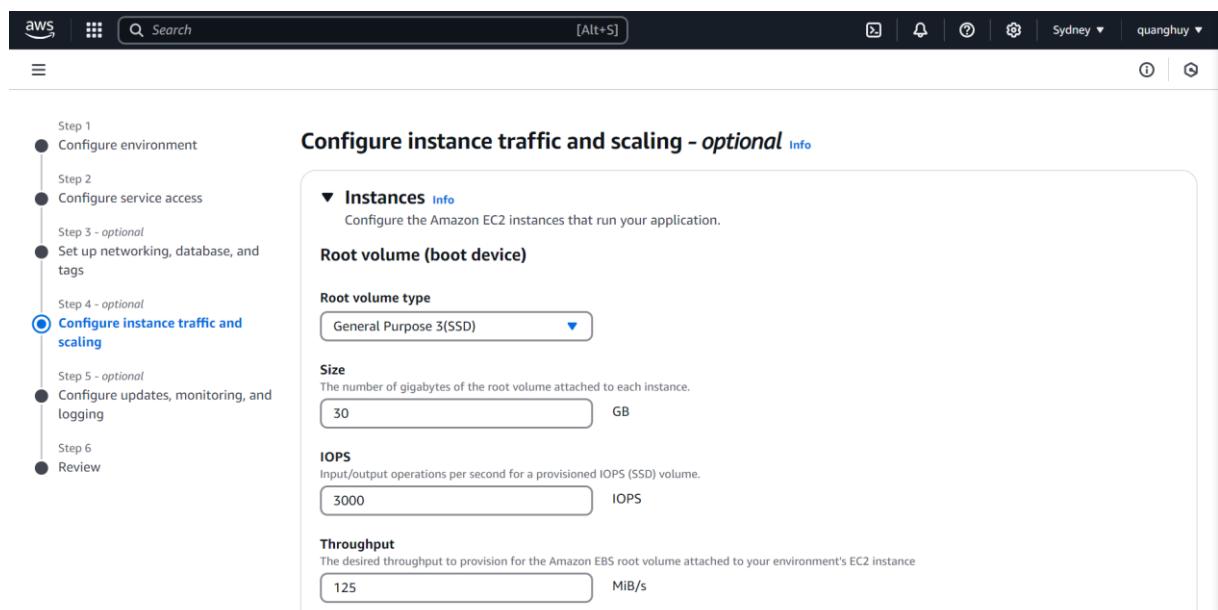
Hình 3.19: Thêm quyền EC2

Thêm các quyền như hình 3.19 để EC2 có thể truy cập được AWS Elastic Beanstalk. Thêm thành công quyền sẽ quay lại bước thiết lập Elastic.



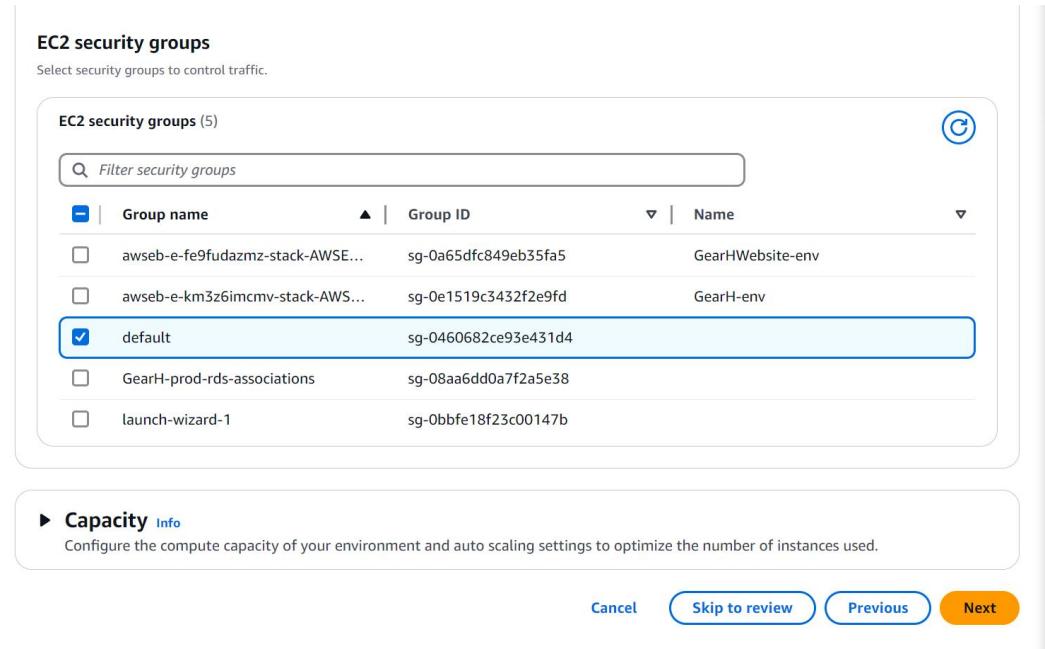
Hình 3.20: Cấu hình máy ảo

Cấu hình đám mây riêng tư ảo (Virtual Private Cloud) và chọn IP công khai.



