Project 3: Serverless Contact Form Using AWS Lambda

Problem Statement

Collecting and managing contact form submissions can be challenging without a backend.

Objective

Use AWS Lambda and API Gateway to collect form data and store it in DynamoDB, without managing a server.

Requirements

- AWS Account
- Lambda function (Node.js or Python)
- API Gateway
- DynamoDB table
- Basic HTML form

Processing Steps

1. **Create a DynamoDB Table:**
   - Go to AWS DynamoDB and create a table (e.g., ContactSubmissions).
   - Use "email" as the primary key.

2. **Write the Lambda Function:**
   - Use Node.js or Python.
   - Parse the form input (e.g., name, email, message).
   - Use the AWS SDK to insert the data into DynamoDB.

3. **Deploy via API Gateway:**
   - Create a new API in API Gateway.
   - Define a POST endpoint.
   - Link it to the Lambda function.

Project 3: Serverless Contact Form Using AWS Lambda

4. **Create HTML Contact Form:**

   - Include fields for name, email, and message.

   - Set form action to the API Gateway POST URL.

   - Use `fetch()` or form submission with `method="POST"`.

5. **Test the Setup:**

   - Host the HTML file or open it locally.

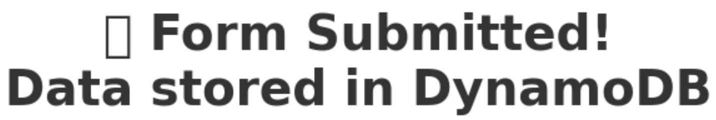
   - Submit the form.

   - Check DynamoDB for stored entries.

Expected Outcome

A serverless contact form that stores submissions in a DynamoDB database automatically.

Sample Output



□ Form Submitted!
Data stored in DynamoDB