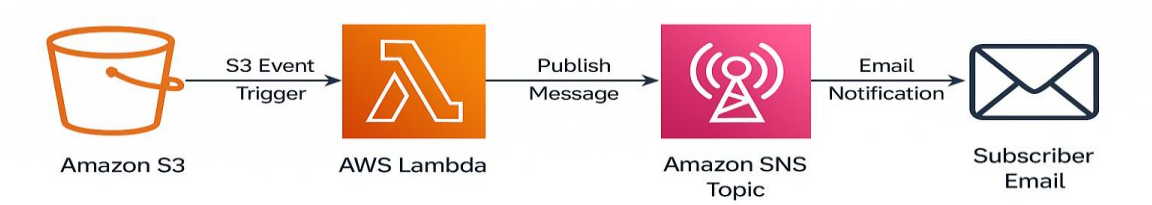
PROJECT: AWS EVENT-DRIVEN ARCHITECTURE

Integration of S3, SNS & Lambda | File Upload Notification System

By: Kiran Rakh

OBJECTIVE

- Build a serverless, event-driven architecture using AWS services:
- - Uploading a file to S3 → Triggers a Lambda function → Sends a notification using SNS → Delivers email alert to the subscriber.
- Overall Architecture Diagram:



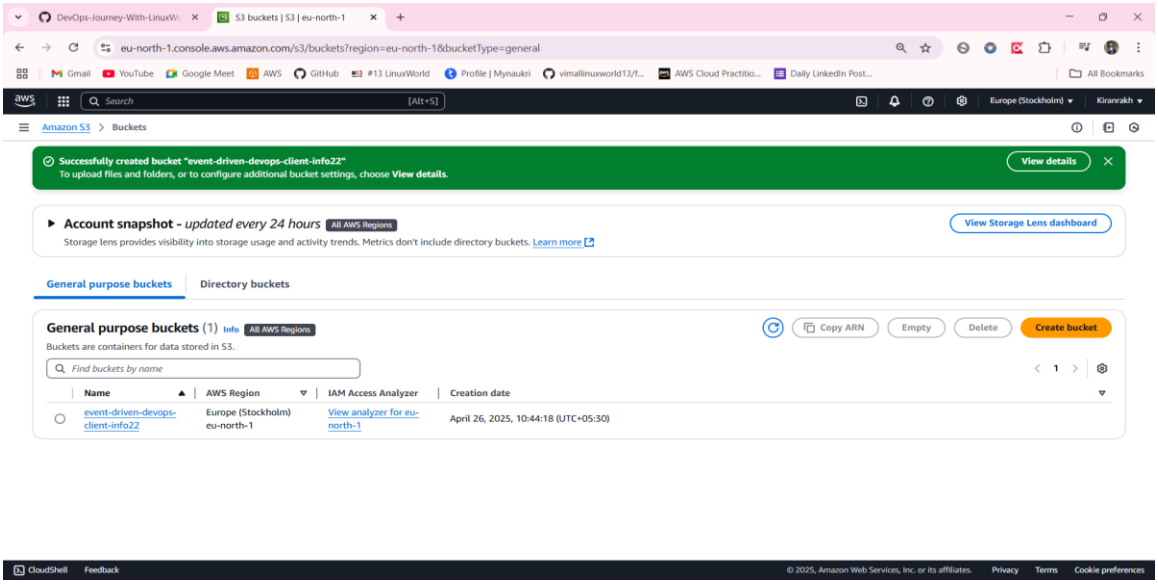
AWS SERVICES USED

- - Amazon S3 (Simple Storage Service)
- - Amazon SNS (Simple Notification Service)
- - AWS Lambda (Python Runtime)
- - AWS CloudWatch (for logs)
- - IAM (Identity and Access Management