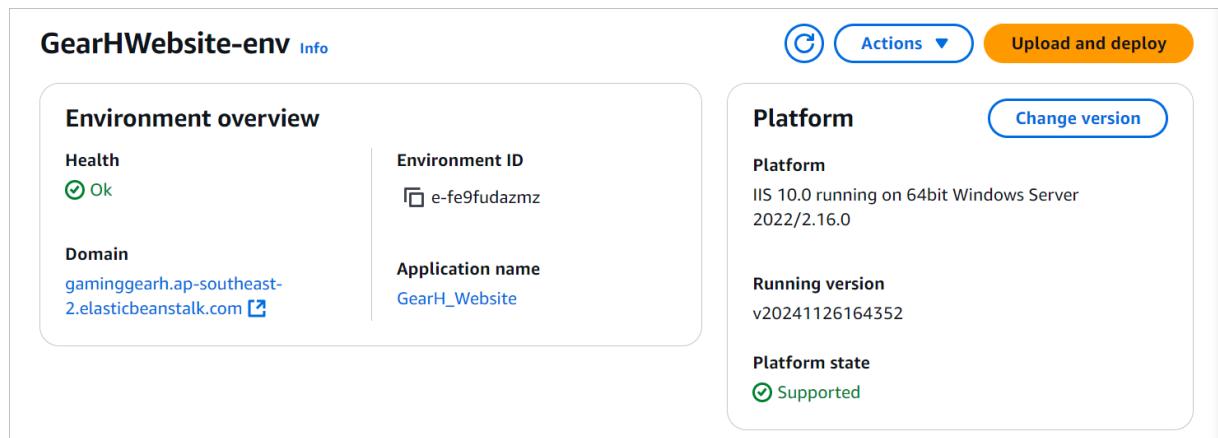
Hình 3.21: Cấu hình Amazon EC2

Cấu hình Amazon EC2 theo hình 3.21 với kích thước 30GB, IOPS 3000 IOPS và thông lượng 125 MiB/s.



Hình 3.22: Cấu hình nhóm bảo mật EC2

Chọn nhóm bảo mật, ở đây chọn nhóm mặc định. Và chọn ‘Skip to review’ và chọn ‘Submit’ để hoàn tất khởi tạo môi trường.



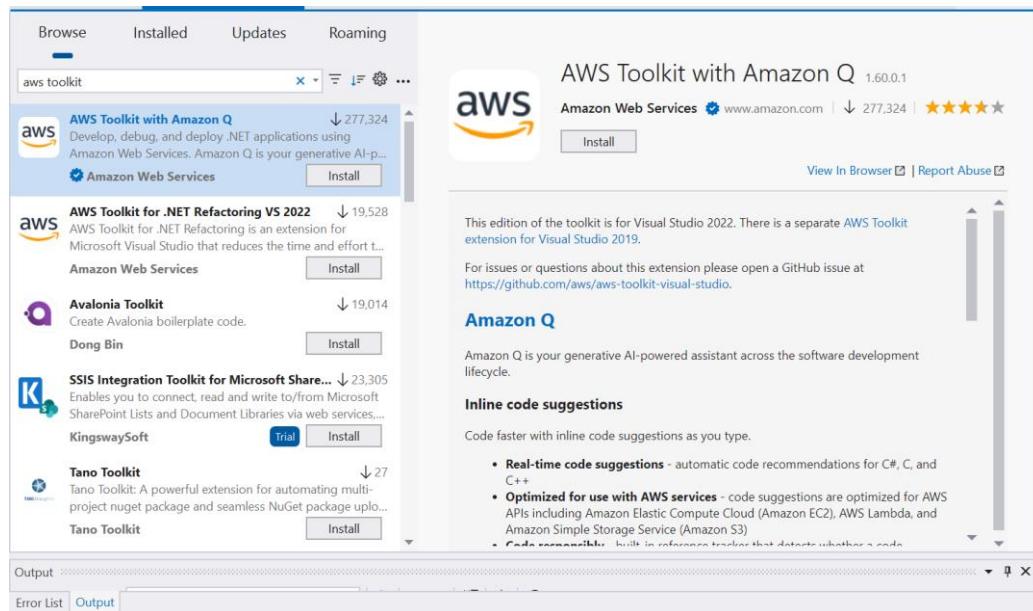
Hình 3.23: Môi trường cài đặt thành công

Sau khi chờ môi trường được thiết lập xong và chuyển sang trạng thái OK có nghĩa là Elastic Beanstalk đã được cấu hình thành công. Mở website với tên miền được cung cấp để xem theo trang code mẫu.



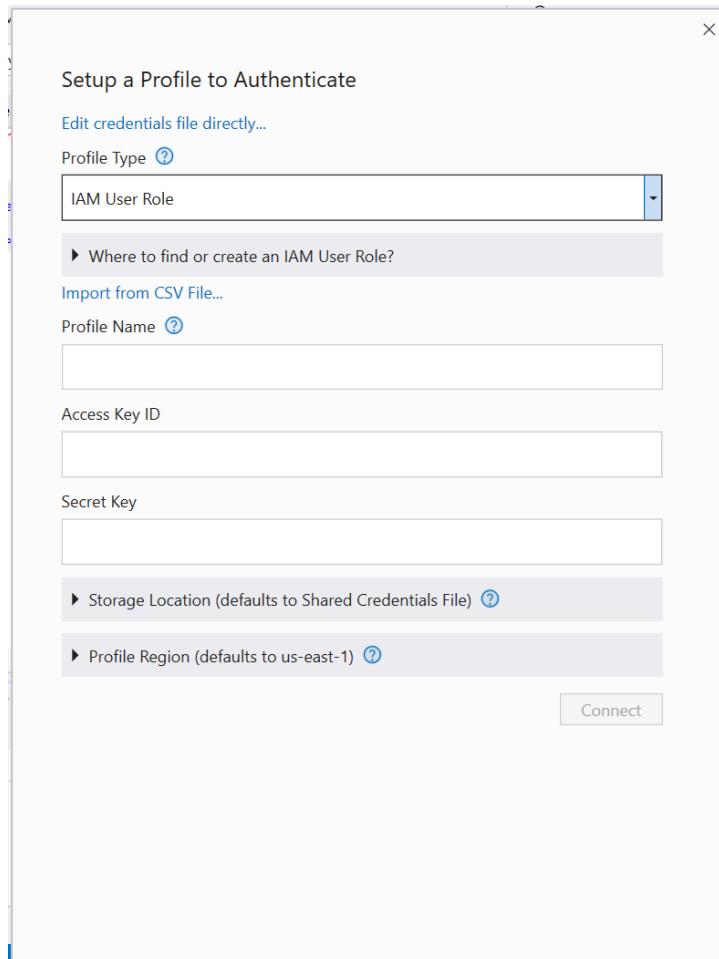
Hình 3.24: Giao diện sau khi thiết lập môi trường

