[**Đề xuất Dự án: Nền tảng Q&A và Polling Real-time với AWS**](#) **1**

[Tối ưu hóa Tương tác và Đo lường Hiệu quả Sự kiện Trực tuyến bằng Kiến trúc Serverless](#) 2

[Tóm tắt Điều hành (Executive Summary)](#) 2

[1. Phát biểu Bài toán (Problem Statement)](#) 3

[1.1. Bối cảnh Hiện tại](#) 3

[1.2. Thách thức Cốt lõi](#) 3

[1.3. Tác động đến các Bên liên quan](#) 3

[1.4. Hậu quả Kinh doanh](#) 4

[2. Kiến trúc Giải pháp (Solution Architecture)](#) 4

[2.1. Tổng quan Kiến trúc](#) 4

[2.2. Lựa chọn Dịch vụ AWS và Lý do](#) 6

[2.3. Thiết kế Thành phần](#) 7

[2.4. Kiến trúc Bảo mật (Security Architecture)](#) 7

[2.5. Thiết kế Mở rộng & Hiệu năng](#) 8

[3. Kế hoạch Triển khai Kỹ thuật (Technical Implementation)](#) 8

[3.1. Các Giai đoạn Triển khai](#) 8

[3.2. Phương pháp Phát triển](#) 9

[3.3. Chiến lược Kiểm thử](#) 9

[4. Tiến độ & Cột mốc (Timeline & Milestones)](#) 10

[5. Ước tính Ngân sách (Budget Estimation)](#) 11

[5.1. Giả định về Lưu lượng](#) 11

[5.2. Bảng kê Chi phí Ước tính (AWS Region: ap-southeast-1)](#) 11

[5.3. Phân tích Lợi tức Đầu tư (ROI)](#) 12

[6. Đánh giá Rủi ro (Risk Assessment)](#) 12

[7. Kết quả Mong đợi (Expected Outcomes)](#) 13

[7.1. Chỉ số Thành công (Success Metrics)](#) 13

[7.2. Lợi ích Kinh doanh](#) 13

[7.3. Cải tiến Kỹ thuật](#) 14

[Phụ lục](#) 14

[A. GraphQL Schema (schema.graphql)](#) 14

[B. Chi tiết Tính toán Chi phí](#) 16

[D. Nguồn tham khảo](#) 16

# **Đề xuất Dự án: Nền tảng Q&A và Polling Real-time với AWS**

Đề tài #164: Serverless GraphQL API với AppSync và Lambda Resolvers

## **Tóm tắt Điều hành (Executive Summary)**

Trong kỷ nguyên số, các sự kiện trực tuyến (webinar, workshop, livestream) đã trở thành công cụ chiến lược không thể thiếu. Tuy nhiên, 81% các nhà tổ chức sự kiện vẫn xem việc duy trì tương tác của khán giả là thách thức lớn nhất (Nguồn: Bizzabo, 2022). Các nền tảng chat truyền thống thường tạo ra "sự hỗn loạn thông tin", độ trễ cao và không có khả năng sàng lọc, dẫn đến trải nghiệm khán giả rời rạc, giảm hiệu quả truyền tải thông điệp và cuối cùng là làm xói mòn giá trị thương hiệu.

Dự án này đề xuất xây dựng một **Nền tảng Hỏi-Đáp (Q&A) và Thăm dò ý kiến (Polling) Real-time** sử dụng kiến trúc serverless 100% trên AWS để giải quyết triệt để bài toán trên. Giải pháp không chỉ cung cấp một kênh tương tác có cấu trúc, tức thời (<300ms độ trễ) mà còn biến tương tác thành dữ liệu kinh doanh có giá trị.

**Kiến trúc giải pháp** xoay quanh **AWS AppSync** làm trung tâm quản lý GraphQL API, kết hợp với các **AWS Lambda Resolvers** để xử lý logic nghiệp vụ linh hoạt. Dữ liệu được lưu trữ trên **Amazon DynamoDB** với thiết kế Single-Table để tối ưu hiệu năng và chi phí, trong khi **Amazon Cognito** đảm bảo xác thực và phân quyền an toàn. Toàn bộ hệ thống được phân phối toàn cầu qua **Amazon CloudFront**, đảm bảo trải nghiệm nhanh chóng cho người dùng ở mọi nơi.

