










Serverless Notification System

August 11, 2025

Architecture Overview

1.  **Client** sends order via **API Gateway** (HTTP POST)
2.  **Lambda** processes and validates order
3.  **DynamoDB** stores order metadata
4.  **SNS** publishes notifications
5.  **Subscribers** receive email/SMS alerts

AWS Services Used

 API Gateway	REST endpoint accepting HTTP POST requests
 Lambda	Python 3.12 function for order processing
 DynamoDB	FileMetadata table (PK: FileName)
 SNS	file-upload-notification topic

Implementation Guide

1. **DynamoDB Setup**

- **Table Name:** FileMetadata
- **Primary Key:** FileName (String)
- No secondary indexes needed
- **Create table**

2. SNS Configuration

- **Go to AWS Console** → SNS → Topics → Create Topic
- **Topic Type:** Standard
- **Name:** file-upload-notification
- **Create the topic**

3. Subscribe to SNS Topic

- Select your created topic.
- **Create Subscription**
- **Subscription Protocol:** Email(or SMS)
- **Endpoint:** your email address (or phone number)
- **Create Subscription**
- Check your email and confirm the subscription link.

4. Lambda Function Setup

- **Go to AWS Console** → Lambda → Create Function
- **Name:** processFileUpload
- **Runtime:** Python 3.12 or node.js
- **Execution role: Create or use existing role with these permissions:**
 - DynamoDB (PutItem on OrdersTable)
 - SNS (Publish to OrderNotificationTopic)
 - CloudWatch Logs (for logging)
 - **Create Function**

5. API Gateway Setup

- **Go to AWS Console** → API Gateway → Create API (REST or HTTP)
- Create a resource & method (POST)
- Integrate POST method with Lambda function processFileUpload
- Deploy the API and note the invoke URL.

6. ADD Trigger Setup

- **Go to Lambda** → Your function → Configuration → Triggers
- Add trigger → API Gateway → Select your API and POST method.



Testing Procedure

- Send test POST request:

```
{
  "body": "{\"customerName\": \"John Doe\",
  \"productName\": \"Wireless Mouse\"}"
}
```

- Verify:
 - You should get a response with orderId and success message.
 - Check DynamoDB if the order is saved.
 - Check your email or SMS for notification from SNS.
 - Check Lambda execution logs

Production Considerations

IAM	Least privilege permissions
 Monitoring	CloudWatch alarms for errors
 Logging	Full request/response logging