

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC TP. HỒ CHÍ**  
**MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN**

---



**BÁO CÁO BỘ MÔN**  
**ĐIỆN TOÁN ĐÁM MÂY**

**Giảng viên hướng dẫn: Cao Tiến Thành**

**Trần Ngọc Vinh-21DH113413**

**Nguyễn Hoàng Phúc – 21DH114014**

**Lê Thành Ân – 21DH112304**

**Năm học 2023 -2024**

## LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến thầy Cao Tiến Thành – giảng viên bộ môn Điện Toán Đám Mây chuyên ngành An Ninh Mạng đã hướng dẫn chúng em tận tình xuyên suốt quá trình thực hiện báo cáo đồ án môn học này.

Em cũng xin gửi lời cảm ơn đến bạn bè trong cùng chuyên ngành đã luôn giúp đỡ nhóm chúng em trong quá trình học tập và nghiên cứu. Nhờ sự chia sẻ kinh nghiệm và những góp ý quý báu của các bạn, em đã có thể hoàn thiện bài báo cáo của mình một cách hiệu quả hơn.

Cuối cùng, em xin gửi lời cảm ơn đến gia đình, những người bạn thân thiết đã luôn động viên, tạo điều kiện cho em trong suốt thời gian học tập và nghiên cứu.

Em xin nhận tất cả những góp ý của Ban Giám Khảo, Quý Thầy/Cô và các bạn để hoàn thiện Báo cáo của mình hơn nữa. Trong quá trình học tập tại trường, chắc chắn sẽ không tránh khỏi những thiếu sót. Chính vì vậy, em mong được thầy/ cô nhận xét góp ý để mình ngày càng hoàn thiện hơn. Em xin chân thành cảm ơn!

## LỜI MỞ ĐẦU

Trong thời đại mà sự tiến lên của công nghệ kỹ thuật không ngừng phát triển, việc bảo vệ thông tin và dữ liệu trở nên càng trở nên quan trọng hơn bao giờ hết. Trong bối cảnh đó, môn học hãy cụ thể hơn là lĩnh vực Điện Toán Đám Mây đóng một vai trò quan trọng trong việc tối ưu hóa lưu trữ dữ liệu và có thể truy cập ở mọi nơi.

Báo cáo này được thực hiện nhằm mục đích tổng kết và phản ánh lại quá trình học tập và nghiên cứu xuyên suốt quá trình học bộ môn Điện Toán Đám Mây. Trong quá trình này, chúng em đã tiến hành nghiên cứu sâu rộng về các kỹ thuật và phương pháp thực hiện sử dụng đám mây. Báo cáo này không chỉ là kết quả của sự nỗ lực của chúng em mà còn là sự kết hợp giữa kiến thức lý thuyết và kinh nghiệm thực tế từ các bài tập và dự án thực hiện trong khóa học

Chúng em xin chân thành cảm ơn sự hướng dẫn và hỗ trợ từ giảng viên cũng như sự đồng lòng và cống hiến của tất cả các thành viên trong nhóm trong quá trình thực hiện báo cáo này.

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

**Giáo viên hướng dẫn**  
(Ký tên, đóng dấu)

# MỤC LỤC

<b>CHƯƠNG I: CƠ SỞ LÝ THUYẾT</b> .....	8
<b>I. Điện Toán Đám Mây:</b> .....	8
1. Khái niệm: .....	8
2. Ưu và nhược điểm của Điện Toán Đám Mây: .....	9
a/ Ưu điểm của Điện Toán Đám Mây: .....	9
b/ Nhược điểm của Điện Toán Đám Mây: .....	11
3. Phân loại dịch vụ của Điện Toán Đám Mây: .....	13
a/ Infrastructure as a Service (IaaS): .....	13
b/ Platform as a Service (PaaS): .....	13
c/ Software as a Service (SaaS): .....	13
4. Phân loại triển khai của Điện Toán Đám Mây: .....	15
a/ Public Cloud: .....	15
b/ Private Cloud: .....	15
c/ Hybrid Cloud: .....	15
d/ Community Cloud: .....	16
<b>II. MQTT:</b> .....	17
1. Khái niệm: .....	17
2. Tính năng và chức năng chính của MQTT: .....	18
3. Ưu và nhược điểm của MQTT: .....	20
4. Các thành phần có trong MQTT và cách thức hoạt động: .....	24
4.1. MQTT Client: .....	27
a/ Tính năng và chức năng chính của MQTT Client: .....	27
b/ Ưu và nhược điểm của MQTT Client: .....	29
4.2. MQTT Broker (MQTT Server): .....	30
a/ Tính năng và chức năng chính của MQTT Broker (MQTT Server): .....	30
b/ Ưu và nhược điểm của MQTT Broker (Server): .....	31
4.3. MQTT Topic: .....	33

a/ Tính năng và chức năng chính của MQTT Topic:.....	33
b/ Ưu và nhược điểm của MQTT Topic:.....	34
4.4. MQTT Payload:.....	36
a/ Tính năng và chức năng chính của MQTT Payload:.....	36
4.5. Quality of Service (QoS) trong MQTT: .....	37
a/ Tính năng và chức năng chính của QoS trong MQTT:.....	38
III. Dịch vụ CloudMQTT: .....	40
1. Khái niệm: .....	40
2. Tính năng và chức năng chính của CloudMQTT: .....	41
3. Ưu điểm và nhược điểm của dịch vụ CloudMQTT .....	42
IV. So sánh MQTT và các công nghệ tương đương:.....	45
V. Tổng kết .....	48
Chương II: TRIỂN KHAI .....	49
I. Triển khai dịch vụ:.....	49
II. Backup dữ liệu:.....	52
III. Bảo mật: .....	53
CHƯƠNG III: KẾT LUẬN .....	55
I. Đánh giá kết quả đạt được:.....	
1. Kết quả đạt được:.....	55
2. Những hạn chế của đồ án: .....	55
II. Kết luận: .....	56
1. Kết luận: .....	56
2. Hướng phát triển trong tương lai: .....	56

# DANH MỤC HÌNH ẢNH

Hình 1. Công nghệ Điện Toán Đám Mây đối với đời sống hiện nay.....	8
Hình 2. Những ưu điểm và nhược điểm của công nghệ Điện Toán Đám Mây.....	11
Hình 3. Phân loại các dịch vụ của Điện Toán Đám Mây theo trật tự.....	14
Hình 4. Phân loại các dịch vụ của Điện Toán Đám Mây theo trật tự 2.....	14
Hình 5. Các hình thức triển khai công nghệ Điện Toán Đám Mây.....	16
Hình 6. Mô hình về triển khai công nghệ Điện Toán Đám Mây.....	17
Hình 7. Giao thức MQTT trong công nghệ Điện Toán Đám Mây.....	18
Hình 8. Các thành phần có trong giao thức MQTT.....	25
Hình 9. MQTT Client đối với MQTT Broker .....	27
Hình 10. MQTT Broker và các công dụng đối với dịch vụ đời sống.....	30
Hình 11. Dịch vụ CloudMQTT .....	40
Hình 12. Giao diện của HiveMQ sau khi đăng nhập và triển khai Broker .....	49
Hình 13. Giao diện đăng ký và phân quyền người dùng .....	49
Hình 14. Giao diện chính của Websocket client .....	50
Hình 15. Giao diện tương tác topic .....	50
Hình 16. Giao diện tương tác của admin.....	51
Hình 17. Thiết bị giả lập đã nhận lệnh .....	51
Hình 18. Tải file project ZIP của cấu hình thiết bị.....	52
Hình 19. Lưu trữ trong máy tính .....	52
Hình 20. Ví dụ về lưu trữ trên dịch vụ điện toán đám mây.....	53
Hình 21. Import các thư viện.....	53
Hình 22. Kết nối máy chủ. Sử dụng SSL để mã hóa đầu cuối .....	53
Hình 23. Cấu hình kết nối MQTT, cấu hình SSL và chủ đề MQTT mà client giao tiếp ..	54

# CHƯƠNG I: CƠ SỞ LÝ THUYẾT

## I. Điện Toán Đám Mây:

### 1. Khái niệm:

Điện Toán Đám Mây (tiếng Anh: Cloud Computing), còn gọi là điện toán máy chủ ảo, là mô hình điện toán sử dụng công nghệ máy tính và phát triển dựa vào mạng Internet. Thay vì sở hữu và vận hành cơ sở hạ tầng máy tính và phần mềm của riêng mình, người dùng truy cập các tài nguyên này qua Internet từ các nhà cung cấp dịch vụ đám mây.



*Hình 1. Công nghệ Điện Toán Đám Mây đối với đời sống hiện nay*

Công nghệ Điện Toán Đám Mây có những đặc điểm chính như sau:

- **Tính linh hoạt:** Người dùng có thể truy cập tài nguyên đám mây mọi lúc, mọi nơi, chỉ cần có kết nối Internet.
- **Khả năng mở rộng:** Người dùng có thể dễ dàng tăng hoặc giảm tài nguyên đám mây tùy theo nhu cầu sử dụng.



- **Tiết kiệm chi phí:** Người dùng chỉ cần trả tiền cho tài nguyên đám mây mà họ sử dụng, thay vì phải đầu tư vào cơ sở hạ tầng riêng.
- **Độ tin cậy cao:** Các nhà cung cấp dịch vụ đám mây có cơ sở hạ tầng mạnh mẽ và dự phòng, giúp đảm bảo tính sẵn sàng và độ tin cậy cao cho các dịch vụ.
- **Bảo mật:** Các nhà cung cấp dịch vụ đám mây áp dụng các biện pháp bảo mật tiên tiến để bảo vệ dữ liệu của người dùng.

## 2. Ưu và nhược điểm của Điện Toán Đám Mây:

### a/ Ưu điểm của Điện Toán Đám Mây:

Những ưu điểm của công nghệ này là vô cùng nhiều, trong đó có thể kể đến như:

- **Truy cập toàn cầu:** Dịch vụ điện toán đám mây độc lập với nền tảng, có thể truy cập từ nhiều thiết bị kết nối internet. Điều này tăng tính di động và năng suất, cho phép nhân viên cộng tác hiệu quả trên các dự án.
- **Dung lượng lưu trữ nhiều hơn:** Điện toán đám mây cung cấp dung lượng lưu trữ và không gian gần như không giới hạn, miễn là bạn có đủ khả năng tài chính để chi trả cho các dịch vụ này
- **Thiết lập dễ dàng và nhanh chóng:** Với công nghệ điện toán đám mây, tất cả những gì người dùng cần là kết nối mạng, đăng ký với nhà cung cấp dịch vụ và thực hiện một số cấu hình đơn giản
- **Cập nhật tự động:** Các nhà cung cấp dịch vụ đám mây luôn cung cấp và chịu trách nhiệm cho vấn đề cập nhật phần mềm và bảo trì hệ thống, giúp người dùng tiết kiệm thời gian và công sức mà vẫn có khả năng tiếp cận với nhiều tính năng và công nghệ mới nhất về cloud computing

- **Chi phí hiệu quả:** Bạn chỉ phải trả tiền cho những gì bạn sử dụng và không phải chịu các chi phí khác như bảo trì phần cứng, phần mềm, phí cấp phép, và bảo mật.
- **Đạt được tính kinh tế theo quy mô:** Có thể tăng sản lượng hoặc năng suất với ít hệ thống hơn, do đó người dung và doanh nghiệp giảm chi phí trên mỗi đơn vị của một dự án hoặc sản phẩm và họ đề ra, tăng năng suất và hiệu quả làm việc khi triển khai với công nghệ này.
- **Giảm chi tiêu cho cơ sở hạ tầng công nghệ:** Dữ liệu và thông tin dễ dàng truy cập với chi phí trả trước tối thiểu theo phương thức trả tiền theo nhu cầu, tương tự như chỉ số đồng hồ điện trong nhà.
- **Hợp lý hóa quy trình kinh doanh:** Hoàn thành nhiều công việc hơn trong thời gian ngắn hơn với ít nguồn lực và nguồn tài nguyên hơn
- **Khả năng tiếp cận rộng khắp:** Dữ liệu và ứng dụng có thể truy cập mọi lúc, mọi nơi, sử dụng bất kỳ thiết bị điện toán thông minh nào, giúp cuộc sống trở nên dễ dàng hơn rất nhiều.
- **Giám sát dự án hiệu quả hơn:** Có thể theo dõi ngân sách và tiến độ dự án hiệu quả hơn, giúp dự án hoàn thành đúng hạn, song tăng năng suất và hiệu quả làm việc của người dung và doanh nghiệp
- **Cần ít đào tạo nhân sự hơn:** Việc thực hiện các dự án và kế hoạch đa phần cần nhiều nhân sự hơn để thực hiện nhiều công việc hơn. Nhân viên có thể tập trung vào các công việc có giá trị gia tăng hơn thay vì dành thời gian cho các nhiệm vụ CNTT tốn thời gian.
- **Giảm thiểu việc bảo trì và cấp phép phần mềm:** Vì không có quá nhiều tài nguyên máy tính tại chỗ, việc bảo trì trở nên đơn giản. Cập nhật và gia hạn hệ thống phần mềm được nhà cung cấp đám mây đảm nhận.

- **Cải thiện tính linh hoạt:** Có thể thực hiện những thay đổi nhanh chóng trong môi trường làm việc mà không gặp phải vấn đề nghiêm trọng nào.