**Lợi ích kinh doanh và kết quả mong đợi** là rất rõ ràng và có thể đo lường:

* **Tăng tỷ lệ tương tác của khán giả lên ít nhất 40%** thông qua Q&A có sắp xếp theo lượt bình chọn (upvote) và polling trực quan.
* **Giảm độ trễ tương tác xuống dưới 300ms**, mang lại trải nghiệm real-time thực sự.
* **Giảm chi phí vận hành trên 60%** so với các giải pháp dựa trên máy chủ truyền thống nhờ mô hình "pay-as-you-go" và tối ưu hóa tài nguyên.
* **Cung cấp báo cáo phân tích sau sự kiện**, biến các câu hỏi và kết quả bình chọn thành insight giá trị về mối quan tâm của khách hàng.
* **Nâng cao hình ảnh thương hiệu** thông qua việc cung cấp một trải nghiệm sự kiện chuyên nghiệp, hiện đại và dựa trên dữ liệu.

Dự án được lên kế hoạch triển khai trong **15 ngày làm việc**, với ngân sách vận hành ước tính dưới **$15/tháng** cho quy mô 500 người dùng/sự kiện, thể hiện hiệu quả chi phí vượt trội. Tôi tin rằng giải pháp này không chỉ giải quyết một bài toán cấp thiết mà còn tạo ra lợi thế cạnh tranh bền vững và có tiềm năng phát triển thành một sản phẩm SaaS độc lập.

## **1. Phát biểu Bài toán (Problem Statement)**

### **1.1. Bối cảnh Hiện tại**

Sự bùng nổ của kinh tế số đã biến các sự kiện trực tuyến thành phương thức giao tiếp chính. Tuy nhiên, trong khi công nghệ streaming video đã phát triển vượt bậc, khía cạnh tương tác hai chiều vẫn còn ở giai đoạn sơ khai. Hầu hết các sự kiện hiện nay đều dựa vào các khung chat văn bản đơn giản, một công cụ được thiết kế cho các cuộc trò chuyện 1-1, không phải cho các cuộc thảo luận quy mô lớn, có cấu trúc.

### **1.2. Thách thức Cốt lõi**

Các kênh tương tác hiện tại đang tạo ra một "nợ trải nghiệm" (experience debt) đáng kể, thể hiện qua các vấn đề:

* **"Sự hỗn loạn thông tin" (Information Overload):** Trong các sự kiện lớn, khung chat nhanh chóng bị quá tải. Các câu hỏi quan trọng bị "trôi" đi trong biển thông tin nhiễu. Điều này không chỉ làm nản lòng người hỏi mà còn khiến diễn giả bỏ lỡ những cơ hội tương tác vàng.
* **Thiếu cơ chế sàng lọc (Lack of Prioritization):** Diễn giả không có cách nào để biết câu hỏi nào được nhiều người quan tâm nhất. Họ buộc phải đọc lướt một cách ngẫu nhiên, dẫn đến việc trả lời các câu hỏi ít liên quan trong khi bỏ qua các mối quan tâm cốt lõi của đám đông.
* **Độ trễ cao và trải nghiệm đứt gãy (High Latency & Disjointed Experience):** Các hệ thống truyền thống không được tối ưu cho tương tác tức thời, tạo ra độ trễ khó chịu. Khán giả phải làm mới trang để xem kết quả poll, phá vỡ luồng theo dõi tự nhiên của sự kiện.

### **1.3. Tác động đến các Bên liên quan**

* **Đối với Khán giả:** Trải nghiệm trở nên thụ động và đáng thất vọng. Họ cảm thấy tiếng nói của mình không được lắng nghe. Theo một khảo sát của Markletic, **49% các nhà tiếp thị cho rằng tương tác với khán giả là yếu tố lớn nhất tạo nên một sự kiện thành công**. Sự thiếu tương tác làm giảm giá trị mà khán giả nhận được.
* **Đối với Diễn giả:** Việc thiếu tương tác trực tiếp khiến diễn giả khó nắm bắt được sự tập trung và phản ứng của khán giả. Do đó, họ có thể không điều chỉnh được nội dung bài nói để phù hợp với những gì người nghe thực sự muốn biết.
* **Đối với Nhà tổ chức:** Hình ảnh thương hiệu bị ảnh hưởng tiêu cực. Một sự kiện với luồng tương tác lộn xộn để lại ấn tượng về sự thiếu chuyên nghiệp, làm giảm khả năng giữ chân khán giả cho các sự kiện tương lai.