Hiện thị trang web như hình 3.24 chứng tỏ Elastic Beanstalk đã được cấu hình thành công. Bước cuối cùng là sẽ tải code lên để tiến hành triển khai website trên Elastic Beanstalk. Đầu tiên cần phải cài đặt Extension AWS Toolkit trong Visual Studio 2022.



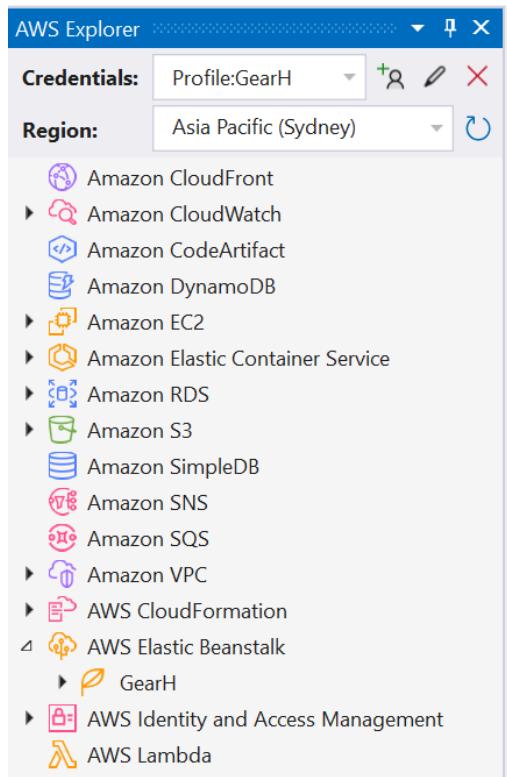
Hình 3.25: Cài đặt tiện ích AWS Toolkit

Sau khi cài đặt AWS Toolkit thành công thêm profile người dùng vào để có thể kết nối với AWS.



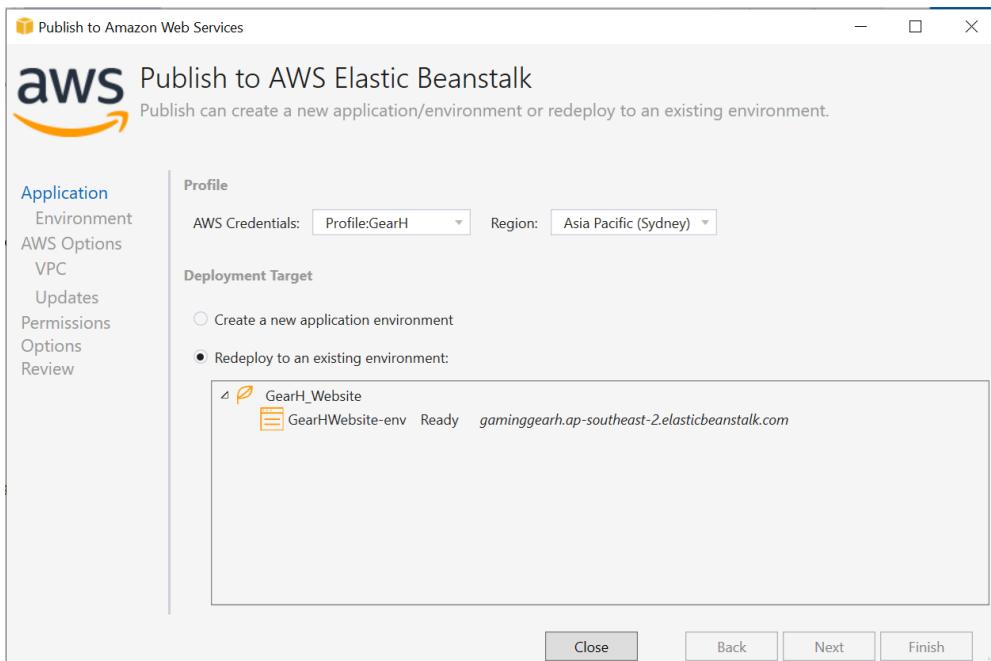
*Hình 3.26: Thêm thông tin người dùng AWS Toolkit*

Để hiện thị thanh công cụ của AWS Toolkit, trên thanh công cụ View chọn AWS Explorer để hiện các dịch vụ của AWS và hiện thị ứng dụng AWS Elastic Beanstalk đã cấu hình trước đó.