*Hình 2. Những ưu điểm và nhược điểm của công nghệ Điện Toán Đám Mây*

#### **b/ Nhược điểm của Điện Toán Đám Mây:**

Song với những ưu điểm là những nhược điểm, ảnh hưởng ít nhiều đến trải nghiệm người dùng:

- **Phụ thuộc vào kết nối Internet:** Điểm hạn chế rõ ràng nhất của điện toán đám mây là nếu mất kết nối Internet, bạn cũng mất liên kết với đám mây, và do đó mất quyền truy cập vào dữ liệu và ứng dụng của mình. Đối với những tình huống bị ảnh hưởng đến tốc độ hay sự ổn định của Internet, công nghệ này sẽ gần như là vô dụng.
- **Vấn đề bảo mật và quyền riêng tư:** Nếu như trước đây các toán bộ các thông tin được lưu giữ trong các ổ cứng thì người dùng có thể chủ động bảo

vệ dữ liệu của mình. Còn đối với điện toán đám mây, các dữ liệu được đưa lên không gian lưu trữ của nhà cung cấp. Điều này vẫn tiềm ẩn nguy cơ bị đánh cắp thông tin nếu như hệ thống bảo mật của nhà cung cấp dịch vụ kém hoặc không có tính ổn định cao

- **Chi phí liên tục:** Mặc dù doanh nghiệp đã phần nào tiết kiệm chi phí chi tiêu cho toàn bộ cơ sở hạ tầng và quản lý nhưng họ cần tiếp tục thanh toán cho các dịch vụ được sử dụng trên điện toán đám mây, thường các dịch vụ này sẽ thường đi kèm với mức giá khá cao để duy trì lâu dài
- **Giới hạn của các nền tảng phát triển:** Không hẳn tất cả các nền tảng phát triển đều có sẵn trên đám mây, do nhà cung cấp dịch vụ có thể không hỗ trợ hoặc không biết về các giải pháp này. Điều này có thể gây khó khăn cho việc triển khai và phát triển ứng dụng, từ đó xả ra nhiều hạn chế để có thể tiếp tục mở rộng quá trình phát triển
- **Tốc độ lưu trữ:** Việc di chuyển dữ liệu lên điện toán đám mây và thực hiện các bản sao lưu có thể mất nhiều thời gian hơn khi so sánh với khi thực hiện trên hệ thống nội bộ. Nếu người dùng cần khôi phục toàn bộ máy chủ, có thể sẽ mất nhiều thời gian hơn. Tuy nhiên, các tệp và thư mục riêng lẻ có thể sẽ không bị ảnh hưởng.
- **Độ tin cậy của nhà cung cấp đám mây:** Các doanh nghiệp và người dùng cần phải phụ thuộc hoàn toàn và đặt niềm tin vào nhà cung cấp dịch vụ đám mây. Nếu có vấn đề gì xảy ra với nhà cung cấp, hoạt động của họ cũng sẽ bị ảnh hưởng không ít, từ đó dẫn đến việc có thể ảnh hưởng đến năng suất, hiệu quả và dữ liệu khổng lồ mà doanh nghiệp và người dùng đã cung cấp trên nền tảng này

### **3. Phân loại dịch vụ của Điện Toán Đám Mây:**

Điện Toán Đám Mây sẽ được phân loại theo 3 mô hình cung cấp dịch vụ dựa trên mức độ trừu tượng của các dịch vụ như IaaS, PaaS và SaaS. Mỗi loại sẽ có những đặc điểm đặc trưng như sau:

#### **a/ Infrastructure as a Service (IaaS):**

IaaS chuyên cung cấp cho người dùng nguồn tài nguyên cơ bản hạ tầng như mạng, máy tính ảo, không gian lưu trữ dữ liệu, CPU, RAM, HDD/SSD,... Khi sử dụng dịch vụ này, người dùng có thể thuê tài nguyên theo nhu cầu, mà không cần phải mua và duy trì phần cứng vật lý. Một số ví dụ về dịch vụ này có thể kể đến như Amazon EC2, Microsoft Azure Virtual Machines, Google Compute Engine,...

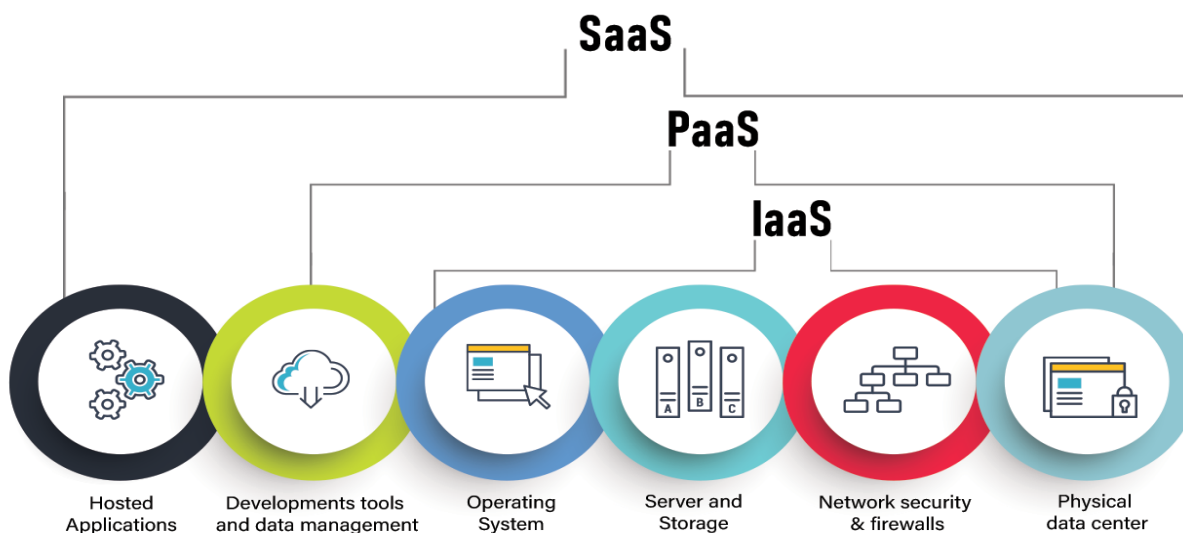
#### **b/ Platform as a Service (PaaS):**

PaaS cung cấp một nền tảng để phát triển và triển khai ứng dụng một cách hoàn chỉnh, cho phép người dùng lựa chọn các phần mềm và công cụ phát triển mong muốn mà không cần phải quan tâm tới việc cập nhật các phiên bản mới hay giới hạn phần cứng của thiết bị. Một số ví dụ của dịch vụ này có thể kể đến như Google App Engine, Heroku, Microsoft Azure App Service,...

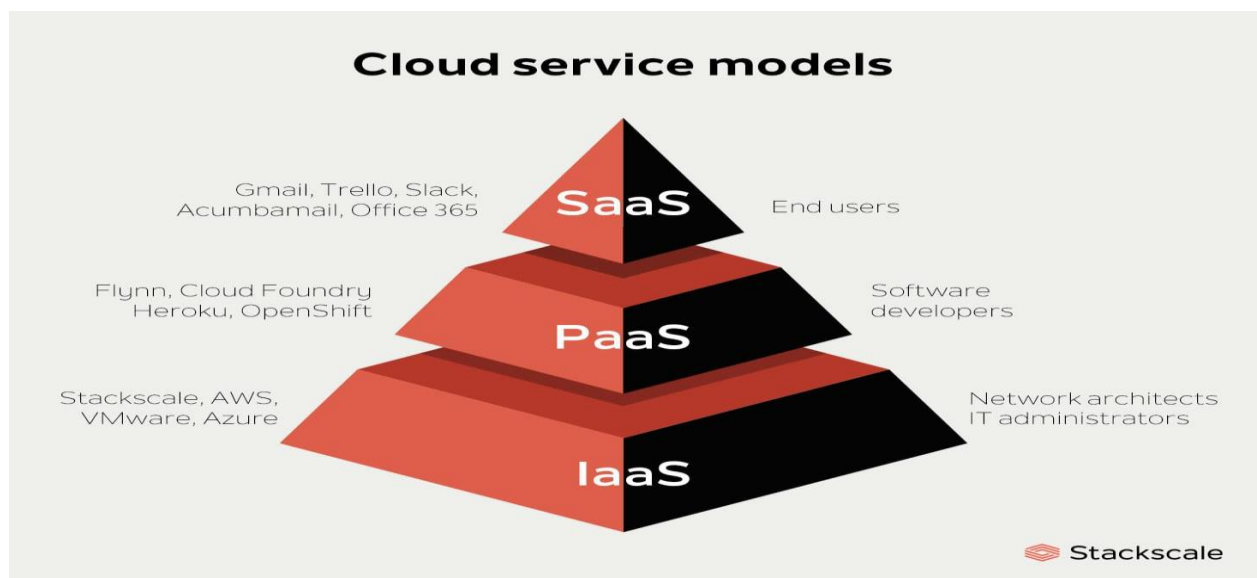
#### **c/ Software as a Service (SaaS):**

SaaS cho phép người dùng đăng ký, truy cập và sử dụng các ứng dụng, phần mềm thông qua Internet mà không am hiểu về các yếu tố kỹ thuật, cài đặt,... Các ứng dụng này sẽ luôn được cập nhật thường xuyên và bảo trì đầy đặn bởi nhà cung cấp dịch vụ. Một số ví dụ có thể kể đến như Google Workspace (Docs, Sheets, Slides), Microsoft 365 (Office 365), Salesforce,...

## SaaS vs. PaaS vs. IaaS



Hình 3. Phân loại các dịch vụ của Điện Toán Đám Mây theo trật tự



Hình 4. Phân loại các dịch vụ của Điện Toán Đám Mây theo trật tự 2

#### **4. Phân loại triển khai của Điện Toán Đám Mây:**

Điện Toán Đám Mây sẽ được phân loại dựa trên nhu cầu và mục đích của nhiều khách hàng khác nhau, từ đó ra đời các hướng đi và phương pháp triển khai khác nhau:

##### **a/ Public Cloud:**

Public Cloud là môi trường của Điện Toán Đám Mây được dùng cho tất cả các khách hàng, thường được hỗ trợ và phát triển bởi các nhà cung cấp dịch vụ lớn như AWS, Azure, GCP, và nhiều nhà cung cấp khác. Public Cloud được sử dụng rộng rãi bởi các tổ chức và cá nhân với mức độ linh hoạt và mở rộng cao, chi phí thanh toán linh hoạt theo mức độ sử dụng và nguồn tài nguyên luôn luôn được chia sẻ cũng như truy cập thông qua Internet.

##### **b/ Private Cloud:**

Private Cloud là môi trường Điện Toán Đám Mây được xây dựng lên và duy trì bởi các tổ chức hoặc doanh nghiệp mà không phụ thuộc vào bất kỳ các nhân nào khác. Đặc điểm chính của phân loại này là phù hợp cho các tổ chức có yêu cầu nghiêm ngặt về bảo mật, tuân thủ quy định hoặc có nhu cầu quản lý dữ liệu nội bộ, cung cấp sự kiểm soát và quản lý bảo mật cao hơn so với Public Cloud.

##### **c/ Hybrid Cloud:**

**Hybrid Cloud** là loại hình kết hợp giữa Public Cloud và Private Cloud, cho phép người sử dụng dịch vụ có khả năng sử dụng, di chuyển dữ liệu và ứng dụng giữa hai môi trường mà không bị ràng buộc giới hạn. Phân loại dịch vụ này cho phép

các tổ chức lựa chọn môi trường phù hợp nhất cho từng ứng dụng và nhu cầu công việc cụ thể, đảm bảo tính di động và sự kết nối giữa các môi trường khác nhau.

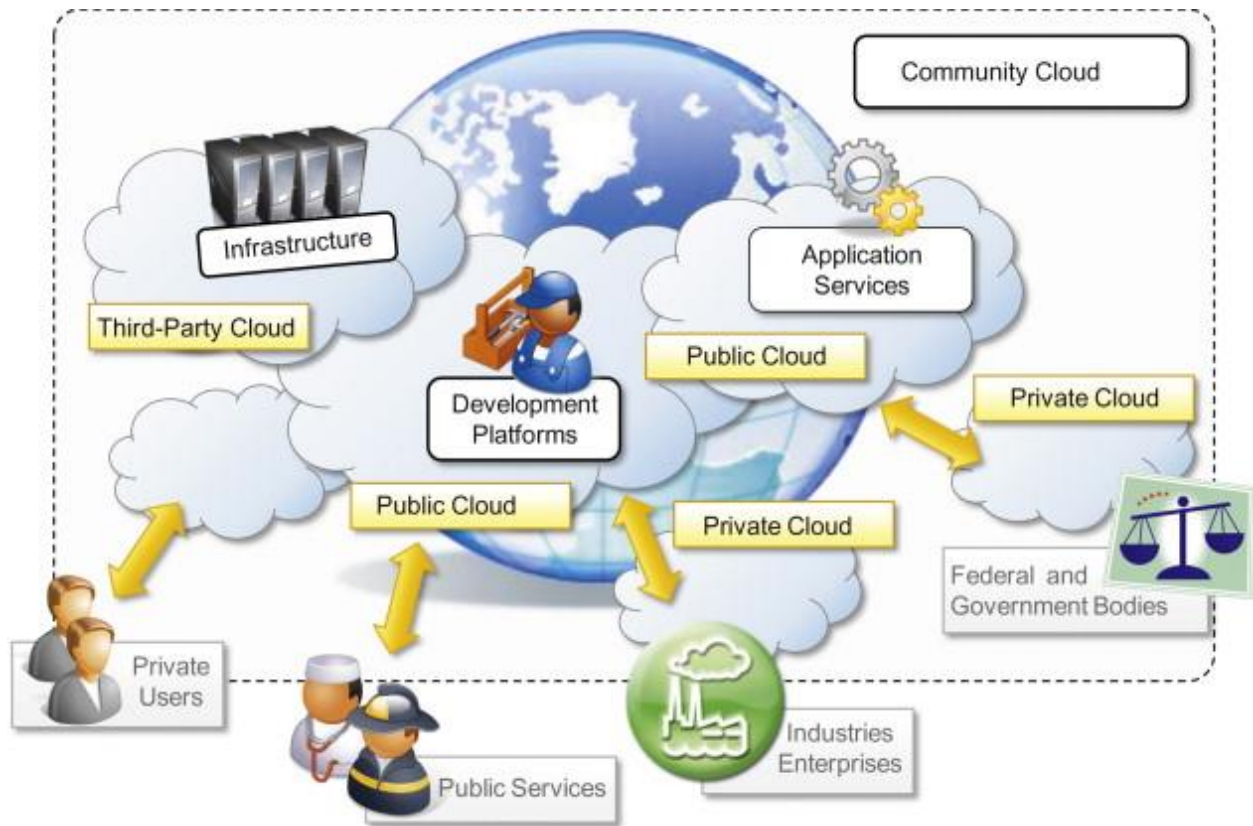
#### **d/ Community Cloud:**

Community Cloud là môi trường Điện Toán Đám Mây được chia sẻ công khai giữa các tổ chức và doanh nghiệp có cùng hướng đi hay cùng lĩnh vực, có chung mục đích và các quy định chung. Đối với phân loại dịch vụ này, các thành viên trong cộng đồng sẽ cùng nhau chia sẻ một số lượng tài nguyên và dịch vụ để đáp ứng các yêu cầu chung.