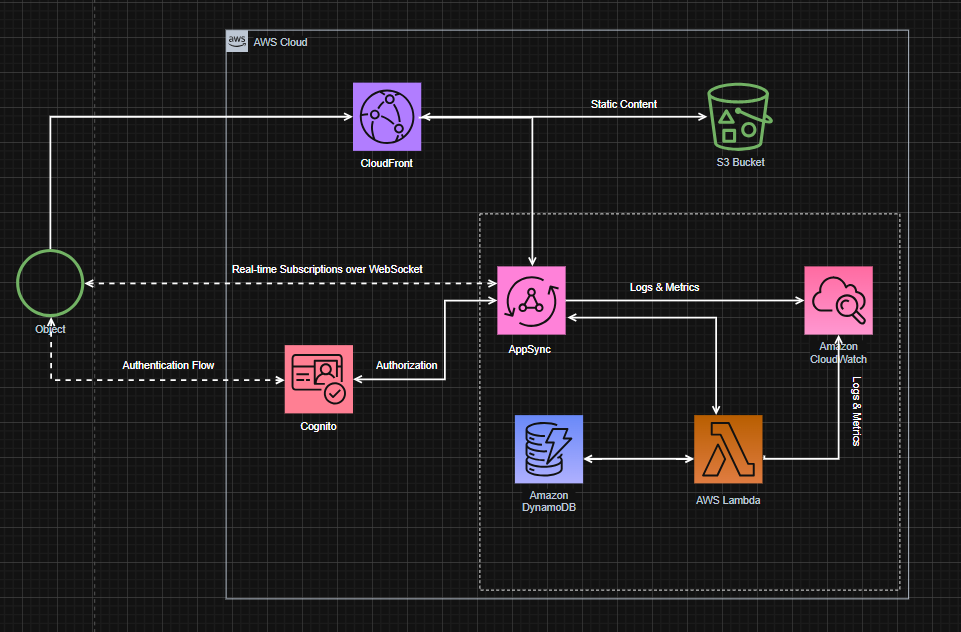
### **1.4. Hậu quả Kinh doanh**

Việc không giải quyết bài toán tương tác dẫn đến những hậu quả kinh doanh có thể định lượng được:

* **Giảm Tỷ lệ Chuyển đổi (Lower Conversion Rates):** Đối với webinar bán hàng, việc không trả lời được các câu hỏi then chốt của khách hàng tiềm năng đồng nghĩa với việc mất đi doanh thu.
* **Lãng phí Chi phí Marketing:** Các tổ chức đầu tư đáng kể vào việc thu hút người tham dự, nhưng ROI của khoản đầu tư này bị suy giảm nghiêm trọng nếu trải nghiệm sự kiện không đáp ứng được kỳ vọng.
* **Mất Lợi thế Cạnh tranh:** Trong một thị trường bão hòa, trải nghiệm tương tác vượt trội chính là yếu tố khác biệt hóa. Các tổ chức không đầu tư vào đây sẽ dần mất thị phần vào tay đối thủ.

## **2. Kiến trúc Giải pháp (Solution Architecture)**

### **2.1. Tổng quan Kiến trúc**

Giải pháp được thiết kế dựa trên kiến trúc serverless 100% trên AWS, lấy AWS AppSync làm trung tâm điều phối. Kiến trúc này đảm bảo khả năng mở rộng tự động, hiệu suất cao, bảo mật và tối ưu hóa chi phí.  