🚀 STEP-BY-STEP IMPLEMENTATION

1️⃣ S3: CREATE BUCKET

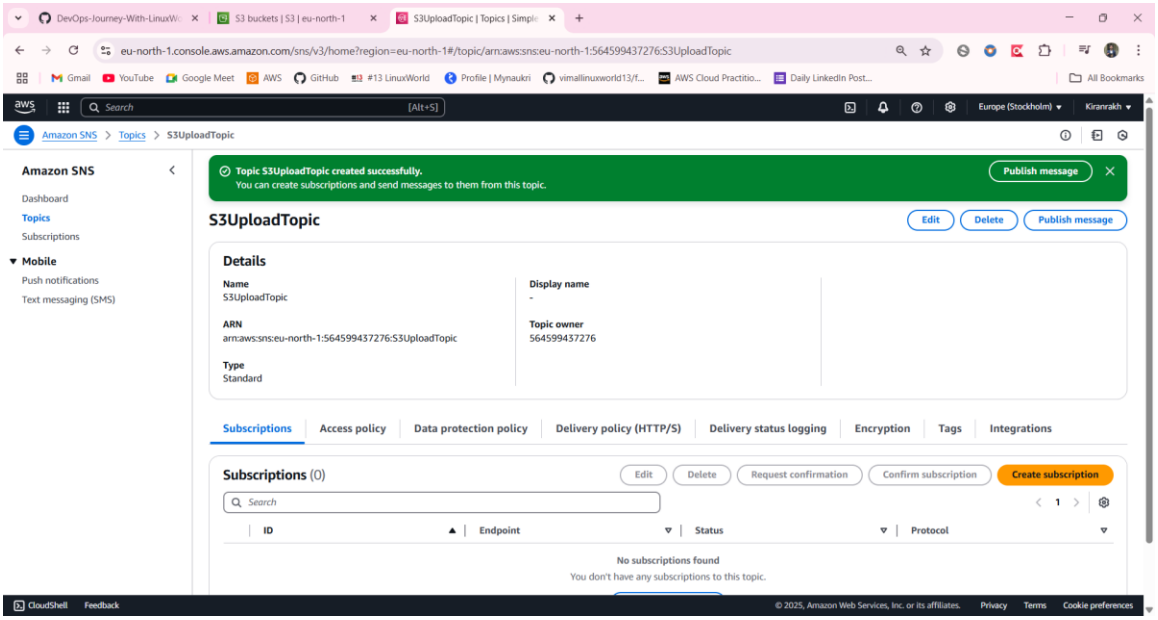
- 1. Go to AWS Console → Services → S3
- 2. Click “Create bucket”
- 3. Enter a unique name (e.g., event-driven-kiran-bucket)
- 4. Leave other settings as default
- 5. Click “Create bucket”