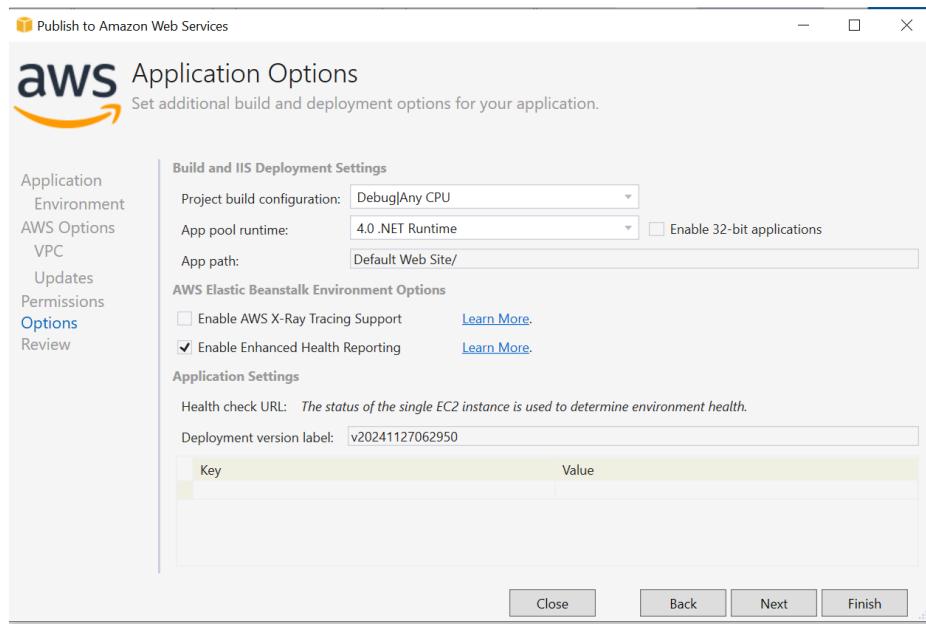
Hình 3.27: Kết nối thành công AWS

Kết nối thành công AWS với thư mục dự án thì sẽ triển khai thư mục code lên môi trường Elastic Beanstalk. Chuột phải vào thư mục tổng và chọn Publish to AWS.



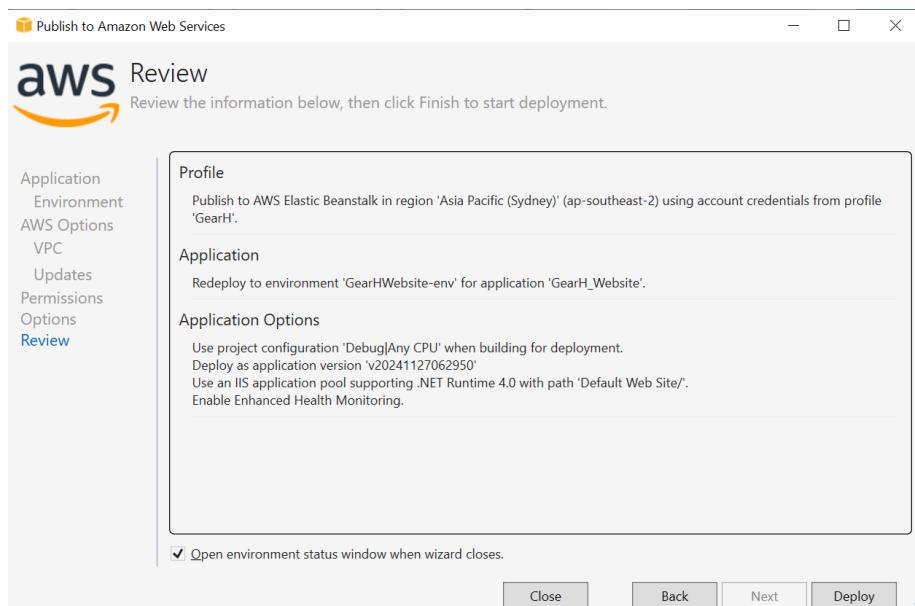
Hình 3.28: Xuất thư mục code lên AWS

Chọn môi trường muốn triển khai và nhấn ‘Next’.



Hình 3.29: Cấu hình đường dẫn lưu trữ thư mục

Thư mục code sẽ được nằm trong Default Web Site/ trên EC2. Nhấn ‘Next’ để chuyển sang bước tiếp theo.



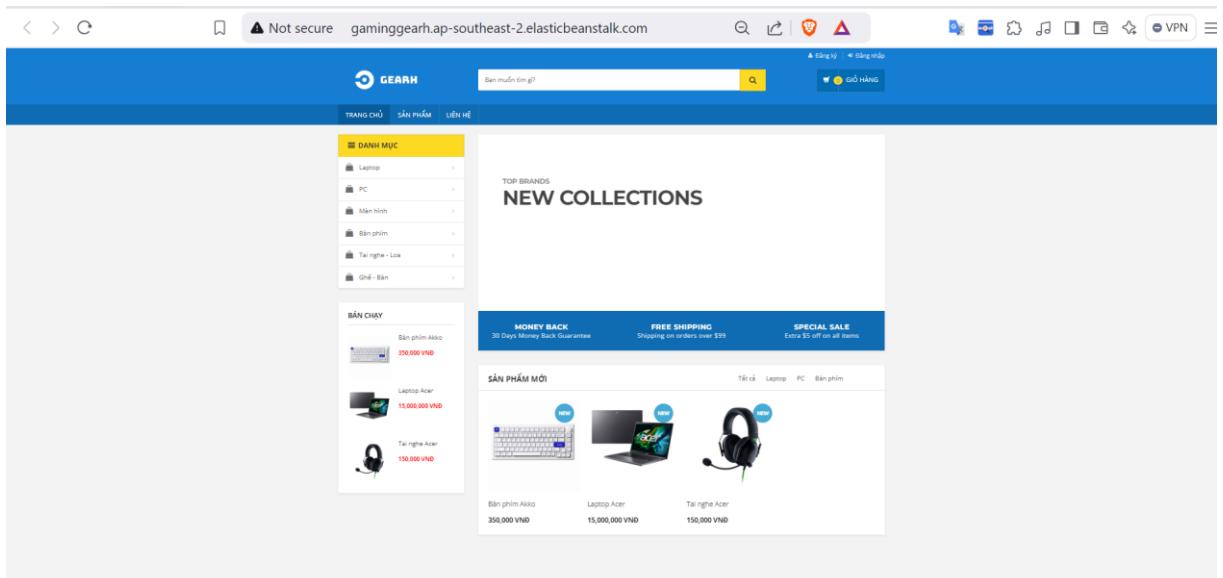
Hình 3.30: Triển khai Website lên AWS Elastic Beanstalk