*Hình 5. Các hình thức triển khai công nghệ Điện Toán Đám Mây*



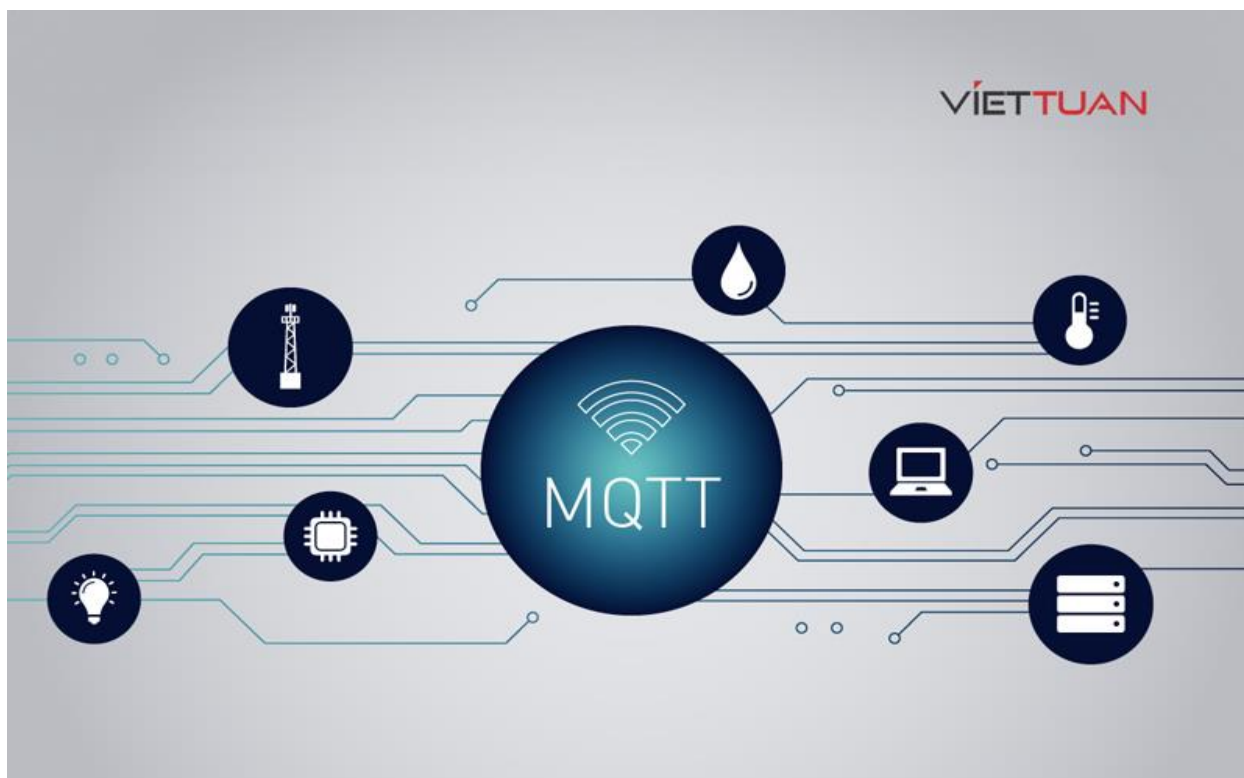


Hình 6. Mô hình về triển khai công nghệ Điện Toán Đám Mây

## II. MQTT:

### 1. Khái niệm:

MQTT (Message Queue Telemetry Transport) là một giao thức truyền thông điệp theo mô hình Cung cấp/Thuê bao (Publish/Subscribe). Giao thức này được sử dụng cho các thiết bị IoT với băng thông thấp, khả năng sử dụng được trong mạng lưới không ổn định và độ tin cậy cao. MQTT dựa trên một máy chủ môi giới (Broker) khá ít xử lý và được thiết kế có tính mở, đơn giản, dễ cài đặt và không đặc trưng cho ứng dụng cụ thể nào.



*Hình 7. Giao thức MQTT trong công nghệ Điện Toán Đám Mây*

## **2. Tính năng và chức năng chính của MQTT:**

**Nhẹ và băng thông thấp:** MQTT được thiết kế để hoạt động hiệu quả trên các mạng có băng thông hạn chế và các thiết bị có tài nguyên hạn chế về CPU và pin. Giao thức này sử dụng mô hình Publish/Subscribe (Pub/Sub), giúp giảm thiểu tải trọng mạng và tiết kiệm năng lượng cho các thiết bị IoT.

**Mô hình Publish/Subscribe (Pub/Sub):** MQTT hoạt động dựa trên mô hình Pub/Sub, trong đó Publisher (người đăng tin) là nguồn gốc của các thông điệp và Publisher gửi các thông điệp tới các chủ đề (topics) trên broker. Subscriber (người nhận tin) sẽ đăng ký để nhận các thông điệp từ các chủ đề mà họ quan tâm. Subscriber cũng đảm nhận vai trò nhận các thông điệp mà broker chuyển tiếp từ các Publisher đã gửi.

**Độ tin cậy và QoS (Quality of Service):** MQTT hỗ trợ ba cấp độ QoS để đảm bảo độ tin cậy của việc truyền thông, ba cấp độ ấy là QoS 0 (At most once) - Tin nhắn được gửi đi mà không có bất kỳ đảm bảo nào, có thể bị mất hoặc nhận được nhiều lần. QoS 1 (At least once) - Broker đảm bảo rằng mỗi tin nhắn được gửi đi ít nhất một lần. Tuy nhiên, có thể nhận được tin nhắn trùng lặp. QoS 2 (Exactly once) - Broker đảm bảo mỗi tin nhắn được gửi đi chỉ nhận được một lần duy nhất. Đây cũng chính là cấp độ đảm bảo tin cậy cao nhất nhưng có thể ảnh hưởng đến hiệu suất mạng.

**Kết nối duy trì và duy nhất:** MQTT cho phép các thiết bị duy trì kết nối lâu dài với broker MQTT. Điều này giúp giảm thiểu thời gian và tài nguyên cần thiết cho việc thiết lập và giải phóng kết nối. Mỗi thiết bị có thể có một Client ID duy nhất để xác định trong hệ thống MQTT.

**Chủ đề và phân tán linh hoạt:** MQTT sử dụng chủ đề để phân loại và định hướng thông điệp. Các chủ đề có thể được cấu trúc theo các cấu trúc hệ thống, ví dụ như home/livingroom/temperature. Người dùng có thể đăng ký để nhận các thông điệp từ một hoặc nhiều chủ đề, cho phép phân tán dữ liệu linh hoạt và hiệu quả.

**Tính bảo mật cao:** MQTT hỗ trợ các phương pháp bảo mật như TLS/SSL để bảo vệ dữ liệu truyền tải giữa các thiết bị và broker. Các phiên bản mới của MQTT cũng hỗ trợ các cơ chế xác thực như username/password để bảo vệ truy cập vào broker.

**Ứng dụng và tính linh hoạt:** MQTT được áp dụng rộng rãi trong các ứng dụng IoT như điều khiển thiết bị, giám sát môi trường, thu thập dữ liệu cảm biến và các ứng dụng có yêu cầu độ trễ thấp và độ tin cậy cao. Được sử dụng trong các hệ thống lớn, nơi có hàng triệu thiết bị IoT cần giao tiếp một cách hiệu quả và đáng tin cậy.

**Dễ dàng sử dụng:** MQTT sử dụng giao thức đơn giản, dễ hiểu và dễ triển khai, phù hợp với nhiều đối tượng lập trình viên, sở hữu nhiều tài liệu và các hướng dẫn chi tiết về MQTT giúp người dùng dễ dàng học hỏi và sử dụng. MQTT cũng bao gồm nhiều công cụ hỗ trợ: Có nhiều công cụ hỗ trợ việc phát triển và sử dụng MQTT như MQTT Broker, MQTT Client, MQTT Explorer,...

**Phi tập trung:** Không có điểm hỏng học duy nhất, giúp hệ thống hoạt động ổn định hơn.

**Thời gian thực:** Phù hợp cho các ứng dụng đòi hỏi truyền tải dữ liệu thời gian thực.

**Hỗ trợ nhiều nền tảng:** Có thể được sử dụng trên nhiều hệ điều hành và nền tảng phần cứng khác nhau.

**Miễn phí và mã nguồn mở:** Giao thức MQTT là miễn phí và mã nguồn mở, giúp người dùng dễ dàng sử dụng và tùy chỉnh.

### **3. Ưu và nhược điểm của MQTT:**

Ưu điểm của MQTT là vô cùng nhiều, một số ưu điểm nổi bật có thể kể đến như:

#### **Hiệu quả trong việc truyền tải thông tin:**

- **Tiêu thụ băng thông thấp:** MQTT sử dụng gói tin nhỏ gọn và có cấu trúc đơn giản, giúp giảm thiểu lượng dữ liệu cần thiết để truyền tải. Điều này rất quan trọng đối với các thiết bị IoT có băng thông hạn chế, chẳng hạn như các thiết bị chạy bằng pin.
- **Tài nguyên hệ thống thấp và tiết kiệm điện năng:** Thành phần này của MQTT không đòi hỏi quá nặng nề về yêu cầu của tài nguyên phần cứng, điều

phù hợp cho các thiết bị nhúng và các thiết bị trong môi trường IOT chạy bằng pin.

- **Trọng lượng gói tin nhẹ:** Sử dụng những gói tin nhỏ gọn, giúp giảm thiểu dữ liệu phụ và truyền tải thông tin một cách hiệu quả, đặc biệt là dành cho việc sử dụng mạng có băng thông thấp và nguồn tài nguyên hạn chế.

#### **Độ tin cậy cao:**

- **Chất lượng dịch vụ đảm bảo tốt (Quality of Service):** MQTT cung cấp ba mức độ chất lượng dịch vụ (QoS) để đảm bảo phân phối tin nhắn đáng tin cậy. QoS 1 đảm bảo rằng tin nhắn được gửi ít nhất một lần, QoS 2 đảm bảo rằng tin nhắn được gửi chính xác một lần và QoS 3 đảm bảo rằng tin nhắn được gửi chính xác một lần theo thứ tự.
- **Khả năng lưu trữ và chuyển tiếp tin nhắn:** Hỗ trợ lưu trữ và chuyển tiếp các tin nhắn khi các thiết bị Client đang ở ngoại tuyến, đảm bảo rằng không dữ liệu quan trọng nào bị bỏ sót và tin nhắn sẽ được gửi đi ngay khi thiết bị nhận đang ngoại tuyến ở thời điểm gửi.
- **Khả năng giám sát và quản lý chặt chẽ:** Cung cấp các công cụ để giám sát và quản lý hệ thống hiệu quả, giúp người sử dụng dịch vụ có thể dễ dàng phát hiện, khắc phục sự cố kịp thời.

#### **Tính bảo mật chặt chẽ:**

- **Khả năng mã hóa:** Hỗ trợ mã hóa bằng giao thức TLS/ SSL để bảo vệ dữ liệu trong quá trình truyền tải gói tin hoặc tin nhắn, nhằm đảm bảo tính bảo mật và riêng tư của người sử dụng.
- **Xác thực và ủy quyền người dùng:** Cung cấp cơ chế để xác thực và ủy quyền để có thể kiểm soát quyền truy cập vào hệ thống hiệu quả hơn, một số cơ chế

xác thực như username/ password ,chứng chỉ x.509 và OAuth là những phương thức được sử dụng phổ biến.

- Tính cô lập và tách biệt: Cũng nhờ việc sử dụng mô hình Pub/ Sub mà các Client có thể cô lập, giảm thiểu nguy cơ lây lan sự cố và tấn công mạng.

### **Tính mở rộng và sự linh hoạt:**

- Dễ dàng triển khai và sử dụng: Có thể quản lý hàng ngàn thiết bị và hàng triệu các tin nhắn mỗi giây để phù hợp hơn với các mô hình hệ thống lớn.
- Sử dụng giao thức đơn giản nhưng hiệu quả: Có cấu trúc giao thức đơn giản, dễ dàng để học hỏi, triển khai và tích hợp với các ứng dụng khác nhau.
- Sở hữu cộng đồng hỗ trợ to lớn: Có cộng đồng người dùng đông đảo và nhà phát triển lớn, hỗ trợ cung cấp nguồn tài nguyên phong phú và dịch vụ hỗ trợ kỹ thuật chuyên sâu.
- Hỗ trợ đa ngôn ngữ và đa nền tảng: Có sẵn trên nhiều ngôn ngữ lập trình và hệ điều hành khác nhau, giúp dễ dàng tích hợp vào các hệ thống hiện có, ngoài ra MQTT Broker còn sở hữu nhiều thư viện và công cụ hỗ trợ MQTT cho nhiều ngôn ngữ lập trình khác nhau, giúp đơn giản hóa việc phát triển ứng dụng IoT.