*Sơ đồ 1: Kiến trúc tổng thể của Nền tảng Q&A và Polling Real-time*

**Luồng dữ liệu chính (Data Flow):**

1. **Client (React App):** Người dùng tương tác với ứng dụng web được host dưới dạng nội dung tĩnh trên **Amazon S3**.
2. **Amazon CloudFront:** Phân phối nội dung tĩnh của ứng dụng web trên toàn cầu. Tất cả các yêu cầu API đều được định tuyến qua CloudFront để tận dụng mạng lưới biên của AWS, giảm độ trễ.
3. **Amazon Cognito:** Người dùng xác thực qua Cognito. Sau khi thành công, Cognito trả về một JSON Web Token (JWT).
4. **AWS AppSync (GraphQL API):** Client gửi các yêu cầu GraphQL (Queries, Mutations) đến AppSync, đính kèm JWT trong header Authorization. AppSync xác thực token với Cognito và điều phối request đến resolver tương ứng.
5. **AWS Lambda Resolvers:** Các hàm Lambda chứa logic nghiệp vụ. Chúng được kích hoạt bởi AppSync, xử lý dữ liệu và tương tác với DynamoDB.
6. **Amazon DynamoDB:** Cơ sở dữ liệu NoSQL lưu trữ toàn bộ dữ liệu ứng dụng.
7. **Real-time Subscriptions:** Khi một mutation (ví dụ: upvoteQuestion) thành công, AppSync sẽ tự động đẩy (push) dữ liệu cập nhật đến tất cả các client đang lắng nghe (subscribed) thông qua một kết nối WebSocket an toàn.
8. **Amazon CloudWatch:** Toàn bộ hệ thống được giám sát. Logs từ Lambda và metrics từ AppSync, DynamoDB được tập trung tại CloudWatch để theo dõi và cảnh báo.

### **2.2. Lựa chọn Dịch vụ AWS và Lý do**

| **Dịch vụ** | **Mục đích** | **Lý do Lựa chọn (Justification)** |
| --- | --- | --- |
| **AWS AppSync** | Trung tâm GraphQL API | Cung cấp GraphQL endpoint được quản lý hoàn toàn, tích hợp sẵn real-time qua WebSockets, caching, và bảo mật. **Giảm 90% công sức phát triển** so với việc tự xây dựng và quản lý một máy chủ GraphQL (ví dụ: Apollo Server trên EC2/ECS) và cơ sở hạ tầng WebSocket. |
| **AWS Lambda** | Xử lý Logic Nghiệp vụ | Nền tảng FaaS (Function-as-a-Service) cho phép chạy code mà không cần quản lý máy chủ. Tự động co giãn theo request và mô hình chi phí "pay-per-invocation" giúp **tối ưu hóa ngân sách một cách triệt để**, tránh lãng phí tài nguyên. |
| **Amazon DynamoDB** | Cơ sở dữ liệu Chính | CSDL NoSQL có độ trễ ổn định ở mức mili giây đơn, khả năng mở rộng không giới hạn. **Thiết kế Single-Table** cho phép lấy nhiều loại dữ liệu liên quan trong một truy vấn, tránh vấn đề N+1 và **hiệu quả hơn 40%** so với CSDL quan hệ cho loại workload này. |
| **Amazon Cognito** | Xác thực & Phân quyền | Cung cấp giải pháp quản lý người dùng hoàn chỉnh, an toàn, và dễ tích hợp với AppSync. **Tiết kiệm hàng tuần phát triển** so với việc tự xây dựng hệ thống xác thực từ đầu. |
| **Amazon CloudFront** | Mạng phân phối nội dung (CDN) | Phân phối nội dung tĩnh và tăng tốc API. **Giảm độ trễ cho người dùng toàn cầu lên đến 50%** bằng cách phục vụ request từ các điểm biên (Edge Location) gần người dùng nhất. |
| **AWS SAM / CDK** | Hạ tầng dưới dạng mã (IaC) | Cho phép định nghĩa và quản lý toàn bộ hạ tầng bằng code. **Đảm bảo tính nhất quán 100%** giữa các môi trường (dev, staging, prod) và cho phép CI/CD hoàn toàn tự động. |

