

Challenges and Learnings from AWS Lambda and DynamoDB

During the process of building a serverless web application using AWS Lambda and DynamoDB, I encountered several challenges, each offering valuable learning opportunities.

1. Setting up DynamoDB Table: First step is to create a table with table name as StudentRecords and primary key as student_id.

2. Setting up Lambda Functions: One of the initial challenges was configuring the correct permissions for Lambda to interact with DynamoDB. I had to ensure that the Lambda function was assigned a role with sufficient permissions otherwise create a new role from AWS policy templates. This helped me understand how critical AWS Identity and Access Management (IAM) roles are for securely managing services in AWS.

3. Understanding API Gateway Integration: Integrating Lambda with API Gateway required careful mapping of HTTP methods to Lambda functions. At first, I found it difficult to correctly configure methods like POST, GET, PUT, and DELETE. Learning how to set up API Gateway endpoints, resources, and deploying the API was eye-opening, and it helped me appreciate how API Gateway acts as a bridge between external HTTP requests and AWS Lambda.

4. Dealing with API Responses and Error Handling: Another challenge was managing API responses and error handling. For instance, ensuring that the API Gateway returned appropriate HTTP status codes for different CRUD operations (e.g., 200 for success, 400 for bad requests) was crucial for creating a robust API. This process taught me the importance of proper response formatting, including handling edge cases like missing or incorrect data.