Song với những ưu điểm thì MQTT còn sở hữu cho mình nhiều điểm hạn chế và điểm yếu của giao thức này:

### **Yêu cầu về tài nguyên sử dụng:**

- Tài nguyên phần cứng cao: Một số Broker (Server) mạnh mẽ có thể yêu cầu, đòi hỏi việc tài nguyên phần cứng đáng kể, việc quản lý nhiều kết nối và luồng thông tin lớn có thể sẽ gây hao tổn tài nguyên của CPU, đặc biệt khi hệ thống có lượng Client lớn hoặc lưu lượng dữ liệu cao.

- Bảo trì tài nguyên: Các Broker cần được quản lý và bảo trì, điều này cần phải có kiến thức và kỹ năng chuyên môn về MQTT và hệ thống mạng.

### **Nguy cơ bị tấn công:**

- Tấn công từ chối dịch vụ (DoS): Do tính chất của giao thức là mở rộng và công cộng, MQTT có thể bị tấn công DoS nhằm gây gián đoạn hoạt động của hệ thống.
- Tấn công giả mạo: Những tin tặc, kẻ tấn công có thể giả mạo danh tính của thiết bị Client hợp pháp nhằm truy cập và thao tác trái phép trên hệ thống.
- Tấn công trung gian: Nếu không sử dụng mã hóa giao thức, việc truyền tải dữ liệu trên MQTT cũng tiềm ẩn nguy cơ bị đánh cắp hoặc sửa đổi bởi tin tặc.

### **Giới hạn trong tính năng sử dụng:**

- Không phù hợp cho hoàn toàn mọi ứng dụng: MQTT được thiết kế cho các ứng dụng nhẹ và có độ trễ thấp nhưng tuy vậy, không có nghĩa các ứng dụng yêu cầu truyền dữ liệu lớn hoặc xử lý phức tạp và phù hợp cho giao thức này.
- Thiếu tính năng nâng cao cho người dùng: Một số tính năng nâng cao mà các giao thức khác có thể hỗ trợ tốt hơn hẳn như xử lý luồng dữ liệu phức tạp hoặc tích hợp với các hệ thống quản lý dữ liệu lớn.

### **Khó khăn trong việc sửa chữa lỗi:**

- Tính phi đồng bộ: Do bản chất phi đồng bộ của MQTT, việc theo dõi và gỡ lỗi sự cố trong hệ thống có thể gặp nhiều khó khăn khi các tiến trình xử lý dữ liệu và các hoạt động không chờ nhau hoàn thành.

- Các yếu tố ngoài lề: Hiệu suất và độ tin cậy của hệ thống MQTT sẽ phải phụ thuộc vào nhiều yếu tố khác như phần cứng, đường truyền mạng, ứng dụng của Client, từ đó khiến cho việc xác định và giải quyết vấn đề trở nên khó khăn và phức tạp hơn.
- Khả năng tương tác giữa các hệ thống khác: MQTT không tương thích trực tiếp với các hệ thống khác không sử dụng giao thức MQTT, do đó cần sử dụng các bộ chuyển đổi hoặc gateway để kết nối với các hệ thống khác.

#### **4. Các thành phần có trong MQTT và cách thức hoạt động:**

Giao thức MQTT có các thành phần chính tùy vào vai trò quan trọng trong việc hoạt động và triển khai giao thức ấy. Một số thành phần phổ biến có thể được kể đến như:

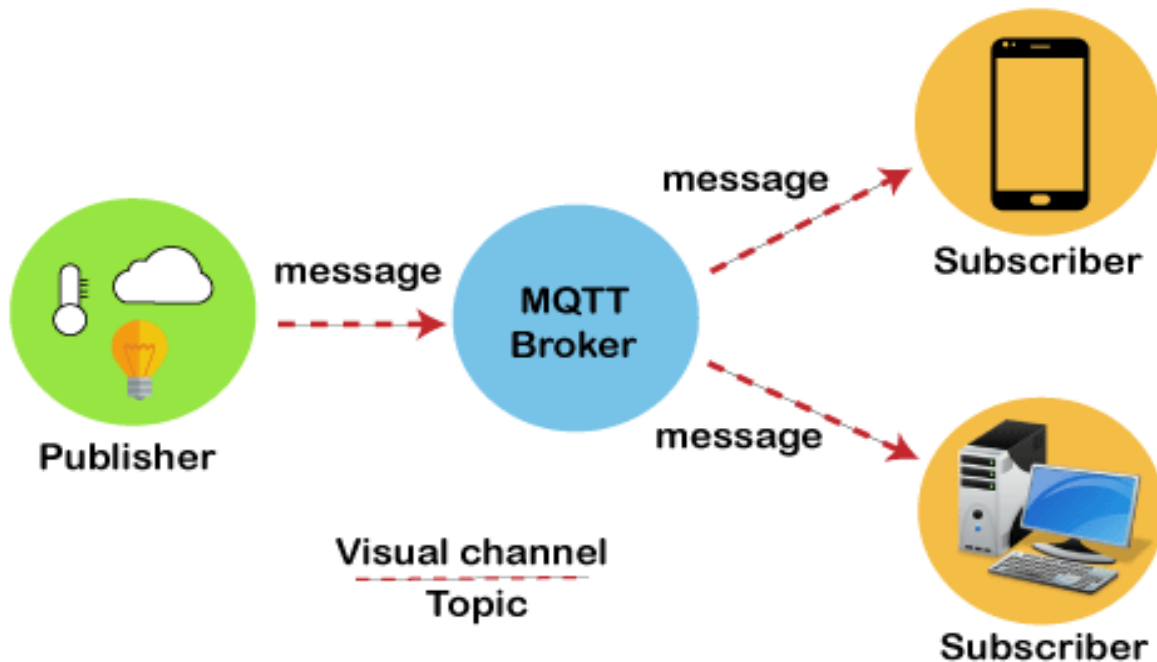
- Client (Máy khách)
- Broker (Máy chủ trung tâm)
- Topic (Chủ đề)
- Payload (Dữ liệu)
- Quality of Service (QoS) (Mức chất lượng dịch vụ)

Ngoài ra, còn có một số thành phần bổ sung khác trong MQTT như:

- Session (Phiên)
- Subscription (Đăng ký)
- Will Message (Tin nhắn di chúc)



# MQTT Architecture



Hình 8. Các thành phần có trong giao thức MQTT

Cách thức hoạt động của MQTT sẽ như sau:

**Kết nối** → **Đăng ký và Xuất bản** → **Thông điệp và MQTT Topic** → **QoS** → **Giải quyết phiên**

**Kết nối (Connection):** MQTT Client khởi tạo một kết nối TCP hoặc TLS với MQTT Broker (thông thường ở cổng 1883 hoặc 8883 nếu sử dụng TLS). Sau khi kết nối thành công, MQTT Client có thể gửi yêu cầu kết nối (Connect Packet) đến Broker.

**Đăng ký (Subscribe) và Xuất bản (Publish):** MQTT Client có thể đăng ký (Subscribe) để nhận các tin nhắn từ một hoặc nhiều MQTT Topic. Khi đăng ký, Client chỉ cần cung cấp MQTT Topic mà nó quan tâm. MQTT Client có thể gửi các tin nhắn (Publish) tới một MQTT Topic cụ thể trên Broker. Tin nhắn này có

thể là dữ liệu, lệnh điều khiển, hoặc bất kỳ loại thông tin nào mà Client muốn chia sẻ.

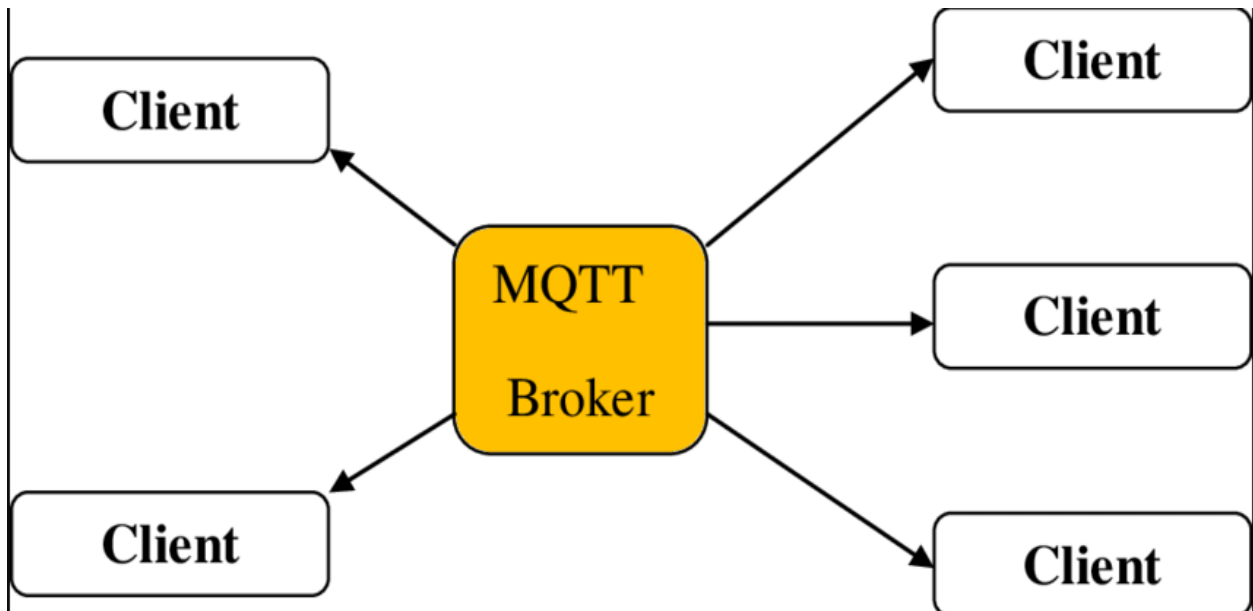
**Thông điệp (Message) và MQTT Topic:** MQTT sử dụng các MQTT Topic để phân loại và phân phối dữ liệu. Mỗi tin nhắn MQTT có thể được đánh dấu với một MQTT Topic duy nhất, giúp Broker biết tin nhắn này cần được gửi tới những Clients đã đăng ký nhận từ Topic đó.

**Chất lượng dịch vụ (Quality of Service - QoS):** MQTT hỗ trợ ba mức độ QoS khác nhau để đảm bảo chất lượng giao tiếp. QoS 0 (At most once) là khi Broker gửi tin nhắn một lần duy nhất tới Subscriber mà không cần xác nhận. QoS 1 (At least once) là khi Broker gửi tin nhắn tới Subscriber ít nhất một lần, đảm bảo ít nhất một tin nhắn được nhận, nhưng có thể bị lặp lại. QoS 2 (Exactly once) là khi Broker đảm bảo rằng mỗi tin nhắn được gửi một lần duy nhất tới Subscriber, bằng cách duy trì một quy trình phức tạp hơn để đảm bảo giao tiếp chính xác.

**Giải quyết phiên (Session Management):** MQTT hỗ trợ việc quản lý phiên (session) giữa Client và Broker. Khi một Client kết nối lại sau khi mất kết nối, Broker có thể tiếp tục gửi tin nhắn những MQTT Topic mà Client đã đăng ký trước đó, nếu phiên của Client vẫn được duy trì.

#### 4.1. MQTT Client:

MQTT Client là một phần mềm hoặc thiết bị được cài đặt trên thiết bị IoT (Internet of Things) để kết nối với MQTT Broker và giao tiếp theo giao thức MQTT. MQTT Client đóng vai trò như một "đại lý" cho thiết bị IoT, cho phép nó gửi và nhận thông tin từ MQTT Broker.



*Hình 9. MQTT Client đối với MQTT Broker*

##### **a/ Tính năng và chức năng chính của MQTT Client:**

**Kết nối (Connect):** MQTT Client có khả năng thiết lập kết nối đến một MQTT Broker bằng cách gửi yêu cầu kết nối và thông tin đăng nhập (nếu có).

**Ngắt kết nối (Disconnect):** Client có thể ngắt kết nối với Broker khi không cần thiết, giải phóng tài nguyên và thông báo cho Broker về việc ngắt kết nối.

**Đăng ký và Hủy đăng ký (Subscribe and Unsubscribe):** MQTT Client có thể đăng ký (subscribe) vào các chủ đề (topics) để nhận tin nhắn từ Broker và hủy đăng ký (unsubscribe) khi không cần thiết nữa.

**Xuất bản (Publish):** Client có khả năng xuất bản (publish) các tin nhắn lên các chủ đề tới MQTT Broker để các Client khác có thể nhận được.

**Nhận tin nhắn (Receive Messages):** MQTT Client có khả năng nhận tin nhắn từ MQTT Broker dựa trên các chủ đề mà Client đã đăng ký.

**Quản lý Chất lượng Dịch vụ (QoS Management):** Client có thể xác định mức độ Chất lượng Dịch vụ (QoS) cho từng tin nhắn xuất bản và tin nhắn nhận, đảm bảo mức độ tin cậy và độ tin cậy của giao tiếp.

**Quản lý phiên làm việc (Session Management):** MQTT Client có thể duy trì phiên làm việc với MQTT Broker, bao gồm lưu trữ trạng thái phiên để hỗ trợ duy trì kết nối và đảm bảo giao nhận tin nhắn một cách liên tục.

**Bảo mật và Chứng thực (Security and Authentication):** Client cung cấp các thông tin chứng thực (username, password) cho MQTT Broker để xác thực và bảo mật thông tin giao tiếp giữa hai bên.

**Quản lý lưu trữ (Storage Management):** MQTT Client có khả năng quản lý lưu trữ tạm thời các tin nhắn chờ gửi hoặc nhận từ MQTT Broker, đảm bảo tính nhất quán và độ tin cậy của hệ thống.