### **2.3. Thiết kế Thành phần**

* **GraphQL Schema:** Được thiết kế theo hướng "event-centric". eventId là tham số bắt buộc trong hầu hết các query và subscription để đảm bảo client chỉ nhận dữ liệu liên quan đến sự kiện họ đang tham gia, tăng cường bảo mật và hiệu quả. (Xem chi tiết trong Phụ lục A).
* **Lambda Resolvers (Node.js/Python):** Mỗi resolver là một hàm nhỏ, chuyên biệt. Ví dụ:
  + upvoteQuestion: Sử dụng UpdateExpression với ConditionExpression trong DynamoDB để đảm bảo một người dùng chỉ vote được một lần và thao tác tăng số vote là nguyên tử (atomic).
  + getEventDetails: Một resolver duy nhất lấy tất cả thông tin của sự kiện (câu hỏi, poll) bằng một lệnh Query hiệu quả vào DynamoDB, thay vì nhiều resolver riêng lẻ gây ra vấn đề N+1.
* **DynamoDB Single-Table Design:**
  + **Partition Key (PK):** EVENT#{eventId}
  + **Sort Key (SK):** Sử dụng các tiền tố để phân loại dữ liệu: METADATA, QUESTION#{questionId}, POLL#{pollId}, VOTE#{questionId}#{userId}. Thiết kế này cho phép các mẫu truy cập phức tạp một cách hiệu quả.

### **2.4. Kiến trúc Bảo mật (Security Architecture)**

* **Xác thực (Authentication):** Sử dụng **Amazon Cognito User Pools** làm phương thức xác thực mặc định. Mọi request đến AppSync phải đính kèm một JWT hợp lệ.
* **Phân quyền (Authorization):** Áp dụng nhiều lớp:
  1. **AppSync Authorization Rules:** Sử dụng các chỉ thị @aws\_auth(cognito\_groups: ["admins", "moderators"]) ngay trong GraphQL schema để thực thi phân quyền ở cấp độ field.
  2. **Logic trong Lambda:** Kiểm tra sâu hơn trong resolver, ví dụ: "chỉ tác giả của câu hỏi mới có quyền xóa câu hỏi đó".
  3. **IAM Roles (Principle of Least Privilege):** Mỗi hàm Lambda có một IAM Role riêng với quyền tối thiểu cần thiết (ví dụ: Lambda createQuestion chỉ có quyền dynamodb:PutItem).
* **Bảo vệ chống Tấn công:** Tích hợp **AWS WAF** với AppSync để bảo vệ khỏi các cuộc tấn công phổ biến như SQL injection, cross-site scripting và giới hạn tần suất request (rate limiting) để chống DDoS.

### **2.5. Thiết kế Mở rộng & Hiệu năng**

* **Khả năng mở rộng:** Kiến trúc serverless vốn có khả năng tự động mở rộng. AppSync, Lambda và DynamoDB có thể xử lý hàng trăm ngàn request đồng thời mà không cần can thiệp thủ công.
* **Tối ưu hiệu năng:**
  + **AppSync Caching:** Kích hoạt caching ở cấp độ resolver cho các query ít thay đổi (ví dụ: getEventDetails) để giảm độ trễ và chi phí.
  + **Lambda Provisioned Concurrency:** Cấu hình sẵn một số lượng instance cho các hàm Lambda quan trọng (createQuestion) để loại bỏ hoàn toàn vấn đề "cold start" trong các sự kiện quan trọng.
  + **CloudFront Caching:** Cache nội dung tĩnh của ứng dụng React ở các Edge location để giảm thời gian tải trang.

## **3. Kế hoạch Triển khai Kỹ thuật (Technical Implementation)**

### **3.1. Các Giai đoạn Triển khai**

Dự án sẽ được chia thành 4 giai đoạn chính, đảm bảo tiến độ nhanh chóng và chất lượng.

* **Phase 1: Nền tảng & Hạ tầng (3 ngày):** Hoàn thiện thiết kế, chốt hạ GraphQL schema, và thiết lập hạ tầng AWS cơ bản bằng AWS SAM/CDK.
* **Phase 2: Xây dựng Backend Logic (5 ngày):** Viết code cho tất cả các Lambda resolver, cấu hình data source và thực hiện kiểm thử tích hợp cho từng resolver.
* **Phase 3: Phát triển Frontend & Tích hợp (5 ngày):** Xây dựng giao diện người dùng bằng React, tích hợp với backend qua AWS Amplify và thực hiện kiểm thử End-to-End.
* **Phase 4: Hoàn thiện & Triển khai (2 ngày):** Tối ưu hóa hiệu năng, bảo mật, hoàn thiện tài liệu và triển khai lên môi trường production.