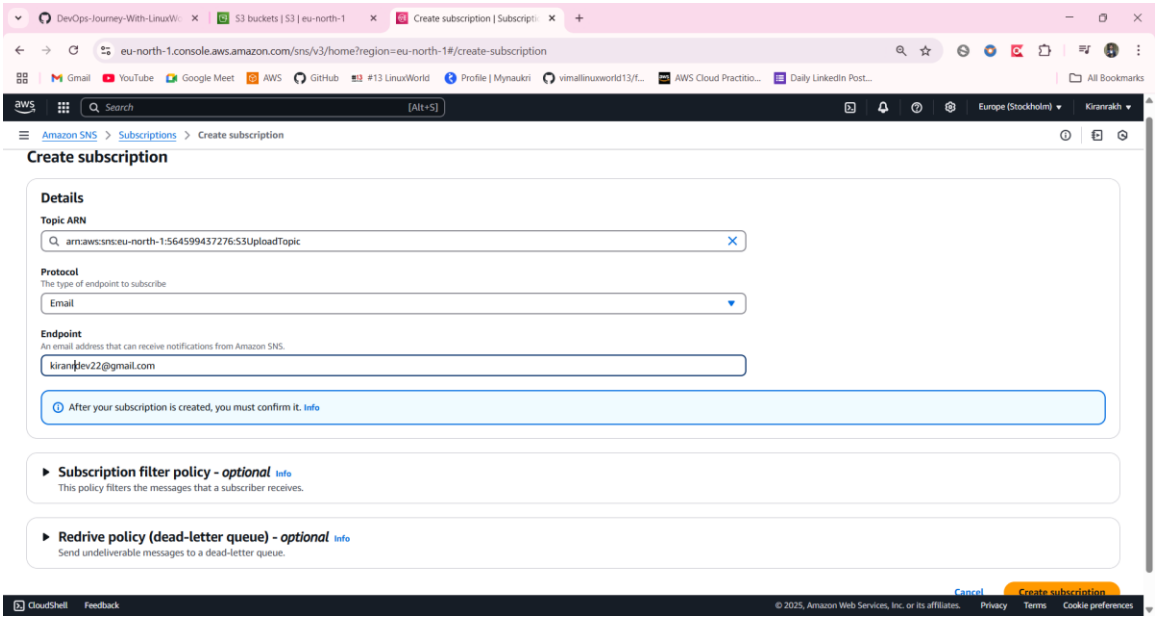


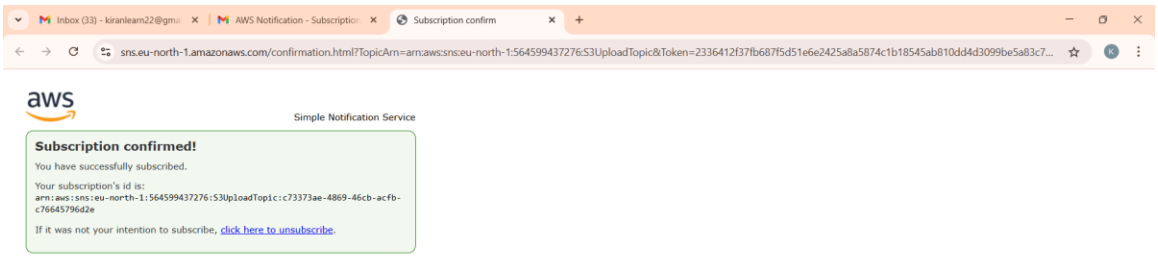
2️⃣ SNS: CREATE TOPIC & SUBSCRIPTION

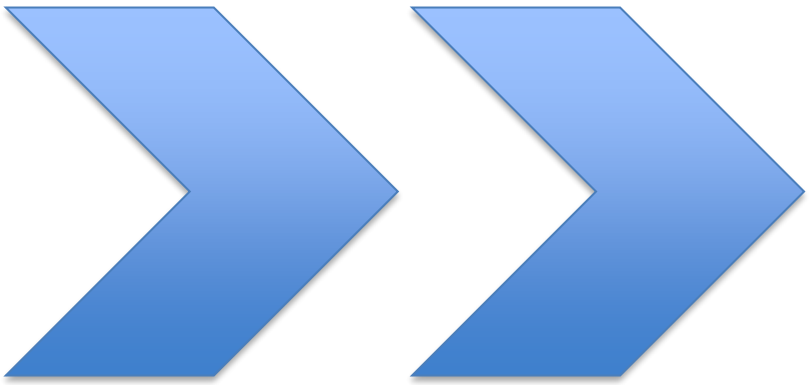
- 1. Go to Services → SNS
- 2. Click “Create topic” → Type: Standard, Name: S3UploadTopic
- 3. Click “Create topic”
- 4. Go to your topic → Click “Create subscription”
- 5. Protocol: Email → Endpoint: kiranrakh155@gmail.com → Click “Create subscription”
- 6. Approve the subscription from your email inbox

- SNS topic and email subscription:



