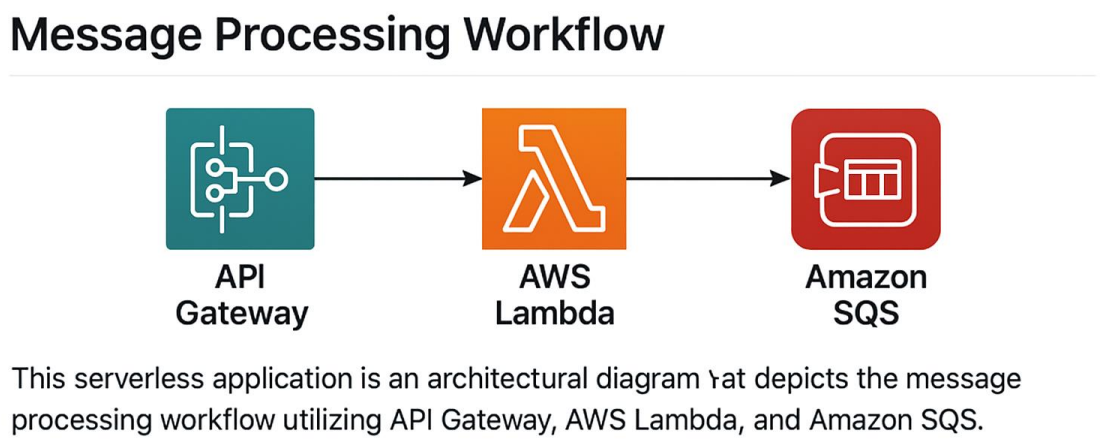
SQS Message Notification Project - README

Overview

This project implements a serverless architecture for sending messages to an Amazon SQS queue using AWS Lambda and API Gateway. The Lambda function is triggered by HTTP requests via API Gateway and sends messages to the SQS queue. The Lambda function now supports external variables passed through the request payload, replacing previously hardcoded values.

Architecture Diagram

The architecture consists of the following components:
1. API Gateway: Accepts HTTP requests from external clients.
2. Lambda Function: Triggered by API Gateway, processes the request and sends a message to SQS.
3. Amazon SQS: Receives messages from the Lambda function.
The Lambda function is configured to parse incoming JSON payloads and extract message parameters dynamically.



Setup Instructions

1. Create an Amazon SQS queue and note the queue URL.
2. Create a Lambda function with permissions to send messages to the SQS queue.

3. Configure API Gateway to trigger the Lambda function via HTTP POST requests.
4. Deploy the API Gateway and note the endpoint URL.
5. Update the Lambda function code to parse external variables from the request payload.

Lambda Function Example

The Lambda function should parse the incoming request body to extract message parameters. Example:

```
import json
import boto3

def lambda_handler(event, context):
    body = json.loads(event['body'])
    message = body.get('message', 'Default message')
    queue_url = body.get('queue_url')

    if not queue_url:
        return {
            'statusCode': 400,
            'body': json.dumps('Missing queue_url in request')
        }

    sqs = boto3.client('sqs')
    response = sqs.send_message(QueueUrl=queue_url, MessageBody=message)

    return {
        'statusCode': 200,
        'body': json.dumps('Message sent successfully')
    }
```