Nhấn ‘Deploy’ để tiến hành triển khai website GearH.

Event Time	Event Type	Version Label	Environment Name	Event Details
11/27/2024 8:20:42 AM	INFO		GearHWebsite-env	Environment update is starting.
11/27/2024 12:38:34 AM	INFO		GearHWebsite-env	createConfigurationTemplate completed successfully
11/27/2024 12:38:33 AM	INFO		GearHWebsite-env	Created new Configuration template: Version 1.0 and createConfigurationTemplate is starting.
11/27/2024 12:38:32 AM	INFO		GearHWebsite-env	
11/27/2024 12:16:24 AM	INFO	v20241126164352	GearHWebsite-env	Environment health has transitioned from Info to Critical
11/27/2024 12:15:24 AM	INFO	v20241126164352	GearHWebsite-env	Environment health has transitioned from Degraded to Critical
11/27/2024 12:15:00 AM	INFO		GearHWebsite-env	Environment update completed successfully.
11/27/2024 12:15:00 AM	INFO		GearHWebsite-env	Successfully deployed new configuration to environment
11/27/2024 12:15:00 AM	INFO		GearHWebsite-env	New application version was deployed to running instances
11/27/2024 12:14:13 AM	INFO		GearHWebsite-env	Started Application Update
11/27/2024 12:13:23 AM	INFO		GearHWebsite-env	Deploying new version to instance(s).
11/27/2024 12:13:05 AM	INFO		GearHWebsite-env	Updating environment GearHWebsite-env's configuration
11/27/2024 12:12:56 AM	INFO		GearHWebsite-env	Environment update is starting.
11/27/2024 12:12:55 AM	INFO	v20241126164352		createApplicationVersion completed successfully.
11/27/2024 12:12:55 AM	INFO	v20241126164352		Created new Application Version: v20241126164352!
11/27/2024 12:12:55 AM	INFO	v20241126164352		createApplicationVersion is starting.

Hình 3.31: Trang thái thành công

Trong phần trạng thái của môi trường sẽ hiện thị các sự kiện trong quá trình triển khai. Nếu thành công sẽ hiện ‘Successfully’. Truy cập vào tên miền ban đầu để kiểm tra website đã được triển khai thành công hay chưa.



Hình 3.32: Giao diện trang chủ Website GearH

## TÀI LIỆU THAM KHẢO

- [1] J. & R. C. Surbiryala, "Cloud computing: History and overview," *IEEE Cloud Summit*, pp. 1-7, 2019.
- [2] M. Smith, "Security issues in on-demand grid and cluster computing," *Sixth IEEE International Symposium on Cluster Computing and the Grid*, vol. 2, pp. 14-24, 2006.
- [3] Bote-Lorenzo và cộng sự, "Grid characteristics and uses: a grid definition," *In European Across Grids Conference*, pp. 291-298, 2003.
- [4] B. Varghese, "History of the Cloud," *ITNOW*, vol. 61, no. 2, pp. 46-48, 2019.
- [5] Salesforce, "Salesforce CRM," [Online]. Available: <https://www.salesforce.com/platform/>. [Accessed 26 October 2024].
- [6] A. O. B. Swati I. Bairagi, "Cloud Computing: History, Architecture, Security," *International Journal of Advent Research in Computer and Electronics*, pp. 102-104, 2015.
- [7] Nurmi, Daniel và cộng sự, "The eucalyptus open-source cloud-computing system," *9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pp. 124-131, 2009.
- [8] Qian, Ling và cộng sự, "Cloud computing: An overview," *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China*, pp. 626-631, 2009.
- [9] Foster, Ian và cộng sự, "Cloud computing and grid computing 360-degree compared," *2008 grid computing environments workshop*, pp. 1-10, 2008.
- [10] Buyya, Rajkumar và cộng sự, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems* 25, pp. 599-616, 2009.
- [11] P. Mell, "The NIST Definition of Cloud Computing," *NIST Special Publication*, 2011.
- [12] Z. Wan, "Cloud Computing infrastructure for latency sensitive applications," *IEEE 12th International Conference on Communication Technology*, pp. 1399-1402, 2010.
- [13] A. Apostu, "New Classes of Applications in the Cloud. Evaluating Advantages and Disadvantages of Cloud Computing for Telemetry Applications," *Database Systems Journal*, 2014.
- [14] T. Alam, "Cloud Computing and its role in the Information Technology," *IAIC Transactions on Sustainable Digital Innovation (ITSDI)*, pp. 110-111, 2020.
- [15] A. M. A. Raji, "Enhancing Public Cloud Security by Developing a Model For User Authentication and Data Integrity Checking," *7th International Conference on Computer Science and Information Technology (SCCSIT7)*, vol. 7, 2020.