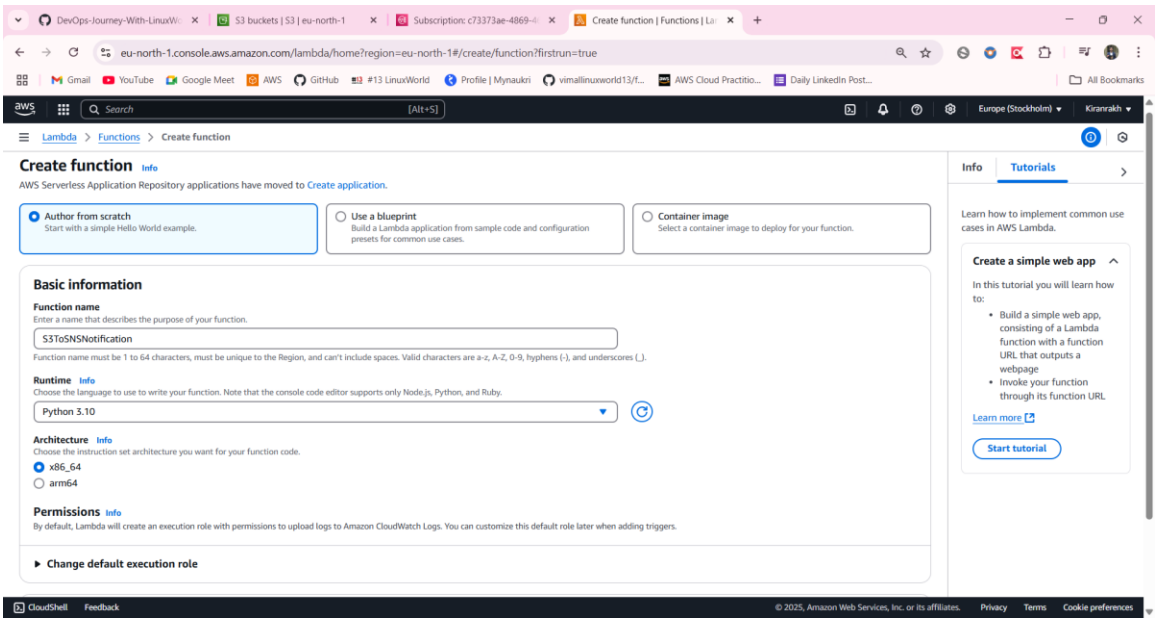


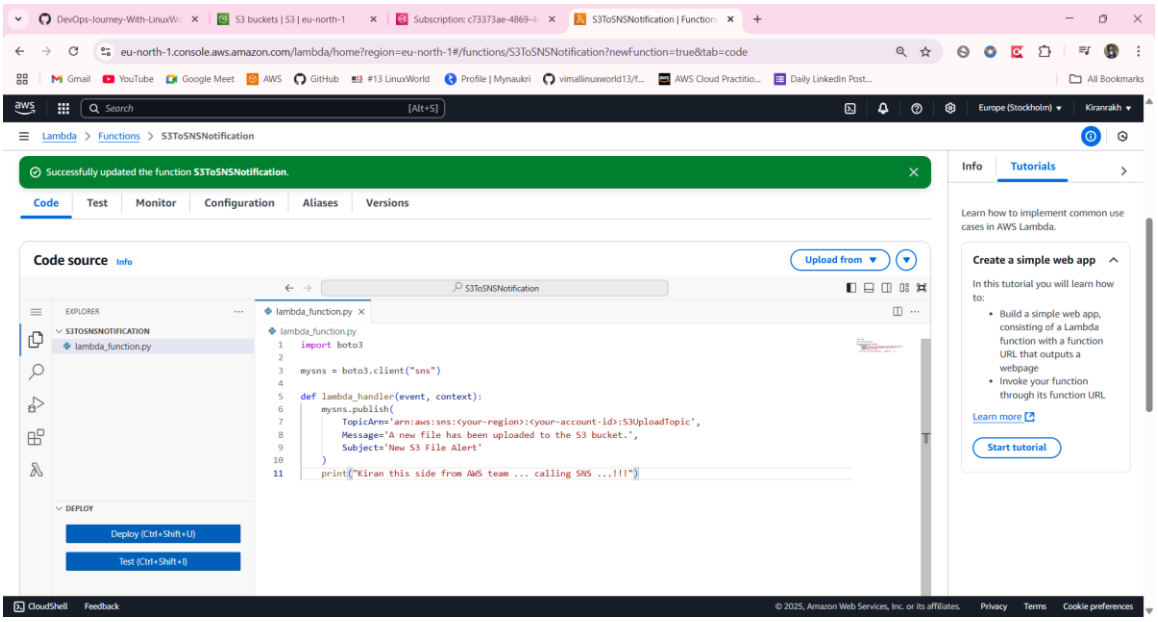

Inbox (33) - kiranlearn22@gma
AWS Notification - Subscription
Subscription confirm
sns.eu-north-1.amazonaws.com/confirmation.html?TopicArn=arn:aws:sns:eu-north-1:564599437276:S3UploadTopic&Token=2336412f37fb687f5d51e6e2425a8a5874c1b18545ab810dd4d3099be5a83c7...
aws
Simple Notification Service
Subscription confirmed!
You have successfully subscribed.
Your subscription's id is:
arn:aws:sns:eu-north-1:564599437276:S3UploadTopic:c73373ae-4869-46cb-acfb-c76645796d2e
If it was not your intention to subscribe, click here to unsubscribe.