## **b/ Ưu và nhược điểm của MQTT Client:**

MQTT Client có những ưu điểm chính tương tự như đã nêu trên, một số nổi bật riêng của thành phần này là:

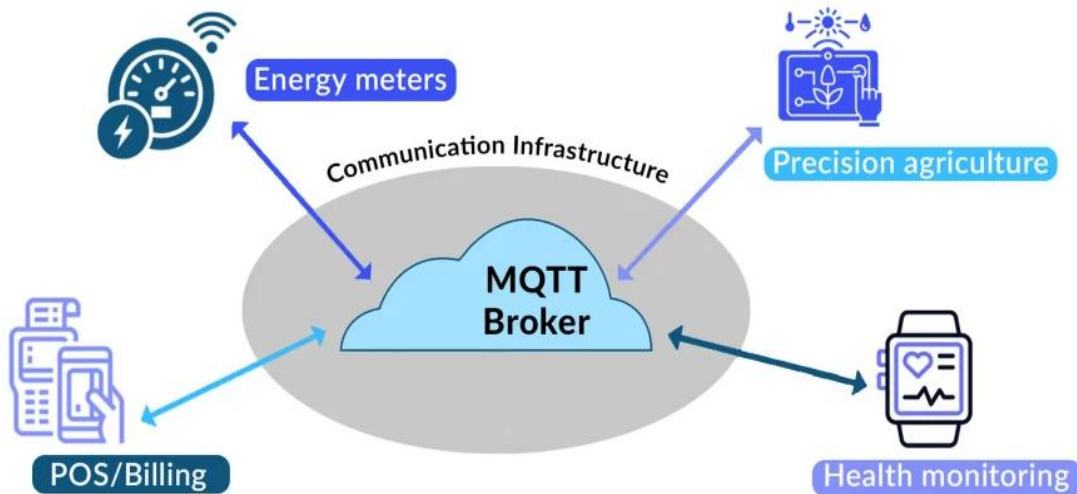
- **Tính linh hoạt và đa dạng:** MQTT Client có thể là các thiết bị nhúng, thiết bị cảm biến, thiết bị điều khiển, hoặc các ứng dụng phần mềm. Điều này cho phép nhiều loại thiết bị và ứng dụng khác nhau tham gia vào mạng MQTT một cách linh hoạt.
- **Tiêu tốn tài nguyên ít hơn:** So với MQTT Broker, MQTT Client thường tiêu tốn ít tài nguyên hệ thống hơn vì chúng chỉ cần duy trì kết nối đến Broker và xử lý các tin nhắn cụ thể mà chúng tham gia.
- **Khả năng tương tác người dùng:** MQTT Client có thể được tích hợp vào các ứng dụng người dùng để tương tác và điều khiển các thiết bị IoT một cách thuận tiện. Điều này giúp cải thiện trải nghiệm người dùng và quản lý thiết bị một cách hiệu quả.

Nhược điểm của MQTT Client là:

- **Phụ thuộc vào kết nối mạng ổn định:** MQTT Client phải duy trì kết nối mạng ổn định để có thể giao tiếp với MQTT Broker một cách liên tục và tin cậy. Sự mất kết nối có thể dẫn đến mất dữ liệu và sự cố trong hoạt động của hệ thống IoT.
- **Thiết kế và triển khai phức tạp:** Đối với các thiết bị nhúng hoặc thiết bị có tài nguyên hạn chế, việc triển khai MQTT Client có thể đòi hỏi sự tối ưu và khó khăn trong việc quản lý tài nguyên. Điều này yêu cầu các nhà phát triển cần có kiến thức vững về cả phần cứng và phần mềm để triển khai một cách hiệu quả.

## 4.2. MQTT Broker (MQTT Server):

MQTT Broker (MQTT Server) là thành phần trung tâm chính của hệ thống giao thức MQTT, đóng vai trò như một trung gian kết nối và truyền tải thông tin giữa các thiết bị (client) trong mạng lưới Internet vạn vật (IoT).



Hình 10. MQTT Broker và các công dụng đối với dịch vụ đời sống

### a/ Tính năng và chức năng chính của MQTT Broker (MQTT Server):

**Quản lý kết nối (Connection Management):** MQTT Broker quản lý các kết nối từ các MQTT Clients, bao gồm thiết lập, duy trì và ngắt kết nối.

**Đăng ký và Hủy đăng ký (Subscribe and Unsubscribe):** MQTT Broker quản lý các yêu cầu đăng ký và hủy đăng ký từ Clients đối với các chủ đề (topics).

**Xuất bản và Đăng nhận tin (Publish and Deliver Messages):** MQTT Broker nhận các tin nhắn từ Clients và gửi chúng đến các Clients đã đăng ký cho các chủ đề tương ứng.

**Quản lý Chất lượng Dịch vụ (QoS Management):** MQTT Broker quản lý và thực hiện các mức độ Chất lượng Dịch vụ (QoS) được yêu cầu bởi các Clients trong quá trình gửi và nhận tin nhắn.

**Bảo mật và Chứng thực (Security and Authentication):** MQTT Broker cung cấp các cơ chế để xác thực và bảo mật thông tin giữa Broker và Clients, bao gồm sử dụng username/password hoặc các phương pháp chứng thực khác như TLS/SSL.

**Quản lý phiên làm việc (Session Management):** MQTT Broker quản lý phiên làm việc của các Clients, bao gồm lưu trữ trạng thái phiên để hỗ trợ duy trì kết nối và đảm bảo giao nhận tin nhắn một cách liên tục khi có mất kết nối tạm thời.

**Quản lý lưu trữ (Storage Management):** Broker quản lý lưu trữ tạm thời các tin nhắn chờ gửi hoặc nhận, đảm bảo tính nhất quán và độ tin cậy của hệ thống.

#### **b/ Ưu và nhược điểm của MQTT Broker (Server):**

MQTT Broker có những ưu và nhược điểm tương đồng khá nhiều so với MQTT Client, một số ưu điểm khác biệt để nhận biết thành phần này là:

- **Quản lý trung tâm và điều phối:** MQTT Broker là thành phần trung tâm trong mạng MQTT, quản lý và điều phối giao tiếp giữa các Clients. Điều này giúp tăng tính linh hoạt và quản lý hiệu quả cho hệ thống IoT.
- **Xử lý tin nhắn hiệu quả:** Broker có khả năng nhận, lưu trữ tạm thời và phân phối tin nhắn đến các Clients theo các chủ đề (topics) khác nhau. Điều này đảm bảo tính nhất quán và độ tin cậy của dữ liệu trong hệ thống.
- **Bảo mật cao:** MQTT Broker cung cấp các cơ chế bảo mật để xác thực và mã hóa dữ liệu, bảo vệ thông tin giao tiếp giữa các Clients và Broker. Các

phương pháp bảo mật như TLS/SSL và xác thực username/password đều được hỗ trợ.

- **Quản lý chất lượng dịch vụ (QoS):** Broker quản lý và hỗ trợ các mức độ QoS khác nhau (QoS 0, 1, 2), giúp đảm bảo độ tin cậy và độ khả dụng của hệ thống theo yêu cầu của từng ứng dụng IoT.
- **Quản lý kết nối và phiên làm việc:** MQTT Broker duy trì quản lý các kết nối từ MQTT Clients và lưu trữ trạng thái phiên làm việc. Điều này hỗ trợ duy trì kết nối liên tục và đảm bảo giao tiếp liên tục giữa các thiết bị và ứng dụng.

Nhược điểm của thành phần MQTT Broker (Server) là:

- **Yêu cầu tài nguyên hệ thống cao:** MQTT Broker phải xử lý nhiều kết nối đồng thời và quản lý lưu trữ tin nhắn, đòi hỏi tài nguyên hệ thống khá lớn. Điều này có thể làm giảm hiệu suất của hệ thống nếu không được thiết kế và cấu hình tối ưu.
- **Điểm yếu về hiệu năng:** Khi số lượng Clients và lưu lượng tin nhắn lớn, MQTT Broker có thể trở thành điểm hạn chế về hiệu suất. Việc quản lý đồng thời các kết nối và phân phối tin nhắn yêu cầu sự tối ưu hóa để đảm bảo đáp ứng nhanh chóng và hiệu quả.



### 4.3. MQTT Topic:

MQTT Topic là một định danh hoặc chuỗi ký tự được sử dụng để định danh một loại dữ liệu hoặc một loại sự kiện cụ thể trong hệ thống MQTT (Message Queuing Telemetry Transport). Topic cho phép các thiết bị MQTT Clients đăng ký để nhận các tin nhắn từ MQTT Broker hoặc gửi các tin nhắn tới MQTT Broker.

#### a/ Tính năng và chức năng chính của MQTT Topic:

**Phân loại và định tuyến thông điệp:** Topic hoạt động như một kênh giao tiếp, giúp phân loại và định tuyến tin nhắn được gửi bởi các thiết bị (Publisher) đến các thiết bị quan tâm (Subscriber) một cách hiệu quả. Mỗi Topic có một tên duy nhất, thể hiện chủ đề hoặc loại dữ liệu cụ thể mà nó mang theo. Ví dụ:

"sensor/temperature" hoặc "device/status". Publisher gán dữ liệu vào Topic phù hợp, sau đó gửi đến MQTT Broker. Broker sẽ phân phối tin nhắn đến tất cả Subscriber đã đăng ký với Topic đó.

**Cấu trúc phân cấp:** Topic có thể được tổ chức theo cấu trúc phân cấp, sử dụng dấu phân cách "/" để phân chia thành các cấp con. Ví dụ:

"home/living\_room/temperature" hoặc "vehicles/fleet/truck1/fuel\_level". Cấu trúc này cho phép phân loại dữ liệu chi tiết hơn và dễ dàng quản lý các nhóm thiết bị liên quan.

**Wildcard:** MQTT hỗ trợ ký tự wildcard "#" và "+". Wildcard cho phép Subscriber nhận nhiều tin nhắn từ các Topic khác nhau. Ví dụ: "#" đăng ký tất cả các Topic, "+" đăng ký tất cả các cấp con của một Topic cụ thể.

**Khả năng lưu trữ và chuyển tiếp:** Một số MQTT Broker cung cấp tính năng lưu trữ tin nhắn đã được xuất bản trong một khoảng thời gian nhất định. Subscriber mới đăng ký Topic có thể nhận được tin nhắn đã lưu trữ trước đó. Một số Broker

khác hỗ trợ chuyển tiếp tin nhắn đến các Broker khác để mở rộng phạm vi truy cập.

**Bảo mật:** MQTT hỗ trợ các cơ chế bảo mật như tên người dùng, mật khẩu và danh sách kiểm soát truy cập (ACL) để kiểm soát quyền truy cập vào Topic và dữ liệu. Điều này giúp đảm bảo tính bảo mật và toàn vẹn dữ liệu trong hệ thống.

## **b/ Ưu và nhược điểm của MQTT Topic:**

Ưu điểm của MQTT Topic là:

- **Tính hiệu quả:** MQTT Topic sử dụng mô hình Publish/Subscribe, giúp giảm thiểu lưu lượng mạng và tối ưu hóa hiệu suất truyền thông. Publisher chỉ cần gửi tin nhắn một lần đến Broker, sau đó Broker sẽ phân phối tin nhắn đến tất cả Subscriber quan tâm, thay vì gửi riêng cho từng Subscriber.
- **Kích thước gói tin MQTT nhỏ gọn:** Giúp tiết kiệm băng thông, đặc biệt phù hợp với các thiết bị có nguồn lực hạn chế hoặc mạng có băng thông thấp.
- **Khả năng mở rộng:** MQTT Topic có khả năng mở rộng cao, có thể hỗ trợ một lượng lớn thiết bị và kết nối trong hệ thống IoT. Cấu trúc phân cấp của Topic giúp dễ dàng quản lý và tổ chức các thiết bị, cho phép thêm hoặc gỡ bỏ thiết bị mà không ảnh hưởng đến toàn hệ thống. Broker có thể được phân tán để xử lý lượng truy cập lớn và đảm bảo tính sẵn sàng cao.
- **Khả năng tùy biến linh hoạt:** Tùy chỉnh cao cho Publisher và Subscriber lựa chọn Topic phù hợp, Wildcard và cấu trúc phân cấp giúp lọc, truy cập dữ liệu cụ thể, hỗ trợ nhiều mức Chất lượng Dịch vụ (QoS) đảm bảo độ tin cậy cho từng ứng dụng.

- **Khả năng bảo mật chặt chẽ:** Hỗ trợ SSL/TLS, xác thực và ủy quyền để bảo vệ dữ liệu, ngăn chặn truy cập trái phép. Broker cấu hình cho phép Publisher/Subscriber ủy quyền truy cập Topic cụ thể.
- **Dễ dàng sử dụng:** MQTT Topic sử dụng giao thức đơn giản và dễ hiểu, giúp việc triển khai và sử dụng trở nên dễ dàng, sở hữu nhiều thư viện và công cụ mã nguồn mở hỗ trợ MQTT cho nhiều ngôn ngữ lập trình khác nhau.

Nhược điểm của MQTT Topic là:

- **Thiếu tính an toàn trong việc truyền tải tin nhắn:** Giao thức không mã hóa tin nhắn dẫn đến việc bắt buộc sử dụng SSL/TLS hoặc phương thức bảo mật khác khi truyền tải tin nhắn trên mạng trở nên không an toàn.
- **Khó khăn trong việc theo dõi:** Cấu trúc phân cấp và wildcard khiến việc theo dõi, quản lý Topic trở nên phức tạp trong hệ thống lớn, cần có công cụ và quy trình quản lý Topic phù hợp để đảm bảo nhất quán, tránh sự trùng lặp.
- **Rủi ro mất tin nhắn:** Tin cậy phụ thuộc vào mức QoS lựa chọn, khi QoS cao sẽ đảm bảo tỷ lệ giao hàng cao nhưng tốn tài nguyên sử dụng, mất kết nối mạng/lỗi hệ thống, tin nhắn có thể bị mất nếu không sử dụng QoS phù hợp.
- **Tính phụ thuộc vào Broker:** Hệ thống phải hoàn toàn phụ thuộc vào Broker để truyền tải nguồn dữ liệu, một khi Broker gặp sự cố thì toàn bộ hệ thống bị ảnh hưởng, cần có chiến lược dự phòng, khả năng phục hồi cao cho Broker.