- [16] J. Yang and Z. Chen, "Cloud computing research and security issues," *International Conference on Computational Intelligence and Software Engineering*, pp. 1-3, 2010.
- [17] S. Dowell, "Cloud to cloud interoperability," *6th International Conference on System of Systems Engineering*, pp. 258-263, 2011.
- [18] R. Ranjan, "The cloud interoperability challenge," *IEEE Cloud Computing*, pp. 20-24, 2014.
- [19] N Loutas và cộng sự, "Cloud Computing Interoperability: The State of Play," *Third IEEE International Conference on Coud Computing Technology and Science*, 2011.
- [20] David Elliott và cộng sự, "A cloud-agnostic container orchestrator for improving interoperability," *2018 IEEE 11th international conference on cloud computing (CLOUD)*, pp. 958-961, 2018.
- [21] A. Salih, "Cloud Computing Service Interoperability and Architectural Concepts," Florida Institute of Technology, 2020.
- [22] Parameswaran và cộng sự, "Cloud interoperability and standardization," *SETlabs briefings*, pp. 19-26, 2009.
- [23] AJ Nelson và cộng sự, "A security and usability perspective of cloud file systems," *6th international conference on system of systems engineering*, pp. 161-166, 2011.
- [24] A. E. Youssef, "Exploring cloud computing services and applications," *Journal of Emerging Trends in Computing and Information Sciences*, pp. 838-847, 2012.
- [25] Qi Zhang và cộng sự, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, pp. 7-18, 2010.
- [26] Nivedita Bisht, Sapna Singh, "Analytical study of different network topologies," *International Research Journal of Engineering and Technology*, pp. 88-90, 2015.
- [27] Q. Liu and Q. Liu, "A Study on Topology in Computer Network," *International Conference on Intelligent Computation Technology and Automation*, pp. 45-48, 2014.
- [28] Kovuri, Karthik và cộng sự, "Evaluation and comparison of hypercube interconnection networks performance," *International Journal of Advanced Science and Technology*, pp. 6954-6962, 2020.
- [29] Canwen Xiao và cộng sự, "DBM: A Dimension-Bubble-Based Multicast Routing Algorithm for 2D Mesh Network-on-Chips," *In Advanced Computer Architecture: 13th Conference*, pp. 43-55, 2020.
- [30] S. W. Fuhrmann, "Performance of a packet switch with crossbar architecture," *IEEE transactions on communications*, pp. 486-491, 1993.

- [31] Vasiliadis, D. C. và cộng sự, "Routing and Performance Analysis of Double-Buffered Omega Networks Supporting Multi-class Priority Traffic," *Third International Conference on Systems and Networks Communications*, pp. 56-63, 2008.
- [32] C. A. Navarro, "A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures," *Communications in Computational Physics*, pp. 285-329, 2014.
- [33] Karloff, Howard và cộng sự, "A model of computation for MapReduce," *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pp. 938-948, 2010.
- [34] Michael J. Flynn, Kevin W. Rudd, "Parallel architectures," *ACM computing surveys*, pp. 67-70, 1996.
- [35] Almutairi, Abdulrahman và cộng sự, "A Distributed Access Control Architecture for Cloud Computing," *IEEE software*, pp. 36-44, 2011.
- [36] H. S. Oluwatosin, "Client-server model," *IOSR Journal of Computer Engineering*, pp. 67-71, 2014.
- [37] Ping Luo và cộng sự, "Distributed classification in peer-to-peer networks," *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 968-976, 2007.
- [38] Pao, Hsiao và cộng sự, "An IaaS Based Interactive Video Platform for E-Learning and Remote Services," *Journal of Computers (Taiwan)*, pp. 20-32, 2013.
- [39] Gesvindr, David và cộng sự, "Architecture design evaluation of PaaS cloud applications using generated prototypes: PaaSArch Cloud Prototyper tool," *Journal of Systems and Software*, 2020.
- [40] S.Satyanarayana, "Cloud computing: SAAS," *Computer Sciences and Telecommunications*, pp. 76-79, 2012.
- [41] Lakshay Malhotra và cộng sự, "Virtualization in cloud computing," *J. Inform*, pp. 1-3, 2014.
- [42] N. Jain and S. Choudhary, "Overview of virtualization in cloud computing," *Symposium on Colossal Data Analysis and Networking*, pp. 1-4, 2016.
- [43] E. Wong, "Next-generation broadband access networks and technologies," *Journal of lightwave technology*, pp. 597-608, 2011.
- [44] Cristina-Elena Vintilă và cộng sự, "Security Analysis of LTE Access Network," *The Tenth International Conference on Networks*, pp. 29-34, 2011.
- [45] Combelles, Pierre và cộng sự, "Fixed wireless access: an overview," *Annals of Telecommunications*, pp. 452-463, 1999.