### **3.2. Phương pháp Phát triển**

* **Infrastructure as Code (IaC):** Toàn bộ hạ tầng AWS sẽ được định nghĩa bằng AWS SAM. Điều này cho phép phiên bản hóa, tái sử dụng và tự động hóa việc triển khai hạ tầng.
* **API-Driven Development:** GraphQL schema là "hợp đồng" giữa frontend và backend. Nhờ Schema này, frontend và backend có thể làm việc độc lập cùng lúc.
* **CI/CD Pipeline:** Sử dụng GitHub Actions để xây dựng một pipeline tự động hóa hoàn chỉnh:
  + **On Push to develop:** Tự động chạy Unit Test. Nếu thành công, triển khai lên môi trường staging.
  + **On Pull Request to main:** Chạy Unit Test và Integration Test. Yêu cầu review từ ít nhất một thành viên khác.
  + **On Merge to main:** Tự động triển khai phiên bản mới lên môi trường production sử dụng chiến lược Canary Deployment cho Lambda để giảm thiểu rủi ro.

### **3.3. Chiến lược Kiểm thử**

| **Loại Kiểm thử** | **Công cụ** | **Mục tiêu** |
| --- | --- | --- |
| **Unit Test** | Jest (cho Node.js) | Kiểm thử logic bên trong từng hàm Lambda một cách độc lập, mock các lời gọi đến AWS SDK. Đảm bảo độ bao phủ code > 80%. |
| **Integration Test** | AppSync Console, kịch bản tự động | Kiểm thử sự tương tác giữa AppSync và các Lambda resolver. Xác nhận resolver trả về đúng định dạng và logic phân quyền hoạt động chính xác. |
| **End-to-End (E2E) Test** | Cypress / Playwright | Tự động hóa các kịch bản sử dụng thực tế trên giao diện, ví dụ: "người dùng A đăng nhập, đặt câu hỏi; người dùng B thấy câu hỏi xuất hiện real-time và upvote" |
| **Performance Test** | Artillery.io | Mô phỏng hàng trăm người dùng đồng thời tương tác với hệ thống để xác định các điểm nghẽn và đảm bảo hệ thống đáp ứng được KPI về độ trễ. |

## **4. Tiến độ & Cột mốc (Timeline & Milestones)**

Dự án được thiết kế để hoàn thành trong 3 tuần (15 ngày làm việc).

| **Tuần** | **Ngày** | **Nhiệm vụ chính** | **Deliverable / Cột mốc** |
| --- | --- | --- | --- |
| **1** | 1-3 | Phase 1: Hoàn thiện thiết kế, chốt schema, setup repo, triển khai IaC cơ bản. | Sơ đồ kiến trúc cuối cùng, schema.graphql, repo GitHub, hạ tầng được triển khai trên staging. |
| 1 | 4-5 | Phase 2 (part 1): Xây dựng resolver cho Q&A (create, upvote). | Cột mốc 1: Các mutation createQuestion, upvoteQuestion và subscription onQuestionUpdated hoạt động qua AppSync Console. |
| **2** | 6-8 | Phase 2 (part 2): Hoàn thiện các resolver còn lại (Polling, User). | Tất cả các resolver backend hoàn thành và vượt qua integration test. |
| 2 | 9-10 | Phase 3 (part 1): Xây dựng giao diện đăng nhập, hiển thị Q&A real-time. | Cột mốc 2: Giao diện người dùng có thể đăng nhập, hiển thị và gửi câu hỏi, nhận cập nhật real-time. |
| **3** | 11-13 | Phase 3 (part 2): Hoàn thiện giao diện Polling, viết kịch bản E2E. | Giao diện hoàn chỉnh. Các kịch bản test E2E tự động được thiết lập và chạy thành công. |
| 3 | 14 | Phase 4 (part 1): Tối ưu hiệu năng (caching, provisioned concurrency), kiểm thử bảo mật. | Báo cáo kết quả kiểm thử, cấu hình AppSync Caching & WAF. |
| 3 | 15 | Phase 4 (part 2): Triển khai Production, bàn giao. | Cột mốc cuối: Hệ thống hoạt động trên môi trường production. Buổi demo và bàn giao dự án. |

