# **Event Announcement System — Architecture Overview**

## **Introduction**

I developed a fully serverless Event Announcement web application leveraging core AWS services. The system enables users to:

* Subscribe to event notifications via email.
* View a dynamic, up-to-date list of upcoming events.
* Create new events through an interactive form on the website.

The frontend features a clean and responsive UI, hosted on Amazon S3 for fast and scalable delivery. All backend processing, including managing subscriptions and event data, is securely handled through AWS Lambda functions, triggered via API Gateway endpoints. This design ensures efficient operation, high availability, and strong security through managed AWS services.

## **Objectives**

The project is designed to:

* Host the frontend static website using Amazon S3.
* Store event data in a JSON file within S3, enabling easy retrieval and updates.
* Enable users to subscribe to event notifications via AWS Simple Notification Service (SNS).
* Provide an intuitive form for users to submit new events, which triggers Lambda functions to update data and notify subscribers.
* Use API Gateway as a secure interface between the frontend and backend Lambda functions.

## **Technologies and AWS Services Used**

* **Amazon S3:** Hosting static frontend assets and storing the events JSON file.
* **Amazon SNS:** Managing email subscriptions and distributing event notifications.
* **AWS Lambda:** Backend compute to handle subscription management and event creation.
* **Amazon API Gateway:** Exposing RESTful API endpoints for frontend-backend communication.
* **IAM Roles and Policies:** Enforcing secure, least-privilege access for Lambda functions and API Gateway integration.

## **Logical Flow**

1. Static frontend files (HTML, CSS, and events.json) are uploaded to an S3 bucket configured for static website hosting, enabling public access to the user interface.
2. API Gateway is configured with two REST endpoints: /subscribe for subscription requests and /create-event for event creation.
3. The Subscription Lambda function processes incoming subscription requests, adding user emails to the SNS topic.
4. The Event Creation Lambda function reads the current events.json from S3, appends new event data submitted by users, updates the file in S3, and triggers SNS notifications to subscribers.

## **Estimated Time and Cost**

* Estimated development time: 2–3 hours.
* Cost: Fully eligible under AWS Free Tier, ensuring minimal or no cost for light usage.