- [46] Ogutu B. Osoro và cộng sự, "A techno-economic framework for satellite networks applied to low earth orbit constellations: Assessing Starlink, OneWeb and Kuiper," *IEEE Access*, pp. 141611-141625, 2021.
- [47] Shaengchart và cộng sự, "Starlink satellite project impact on the Internet provider service in emerging economies," *Research in Globalization*, 2023.
- [48] Y. Miura, "Hard disk drive technology: past, present and future," *Digest of the Asia-Pacific Magnetic Recording Conference*, 2002.
- [49] D. C. Marinescu, "Cloud data storage," *Cloud Computing (Third edition)*, pp. 215-256, 2023.
- [50] L. Hu, "A Design of an SMTP Email Server," *Journal of Electronic Research and Application*, pp. 41-47, 2024.
- [51] Tsirogiannis, Dimitris và cộng sự, "Analyzing the energy efficiency of a database server," *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 231-242, 2010.
- [52] D. Jacobs, "Data management in application servers," *Datenbank-Spektrum*, pp. 5-11, 2004.
- [53] Douglas Kunda và cộng sự, "Web Server Performance of Apache and Nginx: A Systematic," *Computer Engineering and Intelligent Systems*, pp. 43-52, 2017.
- [54] R. L. Haskin, "The Shark continuous-media file server," *IEEE*, pp. 12-15, 1993.
- [55] Liao, Guangdeng và cộng sự, "A new server I/O architecture for high speed networks," *IEEE 17th International Symposium on High Performance Computer Architecture*, pp. 255-265, 2011.
- [56] T. F. Hasan, "CPU scheduling visualization," *Diyala Journal of Engineering Sciences*, pp. 16-29, 2014.
- [57] Giovannetti, Vittorio và cộng sự, "Architectures for a quantum random access memory," *Physical Review A—Atomic, Molecular, and Optical Physics*, 2008.
- [58] Wang, Haiyang và cộng sự, "On the impact of virtualization on dropbox-like cloud file storage/synchronization services," *IEEE 20th International Workshop on Quality of Service*, pp. 1-9, 2012.
- [59] Williams, Hugh và cộng sự, "Compressing integers for fast file access," *The Computer Journal*, pp. 193-201, 1999.
- [60] Kher, Vishal và cộng sự, "Securing distributed storage: challenges, techniques, and systems," *Proceedings of the 2005 ACM workshop on Storage security and survivability*, pp. 9-25, 2005.
- [61] Ford, Daniel và cộng sự, "Availability in globally distributed storage systems," *USENIX Symposium on Operating Systems Design and Implementation*, 2010.
- [62] A. Osadzinski, "The network file system (NFS)," *Computer Standards & Interfaces*, pp. 45-48, 1988.

- [63] Sandberg, Russel và cộng sự, "Design and implementation of the Sun network filesystem," *Proceedings of the summer 1985 USENIX conference*, pp. 119-130, 1985.
- [64] J. H. Howard, "An overview of the andrew file system," *Carnegie Mellon University, Information Technology Center*, 1988.
- [65] Suman De và cộng sự, "A Comparative Study on Distributed File Systems," *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough*, pp. 43-51, 2021.
- [66] Karun, A. Kala và cộng sự, "A review on hadoop—HDFS infrastructure extensions," *IEEE conference on information & communication technologies*, pp. 132-137, 2013.
- [67] D. Borthakur, "HDFS architecture guide," *Hadoop apache project*, pp. 1-13, 2008.
- [68] McKusick, Kirk, and Sean Quinlan, "GFS: evolution on fast-forward," *Communications of the ACM*, pp. 42-49, 2010.
- [69] Ghemawat, Sanjay và cộng sự, "The Google file system," *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pp. 29-43, 2003.
- [70] Strauch, Christof và cộng sự, "NoSQL databases," *Lecture Notes, Stuttgart Media University*, 2011.
- [71] Davoudian, Ali và cộng sự, "A survey on NoSQL stores," *ACM Computing Surveys*, pp. 1-43, 2018.
- [72] Han, Jing và cộng sự, "Survey on NoSQL database," *6th international conference on pervasive computing and applications*, pp. 363-366, 2011.
- [73] Labrinidis, A và cộng sự, "Challenges and opportunities with big data," *Proceedings of the VLDB Endowment*, pp. 2032-2033, 2012.
- [74] Sagiroglu, S; Duygu Sinanc, "Big data: A review," *2013 international conference on collaboration technologies and systems*, pp. 42-47, 2013.
- [75] Yang, Chaowei và cộng sự, "Big Data and cloud computing: innovation opportunities and challenges," *International Journal of Digital Earth*, pp. 13-53, 2017.
- [76] Kuyoro, Shade O. và cộng sự, "Cloud Computing Security Issues and Challenges," *International Journal of Computer Networks*, pp. 247-255, 2011.
- [77] Shahid, Muhammad và cộng sự, "Cloud Computing Security Models, Architectures, Issues and Challenges: A Survey: A Survey," *Smart Computing Review*, pp. 602-616, 2015.
- [78] Lane, Michael và cộng sự, "Managing the risks of data security and privacy in the cloud: a shared responsibility between the cloud service provider and the client organisation," *Bright Internet Global Summit*, 2017.

