AWS AppSync for Building GraphQL APIs

A Project Based Learning Report Submitted in partial fulfilment of the requirements for the award of the degree

of

Bachelor of Technology

in the Department of Computer Science & Engineering

Cloud Based AI/ML Speciality (22SDCS07A)

Submitted by
2210030133: Sharikar Sneha

Under the guidance of

Ms. P. Sree Lakshmi

Department of Computer Science and Engineering

Koneru Lakshmaiah Education Foundation, Aziz Nagar

Aziz Nagar – 500075

March - 2025.

Introduction

AWS AppSync is a managed service that simplifies building scalable GraphQL APIs, enabling applications to interact with multiple data sources in real time. By leveraging AppSync, developers can create APIs that fetch, update, and synchronize data efficiently across web and mobile applications. This project focuses on building a GraphQL API using AWS AppSync, integrating with AWS services such as DynamoDB, Lambda, and Cognito. The API ensures seamless data retrieval, real-time updates, and secure authentication. With AWS's serverless capabilities, the system remains highly available, scalable, and cost-effective, making it ideal for modern applications requiring dynamic data access.

AWS Services

1. **AWS AppSync**: Manages GraphQL API requests, real-time data synchronization, and caching.
2. **Amazon DynamoDB**: Provides a serverless NoSQL database with high performance and scalability.
3. **AWS Lambda**: Executes backend business logic, processing GraphQL queries and mutations.
4. **Amazon Cognito**: Manages user authentication and access control, ensuring secure API interactions.
5. **AWS CloudWatch**: Monitors API performance, logs requests, and detects anomalies.

Project Purpose and Expected Outcome

The primary goal of this project is to develop a scalable, secure, and high-performing GraphQL API using AWS AppSync. This API will serve as an efficient interface for client applications to interact with backend services, providing real-time data access and seamless synchronization across devices. By integrating various AWS services such as DynamoDB, Lambda, and Cognito, the project ensures optimized data retrieval, high availability, and cost-effective scalability.

Security is a key aspect, achieved through Amazon Cognito, which manages user authentication and access control, ensuring only authorized users interact with the API. AWS AppSync’s real-time subscription capabilities further enhance the system by delivering live updates to clients whenever data changes occur.

The expected outcome of this project is a fully functional GraphQL API that allows efficient querying and mutation of data, supports secure user authentication, and enables real-time event-driven updates. This makes it ideal for use cases such as chat applications, collaborative tools, IoT data streaming, and live dashboards. By leveraging AWS's serverless and managed services, the system remains highly scalable, eliminating the need for complex infrastructure management while ensuring a cost-effective and resilient solution for modern application development.

Methodology

Architecture and Workflow

The architecture leverages AWS AppSync to build scalable GraphQL APIs, integrating services for authentication, data storage, and real-time updates. It ensures secure access, efficient query resolution, and seamless data synchronization across clients. The workflow includes:

1. Users send GraphQL queries, mutations, or subscriptions to AWS AppSync from web or mobile clients.
2. AWS Cognito handles authentication, ensuring secure API access with role-based permissions.
3. AppSync processes requests and routes them to Amazon DynamoDB for data retrieval or updates.
4. AWS Lambda executes custom business logic for complex queries before returning responses to AppSync.
5. GraphQL subscriptions enable real-time updates, pushing data changes to subscribed clients.
6. Amazon CloudWatch and AWS X-Ray monitor API performance, logging errors and optimizing request handling.

AWS Services Interaction

- AWS Sumerian interacts with AWS Lambda to process customer queries, fetch real-time data from DynamoDB, and provide responses.
- Users send GraphQL queries, mutations, or subscriptions through a web or mobile client, which AWS AppSync processes and routes accordingly.
- AWS Cognito handles authentication, verifying user credentials and ensuring role-based access before allowing requests.
- AppSync interacts with Amazon DynamoDB to store and retrieve structured data, ensuring fast and scalable database operations.
- For complex processing, AppSync triggers AWS Lambda functions to execute business logic and return computed results.
- Real-time updates are managed through GraphQL subscriptions, where changes in DynamoDB or Lambda outputs trigger automatic client updates.
- Amazon CloudWatch and AWS X-Ray monitor system performance, track API requests, and provide debugging insights for optimization.

Justification for AWS Service Selection

The AWS AppSync enables real-time data synchronization and efficient GraphQL API management. AWS Cognito ensures secure authentication, while DynamoDB provides fast, scalable data storage.

Implementation

AWS Infrastructure Setup

- Create an AWS AppSync API – Set up a GraphQL API in AWS AppSync, defining schemas and resolvers for data queries and mutations.
- Configure Amazon Cognito – Enable authentication and authorization using Cognito User Pools to secure API access.
- Deploy Amazon DynamoDB – Create tables to store structured application data, ensuring fast and scalable query performance.
- Set Up AWS Lambda Functions – Develop backend logic for complex queries and mutations, integrating Lambda as a data source for AppSync.
- Enable Amazon CloudWatch Monitoring – Configure logging and metrics to track API performance, error rates, and request latency.
- Integrate Amazon S3 (if needed) – Store media assets and static content in S3, providing seamless access through the API.
- Test and Optimize – Validate API requests, optimize query performance, and ensure secure, real-time data access across services.

Security Policies, IAM Roles, and Access Controls

- IAM Roles & Policies – Define least-privilege access roles for AWS AppSync, Lambda, and DynamoDB, ensuring only authorized services can access data.
- Amazon Cognito Security – Implement Multi-Factor Authentication (MFA) and role-based access control (RBAC) to manage user authentication securely.
- AppSync API Authorization – Configure authentication modes such as API keys, IAM roles, or Cognito user pools to restrict unauthorized API access.
- DynamoDB Encryption – Enable encryption at rest and in transit to protect sensitive data from unauthorized access.
- Lambda Execution Permissions – Restrict Lambda functions using IAM policies, ensuring they can only access the necessary AppSync queries and database operations.
- CloudWatch Security Auditing – Monitor API calls, track authentication attempts, and log unauthorized access attempts for security compliance.

Automation

- Automated Query Execution – AWS AppSync triggers GraphQL queries and mutations automatically based on user requests, reducing manual intervention.
- Lambda Function Invocation – AWS Lambda executes backend logic dynamically, processing API requests and updating DynamoDB without manual triggers.
- Real-Time Data Sync – AppSync utilizes subscriptions to push real-time updates to connected clients whenever data changes in DynamoDB.
- CloudWatch Monitoring – Automated logging and alerts track API usage, identify

performance bottlenecks, and detect anomalies in real-time.

- DynamoDB Auto Scaling – The database automatically scales read/write capacity based on demand, ensuring optimal performance without manual adjustments.
- Scheduled Backups – AWS Backup automates periodic snapshots of DynamoDB to ensure data recovery and long-term retention.