- **Hạn chế về khả năng tính toán:** Giao thức nhẹ nhưng xử lý Topic lại trở nên phức tạp, việc xử lý nhiều cấp con và wildcard có thể tốn tài nguyên tính toán cho Broker, đặc biệt trong môi trường hệ thống lớn.

#### 4.4. MQTT Payload:

MQTT Payload là phần dữ liệu thực sự được truyền trong một tin nhắn MQTT. Đây là phần nội dung của tin nhắn mà một client (thiết bị gửi) gửi đến một broker (máy chủ trung gian) và từ broker, tin nhắn này có thể được phân phối đến các client khác đã đăng ký chủ đề (topic) tương ứng.

##### a/ Tính năng và chức năng chính của MQTT Payload:

**Truyền Dữ Liệu Cảm Biến:** Trong các hệ thống IoT, payload thường chứa dữ liệu từ các cảm biến như nhiệt độ, độ ẩm, áp suất, v.v. Ví dụ, một cảm biến nhiệt độ có thể gửi một payload chứa giá trị nhiệt độ hiện tại.

**Điều Khiển Thiết Bị:** Payload cũng có thể chứa các lệnh điều khiển để điều khiển các thiết bị từ xa. Ví dụ, một payload có thể yêu cầu bật hoặc tắt một thiết bị điện.

**Truyền Thông Tin Trạng Thái:** Payload có thể chứa thông tin về trạng thái hiện tại của một thiết bị hoặc hệ thống. Ví dụ, một thiết bị có thể gửi payload chứa trạng thái pin hoặc trạng thái kết nối mạng.

**Truyền Thông Tin Cảnh Báo và Báo Cáo:** Payload có thể chứa thông tin cảnh báo khi có sự cố xảy ra, như phát hiện khói hoặc nhiệt độ quá cao, hoặc các báo cáo định kỳ về hoạt động của hệ thống.

**Đồng Bộ Hóa Dữ Liệu:** Payload có thể được sử dụng để đồng bộ hóa dữ liệu giữa các thiết bị và hệ thống, đảm bảo tất cả các thành phần của hệ thống đều có thông tin cập nhật nhất.

**Giao Tiếp giữa Các Ứng Dụng:** Trong các ứng dụng phân tán, payload giúp các thành phần của hệ thống giao tiếp với nhau bằng cách truyền tải thông điệp và dữ liệu cần thiết.

#### 4.5. Quality of Service (QoS) trong MQTT:

QoS (Quality of Service) trong MQTT là một thuộc tính quan trọng xác định mức độ đảm bảo của quá trình truyền tin nhắn giữa các client và broker. QoS giúp kiểm soát độ tin cậy và đảm bảo rằng tin nhắn sẽ được gửi đi và nhận đúng cách, tùy thuộc vào mức độ quan trọng của dữ liệu và tình trạng mạng.

Có ba mức QoS trong MQTT, mỗi mức cung cấp một mức độ đảm bảo khác nhau:

- **QoS 0: At most once (Gửi tối đa một lần).** Đây là mức độ đảm bảo khi tin nhắn được gửi từ client đến broker một lần và không có xác nhận (acknowledgment) từ broker. Nếu có lỗi trong quá trình truyền (ví dụ như mất kết nối), tin nhắn sẽ không được gửi lại. Đây là mức QoS có độ tin cậy thấp nhất nhưng có độ trễ thấp nhất và tiết kiệm băng thông nhất. QoS 0 thích hợp cho dữ liệu không quan trọng hoặc các ứng dụng không yêu cầu độ tin cậy cao, ví dụ như dữ liệu cảm biến môi trường gửi liên tục.
- **QoS 1: At least once (Gửi ít nhất một lần).** Đây là mức độ đảm bảo khi tin nhắn được gửi từ client đến broker và phải nhận được xác nhận (PUBACK) từ broker. Nếu không nhận được xác nhận, tin nhắn sẽ được gửi lại cho đến khi nhận được. Điều này đảm bảo rằng tin nhắn sẽ được nhận ít nhất một lần, nhưng có thể xảy ra trường hợp nhận trùng lặp. QoS 1 phù hợp cho các

ứng dụng cần độ tin cậy cao hơn, nhưng có thể xử lý tin nhắn trùng lặp, như việc điều khiển thiết bị từ xa hoặc cập nhật trạng thái quan trọng.

- **QoS 2: Exactly once (Gửi chính xác một lần).** Đây là mức QoS cao nhất, đảm bảo rằng tin nhắn sẽ được nhận đúng một lần và chỉ một lần. Quá trình truyền tải gồm hai giai đoạn xác nhận giữa client và broker để loại bỏ khả năng trùng lặp tin nhắn. Giai đoạn này bao gồm việc trao đổi các gói tin PUBREC, PUBREL, và PUBCOM. QoS 2 dùng cho các ứng dụng mà việc trùng lặp tin nhắn không được phép xảy ra, như giao dịch tài chính, hoặc các hệ thống yêu cầu dữ liệu chính xác và duy nhất.

#### **a/ Tính năng và chức năng chính của QoS trong MQTT:**

**Đảm bảo Độ Tin Cậy của Tin Nhắn:** QoS 0 sẽ đảm bảo tin nhắn được gửi đi một lần, không yêu cầu xác nhận, phù hợp với các trường hợp dữ liệu không quan trọng hoặc chấp nhận được nếu bị mất. QoS 1 sẽ đảm bảo tin nhắn được gửi ít nhất một lần, yêu cầu xác nhận từ broker, nếu không nhận được xác nhận, client sẽ gửi lại tin nhắn. QoS 2 sẽ đảm bảo tin nhắn được gửi chính xác một lần, sử dụng quá trình trao đổi nhiều bước để loại bỏ khả năng trùng lặp.

**Điều Khiển Tần Suất và Mức Độ Xác Nhận:** Cho phép các nhà phát triển chọn mức độ xác nhận phù hợp với yêu cầu của ứng dụng, từ đó điều chỉnh độ tin cậy và hiệu suất của hệ thống.

**Giảm Thiểu Sự Trùng Lặp Tin Nhắn:** QoS 2 sử dụng một quy trình xác nhận phức tạp để đảm bảo rằng tin nhắn không bị trùng lặp, rất quan trọng đối với các ứng dụng đòi hỏi tính chính xác cao.

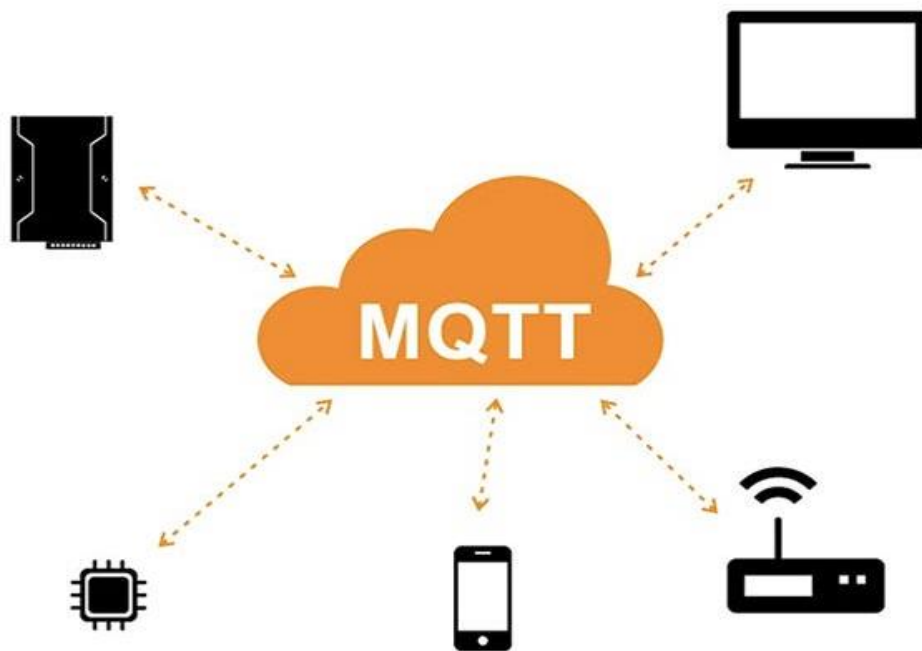
**Tăng Cường Tính Tin Cậy trong Môi Trường Mạng Không Ổn Định:** Cung cấp các mức độ dịch vụ khác nhau giúp hệ thống xử lý các sự cố mạng, chẳng hạn

như mất gói tin hoặc mất kết nối tạm thời, mà không ảnh hưởng đến việc truyền tải dữ liệu quan trọng.

### III. Dịch vụ CloudMQTT:

#### 1. Khái niệm:

CloudMQTT là một dịch vụ cung cấp các phiên bản MQTT (Message Queuing Telemetry Transport) trên nền tảng đám mây. MQTT là một giao thức nhắn tin nhẹ, được thiết kế cho các kết nối từ xa thường có độ trễ cao hoặc không ổn định. Nó rất phù hợp cho các ứng dụng Internet of Things (IoT) do tính hiệu quả và tiêu thụ băng thông thấp.



*Hình 11. Dịch vụ CloudMQTT*

Cloud MQTT là một giải pháp lý tưởng cho các hệ thống IoT, cung cấp môi trường linh hoạt, an toàn và dễ quản lý để kết nối và truyền tải dữ liệu giữa các thiết bị và ứng dụng. Việc sử dụng dịch vụ MQTT trên nền tảng đám mây giúp các nhà phát triển và doanh nghiệp tập trung vào việc phát triển ứng dụng và dịch vụ mà không phải lo lắng về hạ tầng và quản lý broker MQTT.



## **2. Tính năng và chức năng chính của CloudMQTT:**

**Truyền Thông Tin Nhắn (Message Communication):** Cho phép các thiết bị IoT và ứng dụng truyền tải dữ liệu với độ tin cậy cao và độ trễ thấp. Các tin nhắn có thể bao gồm dữ liệu cảm biến, lệnh điều khiển, thông tin trạng thái, và nhiều loại dữ liệu khác.

**Quản Lý Kết Nối (Connection Management):** Hỗ trợ quản lý hàng nghìn đến hàng triệu kết nối từ các thiết bị khác nhau, đảm bảo rằng tất cả các thiết bị đều có thể kết nối và truyền dữ liệu hiệu quả.

**Xử Lý Sự Kiện (Event Handling):** Cho phép cấu hình và xử lý các sự kiện dựa trên dữ liệu nhận được, chẳng hạn như kích hoạt các hành động cụ thể khi nhận được một loại tin nhắn nào đó.

**Lưu Trữ và Phân Tích Dữ Liệu (Data Storage and Analysis):** Tích hợp với các dịch vụ lưu trữ và phân tích dữ liệu, cho phép lưu trữ lịch sử dữ liệu, phân tích dữ liệu thời gian thực, và tạo báo cáo chi tiết.

**Đảm Bảo Chất Lượng Dịch Vụ (Quality of Service - QoS):** Hỗ trợ các mức QoS khác nhau để đảm bảo rằng tin nhắn được truyền tải với mức độ tin cậy phù hợp, từ QoS 0 (ít nhất một lần) đến QoS 2 (chính xác một lần).

**Khả Năng Giám Sát và Báo Cáo (Monitoring and Reporting):** Cung cấp các công cụ giám sát hiệu suất, theo dõi trạng thái hệ thống, và tạo báo cáo chi tiết về hoạt động và hiệu suất của dịch vụ MQTT.

### 3. Ưu điểm và nhược điểm của dịch vụ CloudMQTT

Ưu điểm của dịch vụ CloudMQTT là:

**Dễ dàng triển khai và quản lý:** Cloud MQTT loại bỏ nhu cầu phải triển khai và quản lý broker MQTT trên các máy chủ vật lý hoặc máy chủ ảo của riêng mình. Người dùng có thể dễ dàng sử dụng dịch vụ MQTT được cung cấp sẵn trên các nền tảng đám mây như AWS, Azure, GCP, giúp tiết kiệm thời gian và chi phí cho việc quản lý hạ tầng.

**Khả năng mở rộng (Scalability):** Dịch vụ Cloud MQTT cho phép mở rộng linh hoạt để đáp ứng nhu cầu ngày càng tăng của các thiết bị IoT và lưu lượng dữ liệu. Hạ tầng đám mây có thể tự động mở rộng để xử lý hàng triệu kết nối và tin nhắn một cách hiệu quả.

**Tính sẵn sàng cao (High Availability):** Các nhà cung cấp dịch vụ Cloud MQTT thường cung cấp các cơ chế dự phòng và khôi phục dữ liệu để đảm bảo rằng dịch vụ luôn hoạt động ổn định và không bị gián đoạn, phù hợp với các ứng dụng yêu cầu tính sẵn sàng cao.

**Bảo mật cao:** Dịch vụ Cloud MQTT thường đi kèm với các tính năng bảo mật mạnh mẽ như mã hóa dữ liệu (TLS/SSL), xác thực người dùng và kiểm soát truy cập. Điều này giúp bảo vệ dữ liệu truyền tải và ngăn ngừa các mối đe dọa bảo mật từ bên ngoài.

**Quản lý kết nối (Connection Management):** Cloud MQTT cung cấp các công cụ và giao diện giám sát cho phép người dùng quản lý hàng nghìn đến hàng triệu kết nối từ các thiết bị IoT. Điều này bao gồm khả năng theo dõi trạng thái kết nối, phân tích dữ liệu về sự kiện và hiệu suất, giúp người dùng dễ dàng giám sát và điều chỉnh hệ thống.