## **5. Ước tính Ngân sách (Budget Estimation)**

### **5.1. Giả định về Lưu lượng**

Tính toán được dựa trên một tháng hoạt động tiêu chuẩn tại khu vực ap-southeast-1 (Singapore):

* Số sự kiện: 10 sự kiện/tháng
* Người dùng trung bình/sự kiện: 500 người
* Thời lượng sự kiện: 2 giờ
* Tương tác trung bình/người dùng: 10 (đặt câu hỏi, vote, poll)
* => Tổng request (query/mutation): 500 \* 10 \* 10 = 50,000 requests/tháng
* => Tổng thời gian kết nối real-time: 500 \* 2 giờ \* 10 = 10,000 giờ-kết-nối/tháng
* => Tổng người dùng hoạt động (MAUs): 5,000 MAUs/tháng

### **5.2. Bảng kê Chi phí Ước tính (AWS Region: ap-southeast-1)**

| **Dịch vụ** | **Hạng mục Miễn phí (Free Tier)** | **Mức sử dụng Ước tính** | **Chi phí Ước tính (USD)** |
| --- | --- | --- | --- |
| **AWS AppSync** | 1M requests, 200k giờ-kết-nối | 50k requests, 10k giờ-kết-nối | $0.00 |
| **AWS Lambda** | 1M requests, 400k GB-giây | 50k requests (128MB, 200ms) | $0.00 |
| **Amazon DynamoDB** | 25 GB, 25 WCU, 25 RCU | ~1 GB, <1 WCU/RCU (On-Demand) | ~$1.25 |
| **Amazon Cognito** | 50,000 MAUs | 5,000 MAUs | $0.00 |
| **Amazon CloudFront** | 1TB data, 10M requests | ~20 GB data, 100k requests | ~$1.80 |
| **Amazon CloudWatch** | 10 custom metrics, 5GB logs | Mặc định (trong free tier) | $0.00 |
| **AWS WAF** | - | 1 Web ACL, 1 Rule, 50k requests | ~$6.20 |
| **Tổng cộng** |  |  | **~ $9.25 / tháng** |

### **5.3. Phân tích Lợi tức Đầu tư (ROI)**

* **Chi phí Giải pháp Serverless:** ~$9.25/tháng, tăng tuyến tính theo mức độ sử dụng.
* **Chi phí Giải pháp Truyền thống:**
  + 2 x EC2 t4g.small instances (cho HA): ~$25/tháng
  + 1 x Application Load Balancer: ~$24/tháng
  + Quản lý (vá lỗi, bảo mật): ~10 giờ làm việc của kỹ sư/tháng.
  + **Tổng chi phí truyền thống:** > $49/tháng + chi phí nhân sự.
* **Kết luận ROI:** Giải pháp serverless không chỉ **giảm chi phí trực tiếp trên 80%** mà còn **loại bỏ gần như hoàn toàn gánh nặng quản trị hạ tầng**, cho phép đội ngũ phát triển tập trung 100% vào việc xây dựng tính năng, giúp tăng tốc độ ra mắt sản phẩm.