- [79] Aljoumah, E. và cộng sự, "SLA in cloud computing architectures: A comprehensive study," *International Journal of Grid Distribution Computing*, pp. 7-32, 2015.
- [80] Hussam Aladdin S. Ahmed và cộng sự, "A Review of Challenges and Security Risks of Cloud Computing," *Journal of Telecommunication, Electronic and Computer Engineering*, pp. 87-91, 2017.
- [81] Dahbur, Kamal và cộng sự, "A survey of risks, threats and vulnerabilities in cloud computing," *In Proceedings of the 2011 International conference on intelligent semantic Web-services and applications*, pp. 1-6, 2011.
- [82] Abdurachman, Edi và cộng sự, "Survey on threats and risks in the cloud computing environment," *Procedia Computer Science*, pp. 1325-1332, 2019.
- [83] A. T. Atieh, "Assuring the Optimum Security Level for Network, Physical and Cloud Infrastructure," *ScienceOpen Preprints*, 2021.
- [84] Ukil, Arijit và cộng sự, "A security framework in cloud computing infrastructure," *International Journal of Network Security & Its Applications*, pp. 11-24, 2013.
- [85] Kandukuri và cộng sự, "Cloud security issues," *IEEE international conference on services computing*, pp. 517-520, 2009.
- [86] Albugmi, Ahmed và cộng sự, "Data security in cloud computing," *Fifth international conference on future generation communication technologies*, pp. 55-59, 2016.
- [87] Muhammad Yasir Shabir và cộng sự, "Analysis of Classical Encryption Techniques in Cloud Computing," *TSINGHUA SCIENCE AND TECHNOLOGY*, pp. 102-113, 2016.
- [88] Kacha, Lynda và cộng sự, "An overview on data security in cloud computing," *Cybernetics Approaches in Intelligent Systems: Computational Methods in Systems and Software*, pp. 250-261, 2018.
- [89] Alejandro Peñuelas-Angulo và cộng sự, "Revocation in attribute-based encryption for fog-enabled internet of," *Internet of Things*, 2023.
- [90] Zhang, Yinghui và cộng sự, "Attribute-based encryption for cloud computing access control: A survey," *ACM Computing Surveys*, pp. 1-41, 2020.
- [91] Zhao, Feng và cộng sự, "A cloud computing security solution based on fully homomorphic encryption," *16th international conference on advanced communication technology*, pp. 485-488, 2014.
- [92] Jabbar, Ihsan và cộng sự, "Using fully homomorphic encryption to secure cloud computing," *Internet of things and cloud computing*, pp. 13-18, 2016.
- [93] Lianggui Liu; Qiuxia Chen, "A Novel Category Group Index Mechanism for Efficient Ranked Search of Encrypted Cloud Data," *IEEE Access*, pp. 1-12, 2020.

- [94] Wang, Yunling và cộng sự, "Secure searchable encryption: a survey," *Journal of communications and information networks*, pp. 52-65, 2016.
- [95] Bernaschi và cộng sự, "REMUS: A security-enhanced operating system," *ACM Transactions on Information and System Security*, pp. 36-61, 2002.
- [96] Fernandez, Eduardo B. và cộng sự, "Even more patterns for secure operating systems," *Proceedings of the 2006 conference on Pattern languages of programs*, pp. 1-9, 2006.
- [97] Mazhar Ali và cộng sự, "Security in cloud computing: Opportunities and challenges," *Information Sciences*, pp. 357-383, 2015.
- [98] Hamid Banirostam, Alireza Hedayati, Ahmad Khadem Zadeh, "Using Virtualization Technique to Increase Security and Reduce Energy Consumption in Cloud Computing," *International Journal of Research in Computer Science*, pp. 25-30, 2014.
- [99] Reshetova, E. và cộng sự, "Security of OS-Level Virtualization Technologies," *Lecture Notes in Computer Science*, pp. 77-93, 2014.
- [100] Li, Shih-Wei và cộng sự, "Protecting cloud virtual machines from hypervisor and host operating system exploits," *28th USENIX Security Symposium*, pp. 1357-1374, 2019.
- [101] Litchfield, Alan và cộng sự, "A Systematic Review of Vulnerabilities in Hypervisors and Their Detection," *A Systematic Review of Vulnerabilities in Hypervisors*, pp. 1-10, 2017.
- [102] Riabko, Andrii V và cộng sự, "Investigating the effect of virtual machine migration accounting on reliability using a cluster model," *Journal of Edge Computing*, pp. 37-63, 2023.
- [103] Reuben, Jenni Susan, "A survey on virtual machine security," *Helsinki University of Technology*, 2007.
- [104] Mathew, S. và cộng sự, "Overview of amazon web services," *Amazon Whitepapers*, 2014.
- [105] Hashemipour, S. và cộng sự, "Amazon web services (aws)—an overview of the on-demand cloud computing platform," *International Conference for Emerging Technologies in Computing*, pp. 40-47, 2020.
- [106] Singh, T. và cộng sự, "The effect of Amazon web services (AWS) on cloud-computing," *International Journal of Engineering Research & Technology*, pp. 480-482, 2021.
- [107] Copeland, M. và cộng sự, "Microsoft azure and cloud computing," *Microsoft Azure: Planning, Deploying, and Managing Your Data center in the Cloud*, pp. 3-26, 1025.
- [108] J. Mäenpää, "Cloud computing with the Azure platform," *Seminar on Internet Working*, 2009.

- [109] Hassan, M. và cộng sự, "Cloud Computing Services and Microsoft Azure. Why Microsoft Azure?," *SSRN 4103377*, 2022.
- [110] Madhuri, T. và cộng sự, "Microsoft Azure v/s Amazon AWS Cloud services: A comparative study," *International Journal of Innovative Research in Science*, pp. 3904-3908, 2016.
- [111] Zahariev, A., "Google app engine," *Helsinki University of Technology*, pp. 1-5, 2009.
- [112] Malawski, M. và cộng sự, "How to use Google App engine for free computing," *IEEE Internet Computing*, pp. 50-59, 2011.