**Hỗ trợ nhiều mức độ QoS (Quality of Service):** Cloud MQTT hỗ trợ các mức độ QoS khác nhau (QoS 0, QoS 1, QoS 2), cho phép người dùng lựa chọn mức độ tin cậy phù hợp với yêu cầu của ứng dụng IoT. Điều này đảm bảo rằng dữ liệu được truyền tải một cách chính xác và đáng tin cậy.

**Tích hợp dễ dàng:** Cloud MQTT có khả năng tích hợp dễ dàng với các dịch vụ khác trên nền tảng đám mây như cơ sở dữ liệu, phân tích dữ liệu, AI/ML, và các dịch vụ lưu trữ khác. Điều này giúp tạo ra các giải pháp IoT toàn diện và mở rộng.

**Tiết kiệm chi phí và tối ưu hóa hiệu suất:** Sử dụng Cloud MQTT giúp người dùng tiết kiệm chi phí cho việc triển khai và duy trì hạ tầng MQTT, đồng thời tối ưu hóa hiệu suất hoạt động của hệ thống IoT bằng các tính năng như cân bằng tải và tự động mở rộng.

Nhược điểm của CloudMQTT là:

**Phụ thuộc vào kết nối internet:** Cloud MQTT yêu cầu kết nối internet ổn định để hoạt động hiệu quả. Nếu có sự cố với kết nối internet, các thiết bị IoT sẽ không thể truyền tải dữ liệu hoặc không đảm bảo được mức độ QoS mong muốn.

**Chi phí sử dụng:** Mặc dù sử dụng Cloud MQTT có thể giảm thiểu chi phí cho việc triển khai và duy trì hạ tầng MQTT, nhưng các dịch vụ đám mây không phải lúc nào cũng rẻ. Người dùng cần tính toán kỹ lưỡng chi phí sử dụng dịch vụ Cloud MQTT, đặc biệt khi có số lượng lớn các thiết bị IoT hoặc yêu cầu lưu lượng truy cập cao.

**Bảo mật dữ liệu:** Mặc dù các nhà cung cấp dịch vụ Cloud MQTT thường cung cấp các tính năng bảo mật như mã hóa dữ liệu và xác thực người dùng, tuy nhiên việc truyền tải dữ liệu qua internet vẫn có nguy cơ bị tấn công và đánh cắp thông tin. Do đó, cần có các biện pháp bảo mật phù hợp để bảo vệ dữ liệu.

**Khả năng kiểm soát và quản lý:** Việc sử dụng dịch vụ Cloud MQTT có thể làm giảm sự kiểm soát trực tiếp và quản lý của người dùng đối với hạ tầng MQTT. Người dùng phải phụ thuộc vào các công cụ và giao diện quản trị được cung cấp bởi nhà cung cấp dịch vụ để giám sát và điều chỉnh hệ thống.

**Hiệu suất và độ trễ:** Mặc dù dịch vụ Cloud MQTT thường được thiết kế để đảm bảo hiệu suất cao và độ trễ thấp, nhưng việc sử dụng qua internet vẫn có thể gây ra một số độ trễ so với việc triển khai MQTT trên cơ sở hạ tầng vật lý. Điều này có thể ảnh hưởng đến tính chính xác và thời gian phản hồi của hệ thống IoT.

**Rủi ro về sự cố hạ tầng đám mây:** Mặc dù các nhà cung cấp dịch vụ đám mây cố gắng cung cấp tính sẵn sàng cao, nhưng vẫn có thể xảy ra sự cố hạ tầng hoặc gián đoạn dịch vụ. Khi đó, hệ thống IoT sẽ không thể truyền tải dữ liệu hoặc có thể gặp khó khăn trong việc kết nối và điều khiển các thiết bị.

**Tuân thủ quy định và chính sách của nhà cung cấp:** Việc sử dụng dịch vụ Cloud MQTT có thể đặt ra các vấn đề về tuân thủ quy định và chính sách của nhà cung cấp đám mây. Người dùng cần phải hiểu rõ các điều khoản dịch vụ và chính sách bảo mật của nhà cung cấp để đảm bảo rằng dữ liệu của họ được bảo vệ và tuân thủ các quy định pháp lý.

**Hạn chế về tùy chỉnh và linh hoạt:** Một số dịch vụ Cloud MQTT có thể hạn chế trong việc tùy chỉnh và linh hoạt so với việc triển khai MQTT trên các máy chủ riêng. Điều này có thể ảnh hưởng đến khả năng đáp ứng các yêu cầu đặc biệt và tùy chỉnh của từng ứng dụng IoT cụ thể.

#### IV. So sánh MQTT và các công nghệ tương đương:

Đặc tính	MQTT	HTTP	CoAP
Năm	1999	1997	2010
Hình thức	Truyền tải bản tin từ xa	Giao thức truyền siêu văn bản	Giao thức ứng dụng ràng buộc
Kiến trúc	Client/Broker	Client/Server	Client/Server
Mô hình	Xuất bản/Đăng ký	Yêu cầu/phản hồi	Yêu cầu/phản hồi
Kích thước Header	2 Byte	Undefined	4 Byte
Kích thước bản tin	Nhỏ và không xác định (tối đa 256MB)	Lớn và không xác định (phụ thuộc vào Web Server)	Nhỏ và không xác định (thường nhỏ để phù hợp với IP datagram)
Phương pháp	Connect, Disconnect, Publish, Subscribe, Unsubscribe, Close	Get, Post, Head, Put, Patch, Options, Connect, Delete	Get, Post, Put, Delete
Hỗ trợ bộ nhớ cache và proxy	Một phần	Có	Có

<b>Chất lượng (QoS)/Độ tin cậy</b>	QoS 0 – Truyền tối đa 1 lần QoS 1 – Truyền tối thiểu 1 lần QoS 2 – Truyền duy nhất 1 lần	Giới hạn (thông qua TCP)	Bản tin có thể xác nhận hoặc không thể xác nhận
<b>Giao thức truyền</b>	TCP	TCP	UDP, (Stream Control Transmission Protocol)
<b>Bảo vệ</b>	TLS/SSL	TLS/SSL	DTLS, IPsec
<b>Cổng mặc định</b>	1883/8883 (TLS/SSL)	80/443 (TLS/SSL)	5683 (UDP Port)
<b>Định dạng mã hóa</b>	Nhị phân	Văn bản	Nhị phân
<b>Cấp phép</b>	Nguồn mở	Miễn phí	Nguồn mở
<b>Hỗ trợ tổ chức</b>	IBM, Facebook, Eurotech, Cisco, Red Hat, Software AG, Tibco, ITSO, M2Mi, Amazon Web	Tiêu chuẩn giao thức web toàn cầu	Hỗ trợ cộng đồng web rộng lớn, Cisco, Contiki, Erika, IoTivity

	Services (AWS), InduSoft, Fiorano		
--	--------------------------------------	--	--

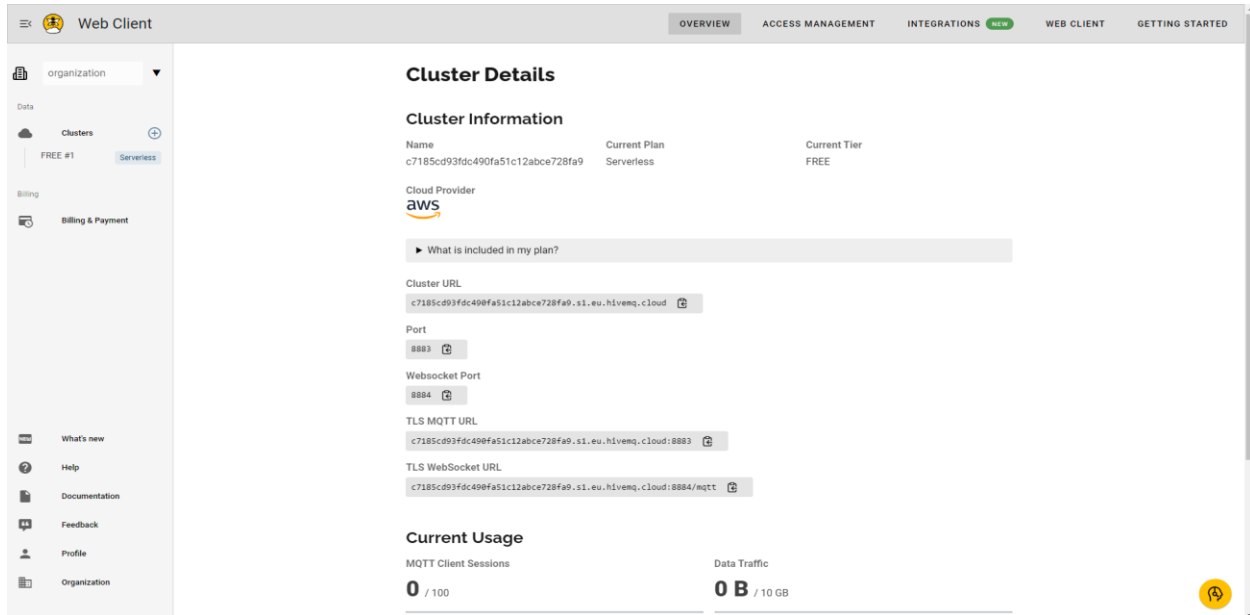
## **V. Tổng kết**

CloudMQTT triển khai một hệ thống bảo mật nhiều lớp bao gồm chứng thực người dùng, mã hóa dữ liệu, quản lý quyền truy cập chi tiết, giám sát và ghi nhật ký, và các biện pháp phòng ngừa tấn công. Những biện pháp này đảm bảo rằng dữ liệu và hệ thống MQTT của bạn luôn được bảo vệ tốt nhất, cung cấp một môi trường an toàn và đáng tin cậy cho các ứng dụng IoT của người dùng.



## Chương II: TRIỂN KHAI

### I. Triển khai dịch vụ:



Hình 12. Giao diện của HiveMQ sau khi đăng nhập và triển khai Broker

Tạo tài khoản và phân quyền cho người dùng

### Access Management

#### Credentials

Define one or more sets of credentials that allow MQTT clients to connect to your HiveMQ Cloud cluster. To learn more [check out our Security Fundamentals guide](#).

Username \*

User1

At least 5 characters

Password \*

@Admin1234

At least 8 characters, 1 digit, 1 uppercase character

Confirm Password \*

.....

Passwords must match

Permission \*

Publish Only

Add permissions to limit access

> CREATE CREDENTIAL

Hình 13. Giao diện đăng ký và phân quyền người dùng

Điền các thông tin và tài khoản người dùng do Admin cấp để kết nối tới web client

The screenshot shows the 'Connection' tab of the HiveMQ Cloud interface. At the top, there is a banner for 'HIVEMQ CLOUD' with the text 'Need a fully managed MQTT broker? Get your own Cloud broker and connect up to 100 devices for free.' and a yellow button 'Get your free account'. Below the banner, the 'Connection' section is titled 'Connection' and shows a status of 'disconnected' with a red dot. The interface includes several input fields: 'Host' (b162fce66c8843ce98266d6be7589ca3.s1.eu.hivemq.cl), 'Port' (8884), 'ClientID' (clientId-ROLxpG7IKg), 'Username' (User1), 'Password' (masked with dots), 'Keep Alive' (60), 'SSL' (checkbox), 'Clean Session' (checkbox), 'Last-Will Topic' (empty), 'Last-Will QoS' (0), 'Last-Will Retain' (checkbox), and 'Last-Will Message' (empty text area). A blue 'Connect' button is located next to the 'ClientID' field.

Hình 14. Giao diện chính của Websocket client

Sau khi kết nối thành công thì gửi topic đến web client

The screenshot shows the 'Publish' and 'Subscriptions' sections of the HiveMQ Cloud interface. The 'Publish' section is active and shows a 'Topic' field with the value 'CONTROL', a 'QoS' dropdown set to '2', and a 'Retain' checkbox. A blue 'Publish' button is next to the 'Topic' field. Below the 'Topic' field is a 'Message' text area containing the text 'redon'. The 'Subscriptions' section is also visible, showing a blue button 'Add New Topic Subscription'. Below the 'Publish' section is a 'Messages' section, which is currently empty.

Hình 15. Giao diện tương tác topic

Bên Web Client sẽ nhận được lệnh và gửi tới thiết bị giả lập

Messages 7

Send and see messages that are published to the topics you are subscribed to. If you cannot see any messages, make sure you are subscribed to the correct topics. You can always subscribe to the (#) wildcard to receive all messages.