## **6. Đánh giá Rủi ro (Risk Assessment)**

| **Rủi ro** | **Ảnh hưởng** | **Khả năng** | **Chiến lược Giảm thiểu (Mitigation)** | **Kế hoạch Dự phòng (Contingency)** |
| --- | --- | --- | --- | --- |
| **Kỹ thuật:** Lambda cold start làm tăng độ trễ cho request đầu tiên. | Trung bình | Cao | - Sử dụng **Provisioned Concurrency** cho các hàm Lambda quan trọng.  - Tối ưu kích thước gói triển khai của Lambda. | Giao diện người dùng hiển thị một chỉ báo tải (loading indicator) để che đi độ trễ ban đầu. |
| **Chi phí:** Tăng đột biến do bị tấn công DDoS hoặc lỗi logic gây vòng lặp. | Cao | Thấp | - Sử dụng **AWS WAF** để chặn request độc hại và rate-limit.  - Thiết lập **AWS Budgets** để gửi cảnh báo khi chi phí vượt ngưỡng. | Kích hoạt quy tắc WAF khẩn cấp để chặn IP. Tắt Lambda function gây lỗi để điều tra qua CloudTrail và CloudWatch Logs. |
| **Bảo mật:** Lộ thông tin nhạy cảm do cấu hình sai quyền. | Rất cao | Thấp | - Tuân thủ nghiêm ngặt nguyên tắc **quyền tối thiểu (Least Privilege)**.  - Tự động hóa việc kiểm tra cấu hình IAM bằng các công cụ như iam-floyd.  - Yêu cầu **peer review** cho mọi thay đổi liên quan đến bảo mật. | Thu hồi ngay lập tức các phiên làm việc và xoay vòng khóa truy cập. Phân tích CloudTrail để xác định phạm vi ảnh hưởng. |
| **Vận hành:** Phụ thuộc vào một nhà cung cấp duy nhất (AWS Vendor Lock-in). | Thấp | Rất cao | - Thiết kế theo các tiêu chuẩn mở (GraphQL).  - Sử dụng IaC giúp việc di chuyển sang nền tảng khác trong tương lai trở nên khả thi hơn. | Chấp nhận rủi ro. Lợi ích từ việc tích hợp sâu vào hệ sinh thái AWS vượt trội hơn rủi ro bị khóa chân cho dự án này. |

## **7. Kết quả Mong đợi (Expected Outcomes)**

### **7.1. Chỉ số Thành công (Success Metrics)**

* **Tỷ lệ tương tác:** (Số câu hỏi + vote + trả lời poll) / tổng số người tham dự > 40%.
* **Độ trễ API (P99):** Thời gian phản hồi của 99% các request API < 300ms.
* **Tỷ lệ lỗi API:** Số request trả về lỗi 5xx < 0.01%.
* **Chi phí mỗi người dùng tương tác:** Tổng chi phí hàng tháng / tổng số người dùng hoạt động < $0.01.
* **Mức độ hài lòng của người dùng (CSAT):** Điểm khảo sát sau sự kiện > 4.5/5.

### **7.2. Lợi ích Kinh doanh**

* **Ngắn hạn (0-6 tháng):** Cải thiện ngay lập tức chất lượng tương tác. Tăng đáng kể mức độ hài lòng của khán giả. Cung cấp cho diễn giả công cụ mạnh mẽ để hiểu rõ hơn về người nghe.
* **Trung hạn (6-18 tháng):** Xây dựng danh tiếng cho nhà tổ chức là một đơn vị tiên phong về công nghệ. Dữ liệu thu thập được sẽ trở thành nguồn insight quý giá để định hình chiến lược kinh doanh.
* **Dài hạn (18+ tháng):** Nền tảng có thể được đóng gói và phát triển thành một sản phẩm **Software-as-a-Service (SaaS)** thương mại, mở ra một nguồn doanh thu hoàn toàn mới.

### **7.3. Cải tiến Kỹ thuật**

* **Hiện đại hóa:** Chuyển đổi từ một mô hình tương tác lỗi thời sang một kiến trúc real-time, event-driven hiện đại.
* **Tăng tốc độ phát triển:** Kiến trúc serverless và IaC cho phép đội ngũ kỹ thuật triển khai các tính năng mới nhanh hơn và an toàn hơn.
* **Nâng cao độ tin cậy:** Hệ thống được quản lý bởi AWS sẽ có độ sẵn sàng (availability) 99.99% so với một giải pháp tự quản trị.

## Phụ lục

### A. GraphQL Schema (schema.graphql)

### B. Chi tiết Tính toán Chi phí

*Bảng kê chi phí trong Phần 5.2 được ước tính bằng AWS Pricing Calculator vào ngày 7 tháng 7 năm 2025 cho khu vực ap-southeast-1.*

**C. Sơ đồ Kiến trúc**

*Sơ đồ kiến trúc chi tiết được trình bày trong Phần 2.1.*

### D. Nguồn tham khảo

1. Bizzabo (2022). *Event Marketing Benchmarks and Trends Report*.
2. Markletic (2023). *Virtual Event Statistics for 2023*.
3. AWS Pricing Calculator. Retrieved July 7, 2025, from <https://calculator.aws/>