AWS Serverless Contact Form Project — Full Step-by-Step Documentation

Project Overview

Project Title: Contact Form API Using API Gateway + Lambda + SES

Objective: Create a serverless contact form where users can submit inquiries that automatically trigger an email to a specified recipient using AWS services.

Step 1: SES Setup

1. Log in to AWS Console → SES (Simple Email Service).
2. Verify two email addresses:
    a. Sender email (SOURCE_EMAIL)
    b. Recipient email (DEST_EMAIL)
3. Ensured SES is in the correct region (e.g.: us-west-1).
4. No sandbox limitations as both emails were verified.

Goal: SES verified and ready.

Step 2: Lambda Function Creation

2.1 Create IAM Role

- Select Lambda as AWS Service for role
- Role Name: Lambda-SESSendEmail-Role (or any as desired)
- Attached Policies:
    - AmazonSESFullAccess (or custom SES policy such AmazonSESSendingAccess for least access privilege).
    - CloudWatchLogsFullAccess

Goal: Ensure Lambda has permission to invoke SES and write logs.

2.2 Create Lambda Function

- Function Name: ContactFormHandler (or any as desired)
- Runtime: Python 3.12
- Permissions: Used the IAM role created above.

2.3 Lambda Code

- Wrote Python code to:
    - Parse JSON body from API Gateway
    - Send email via SES
    - Return success or error response
- Updated Lambda to handle 'body' key safely and account for Proxy Integration.

2.4 Environment Variables

Key	Value
SES_REGION	us-west-1
SOURCE_EMAIL	verified_sender@example.com
DEST_EMAIL	verified_recipient@example.com

2.5 Lambda Testing

- Used Lambda test event with JSON payload:

```
{
  "body": "{\"name\": \"<Insert any name>\", \"email\":<\Insert verified SES DEST_EMAIL\>", \"message\": \"Hello from Lambda test!\"}"
}
```

Example:

```
{
  "body": "{\"name\": \"John Doe\", \"email\":\"john@example.com\", \"message\": \"Hello from Lambda test!\"}"
}
```

- Received 200 OK response.
- Email successfully delivered to recipient.

Progress: Lambda fully functional.

Step 3: API Gateway Setup

3.1 Create REST API

- Name: ContactFormAPI (or any as desired)
- Endpoint type: Regional

3.2 Create Resource + Method

- Resource Path: /contact
- Method: POST
- Integration: Lambda Function → ContactFormHandler (or name you selected)
- Enable CORS to display **OPTIONS**

3.3 Deploy API

- Stage Name: dev (or any as desired)
- Endpoint URL: Use Invoke Url in POST method.
  e.g.: https://4q5ln4zyoi.execute-api.us-west-1.amazonaws.com/dev/contact

3.4 Test with Postman

- Method: POST
- Headers: Content-Type: application/json
- Body (Select raw and JSON format):

```
{
 "name": "John Doe",
 "email": "johndoe@example.com",
 "message": "Test message to confirm Lambda Proxy Integration and SES email delivery."
}
```

- Receive a 200 OK response.
- Email delivered to DEST_EMAIL.

Goal: API Gateway connected and fully functional.

- Confirmed correct configuration for Lambda Proxy Integration:
- Sent test requests → received 200 OK + email.

Progress: End-to-end flow validated.

Conclusion & Next Steps

- Serverless contact form fully functional: API Gateway → Lambda → SES.
- Emails reliably delivered and logged in CloudWatch.
- Future improvements:
  - Design and host web platform to accept user response.
  - Add CAPTCHA to prevent spam
  - Deploy SES in production with domain and DKIM
  - Rate-limiting in API Gateway
  - Store submissions in DynamoDB