MESSAGE	TOPIC	QOS	ACTIONS
<div>Your message</div>	<div>Topic</div>	<div>QoS: 0</div>	<div>Send Message</div>
Hello Web Client	<div>HELLO</div>	2	<div></div>
redon	<div>CONTROL</div>	2	<div></div>
redon	<div>CONTROL</div>	2	<div></div>
redon	<div>CONTROL</div>	2	<div></div>
redon	<div>CONTROL</div>	2	<div></div>
redon	<div>CONTROL</div>	2	<div></div>
redon	<div>CONTROL</div>	2	<div></div>

Con

Hình 16. Giao diện tương tác của admin

Và đây là kết quả bên thiết bị giả lập

00:59.278 99%

ESP32

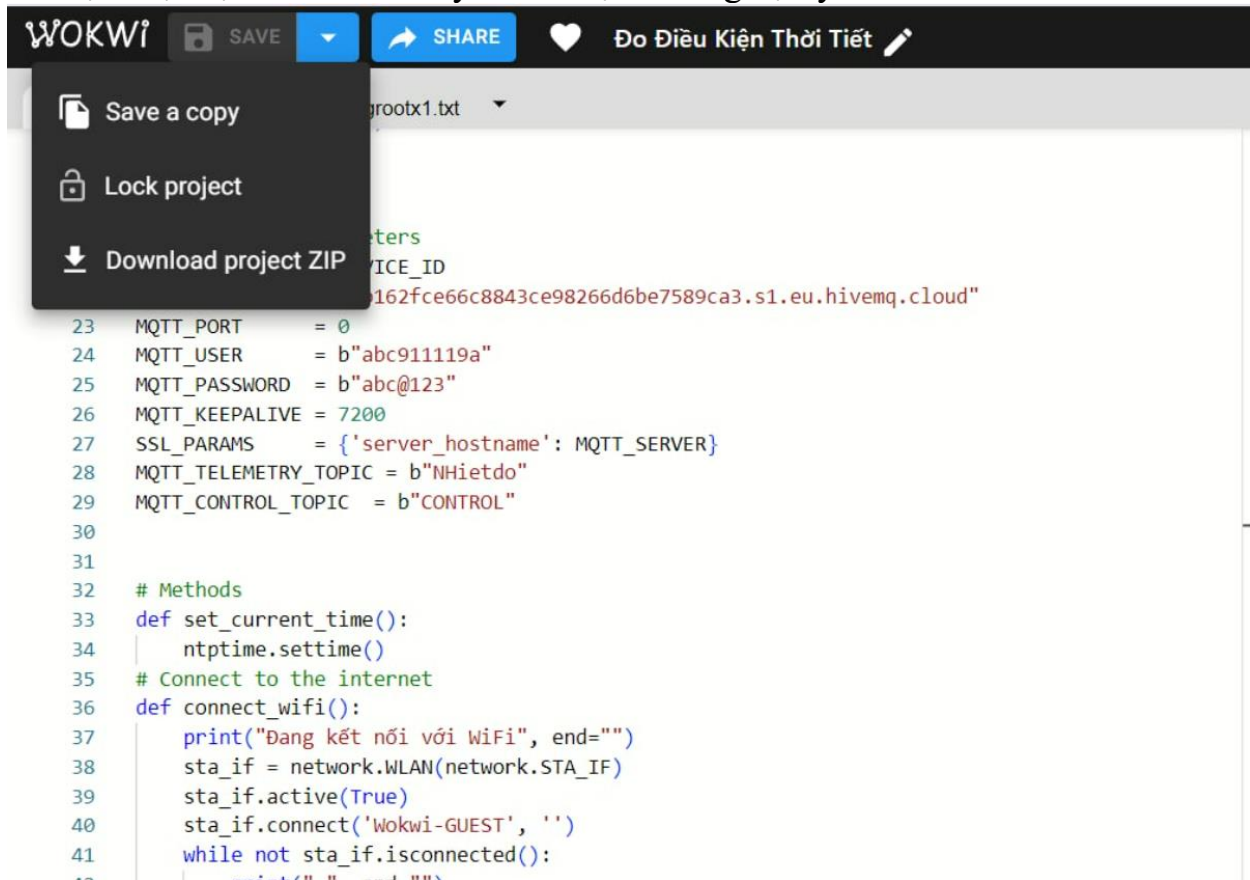
DHT22

Dữ liệu đo được:  
{ "Ngày giờ": [2024, 7, 16, 16, 15, 14, 1, 198], "Địa điểm": "TPHCM", "device ID": "HP", "Nhiệt độ": 31.4, "Độ ẩm": 50.0 }  
.Đang đo dữ liệu...  
Dữ liệu đo được:  
{ "Ngày giờ": [2024, 7, 16, 16, 15, 19, 1, 198], "Địa điểm": "TPHCM", "device ID": "HP", "Nhiệt độ": 29.4, "Độ ẩm": 49.0 }  
/nData received: topic = b'CONTROL', message = b'redon'  
Dữ liệu đo được:  
{ "red\_led": "ON", "blue\_led": "OFF" }

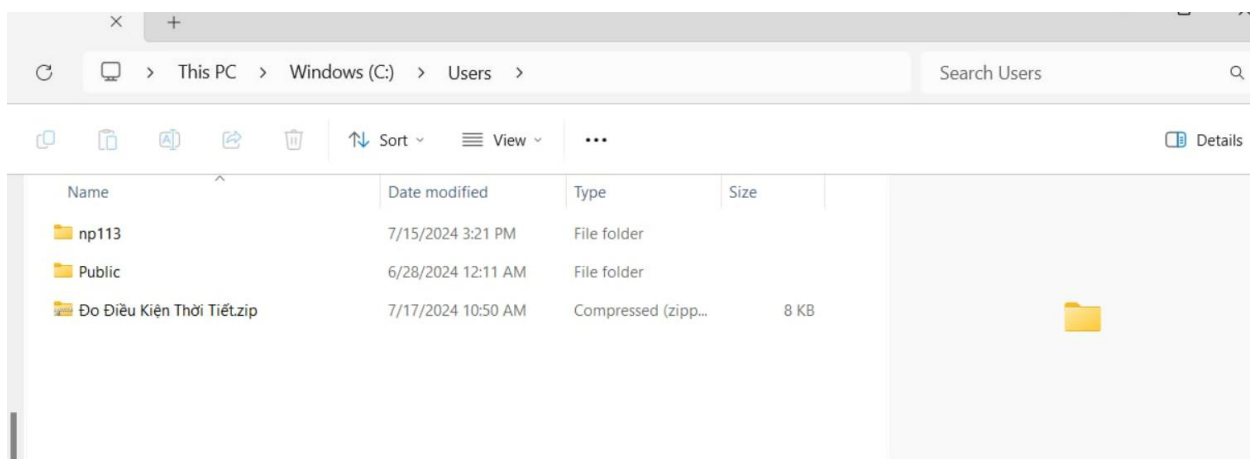
Hình 17. Thiết bị giả lập đã nhận lệnh

## II. Backup dữ liệu:

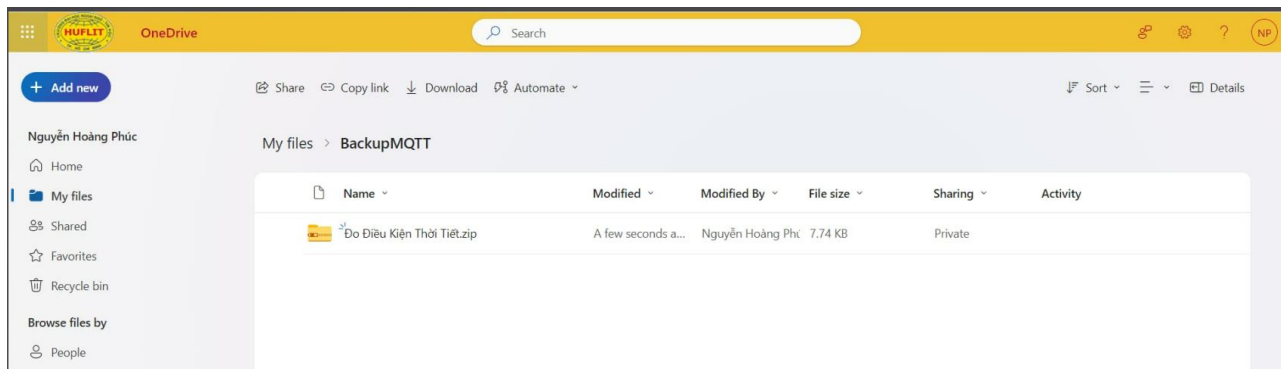
Backup cấu hình của file bằng việc tải toàn bộ file project ZIP. Sau đó lưu trữ bằng các dịch vụ điện toán đám mây khác hoặc ổ cứng vật lý



Hình 18. Tải file project ZIP của cấu hình thiết bị



Hình 19. Lưu trữ trong máy tính



Hình 20. Ví dụ về lưu trữ trên dịch vụ điện toán đám mây

### III. Bảo mật:

Sử dụng chứng thực SSL

SSL (Secure Sockets Layer) là một giao thức bảo mật được sử dụng để thiết lập một kết nối mã hóa giữa máy khách (như trình duyệt web) và máy chủ. Mục đích của SSL là đảm bảo rằng tất cả dữ liệu trao đổi giữa máy khách và máy chủ đều được bảo mật và không bị truy cập bởi bên thứ ba.

```

5
6  from machine import Pin
7  import dht
8  from umqtt.simple import MQTTClient
9  import ujson
0  import network, time, ntptime, random
1  import ussl as ssl
2

```

Hình 21. Import các thư viện

```

def connect_mqtt():
    print("Kết nối với MQTT server... ", end="")
    # MQTTClient(client_id, server, port=0, user=None, password=None, keepalive=0, ssl=False, ssl_params={})
    mqtt_client = MQTTClient(
        MQTT_CLIENT_ID,
        MQTT_SERVER,
        port=MQTT_PORT,
        user=MQTT_USER,
        password=MQTT_PASSWORD,
        keepalive=MQTT_KEEPALIVE,
        ssl=True,
        ssl_params=SSL_PARAMS)
    mqtt_client.set_callback(on_message)
    mqtt_client.connect()
    print("MQTT server kết nối thành công!")
    mqtt_client.subscribe(MQTT_CONTROL_TOPIC)
    return mqtt_client

```

Hình 22. Kết nối máy chủ. Sử dụng SSL để mã hóa đầu cuối

```
# MQTT Server Parameters
MQTT_CLIENT_ID = DEVICE_ID
MQTT_SERVER     = b"b162fce66c8843ce98266d6be7589ca3.s1.eu.hivemq.cloud"
MQTT_PORT       = 0
MQTT_USER       = b"abc911119a"
MQTT_PASSWORD   = b"abc@123"
MQTT_KEEPALIVE  = 7200
SSL_PARAMS      = {'server_hostname': MQTT_SERVER}
MQTT_TELEMETRY_TOPIC = b"NHietdo"
MQTT_CONTROL_TOPIC  = b"CONTROL"
```

*Hình 23. Cấu hình kết nối MQTT, cấu hình SSL và chủ đề MQTT mà client giao tiếp*

# CHƯƠNG III: KẾT LUẬN

## I. Đánh giá kết quả đạt được:

### 1. Kết quả đạt được:

Trong bối cảnh phát triển công nghệ IoT, giao thức MQTT đã chứng tỏ vai trò quan trọng của mình trong việc cung cấp một phương thức giao tiếp hiệu quả, nhẹ nhàng và tin cậy giữa các thiết bị IoT. Đồ án này đã đi sâu vào việc sử dụng MQTT để điều khiển thiết bị IoT giả lập, đạt được những kết quả quan trọng sau:

- Hiểu biết sâu về giao thức MQTT: Đồ án đã cung cấp một cái nhìn tổng quan về MQTT, từ nguyên lý hoạt động đến các thành phần chính như broker, client, các chủ đề (topics), và thông điệp (messages).
- Thiết lập và cấu hình môi trường: Việc triển khai HiveMQ broker và cấu hình bảo mật cao đã giúp đảm bảo sự an toàn trong giao tiếp giữa các thiết bị.
- Thực hiện điều khiển thiết bị IoT giả lập: Sử dụng Wokwi để giả lập các thiết bị IoT như ESP32 và điều khiển LED thông qua MQTT WebSocket. Thiết lập kênh giao tiếp giữa ESP32 và HiveMQ Cloud, giúp minh chứng khả năng điều khiển thiết bị từ xa qua internet.

### 2. Những hạn chế của đồ án:

Việc triển khai một dự án MQTT để điều khiển thiết bị IoT giả lập có nhiều lợi ích, nhưng cũng tồn tại một số hạn chế cần xem xét. Dưới đây là một số hạn chế phổ biến:

- Hiệu suất và Độ trễ: Khi sử dụng môi trường giả lập, hiệu suất thực tế và độ trễ có thể không phản ánh chính xác môi trường thực tế. Các thiết bị vật lý có thể có các vấn đề khác như sự cố mạng hoặc hạn chế phần cứng mà giả lập không thể tái hiện.
- Khả năng mở rộng: Môi trường giả lập có thể không xử lý được số lượng lớn thiết bị cùng một lúc như môi trường thực tế. Điều này có thể ảnh hưởng đến việc kiểm tra khả năng mở rộng và hiệu suất của hệ thống.
- Môi trường mạng không thực tế: Mạng giả lập không thể mô phỏng chính xác các vấn đề mạng như nhiễu sóng, mất gói tin, hoặc các điều kiện mạng không ổn định mà các thiết bị IoT thực tế có thể gặp phải.
- Phụ thuộc vào công cụ giả lập: Kết quả và độ chính xác của môi trường giả lập phụ thuộc rất nhiều vào công cụ và nền tảng giả lập được sử dụng. Các công cụ khác nhau có thể có khả năng và hạn chế khác nhau.

## **II. Kết luận:**

### **1. Kết luận:**

Hệ thống IoT giả lập sử dụng giao thức MQTT đã hoạt động ổn định và hiệu quả, cho phép điều khiển thiết bị ESP32 một cách chính xác và nhanh chóng. Các yêu cầu về bảo mật cũng được đảm bảo, giúp hệ thống an toàn trước các nguy cơ tấn công. Đồ án này mở ra nhiều cơ hội ứng dụng trong thực tế, từ việc điều khiển thiết bị nhà thông minh đến các hệ thống giám sát và điều khiển công nghiệp.

### **2. Hướng phát triển trong tương lai:**

- **Mở Rộng Thiết Bị:** Tích hợp thêm nhiều loại thiết bị IoT khác nhau vào hệ thống.
- **Nâng Cao Bảo Mật:** Áp dụng các phương pháp bảo mật nâng cao hơn như xác thực hai yếu tố, quản lý truy cập linh hoạt.
- **Tối Ưu Hóa Hiệu Suất:** Cải thiện hiệu suất của hệ thống để đảm bảo hoạt động mượt mà trong các môi trường mạng khác nhau.



## TÀI LIỆU THAM KHẢO

<https://www.google.com.vn/>

<https://www.youtube.com/>

<https://github.com/>