3 LAMBDA: CREATE NEW FUNCTION

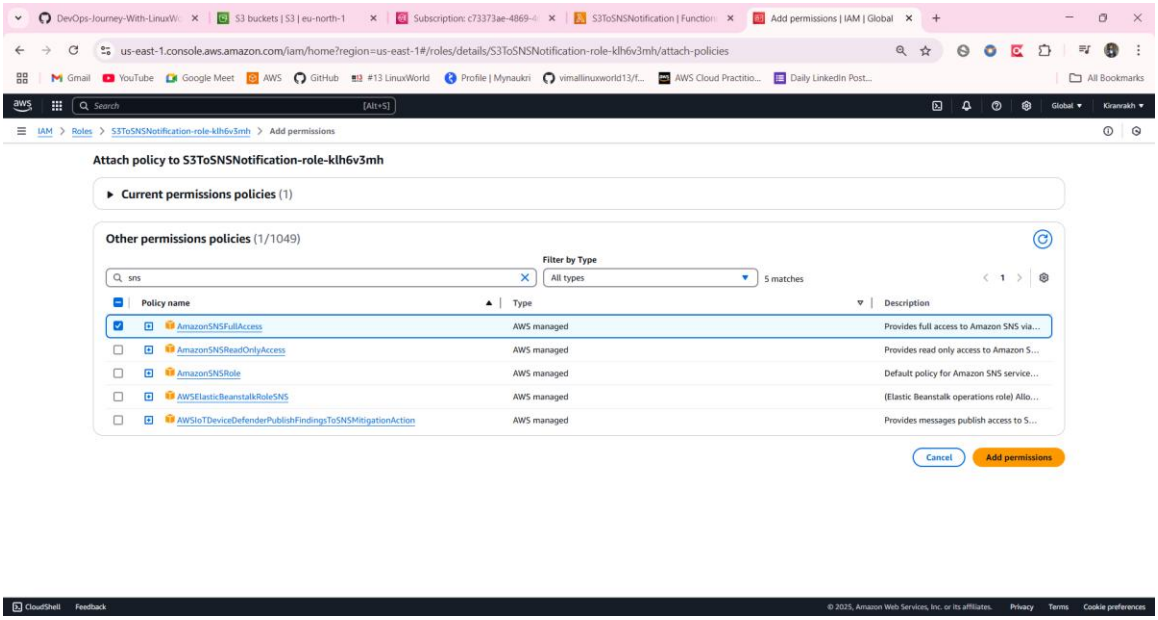
- 1. Go to Services → Lambda → Create function
- 2. Name: S3ToSNSNotification, Runtime: Python 3.x
- 3. Click “Create function”
- Lambda function code and configuration:





4 IAM: PERMISSION SETUP FOR LAMBDA

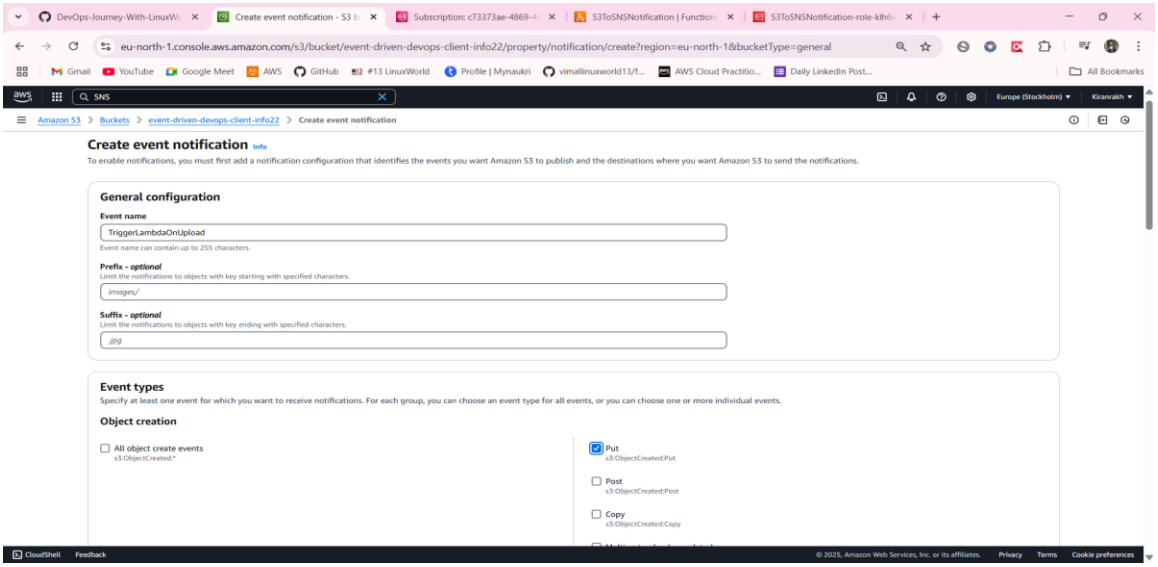
- 1. Go to Lambda → Select your function → Configuration → Permissions
- 2. Click the role name → Add permissions → Attach policies
- 3. Search for and attach 'AmazonSNSFullAccess'
- IAM role and policy attachment:



5 LINK S3 TO LAMBDA (EVENT NOTIFICATION)

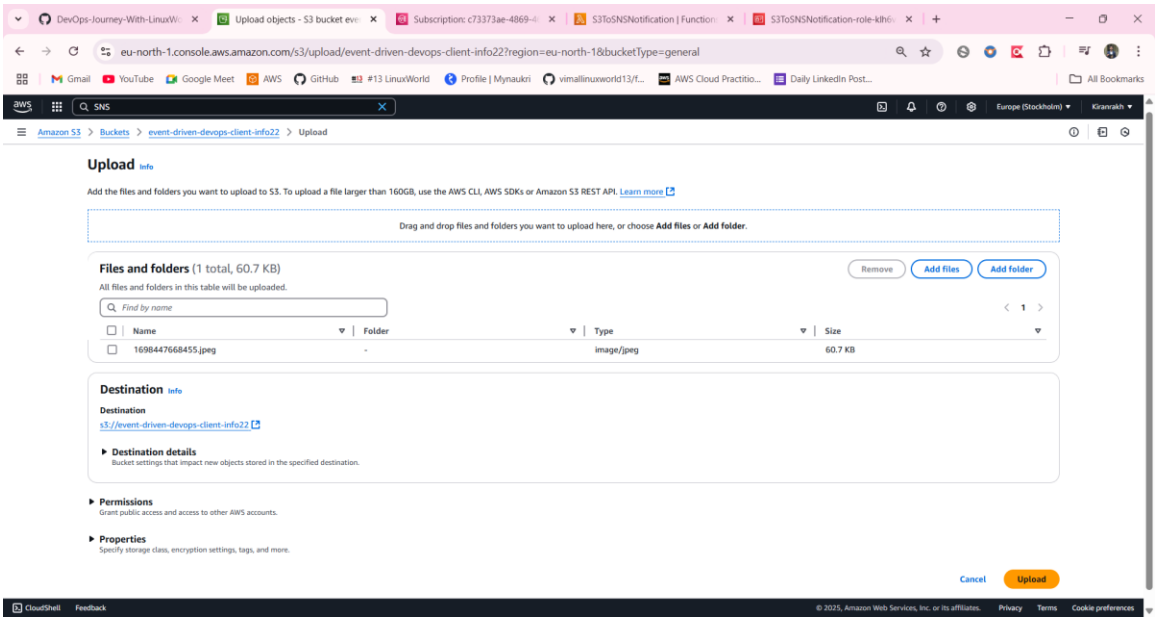
- 1. Go to S3 → Your bucket → Properties → Event notifications
- 2. Create event notification → Name: TriggerLambdaOnUpload
- 3. Event type: PUT (Object Created)
- 4. Destination: Lambda Function → Choose: S3ToSNSNotification

- 5. Save configuration || S3 event notification setup:



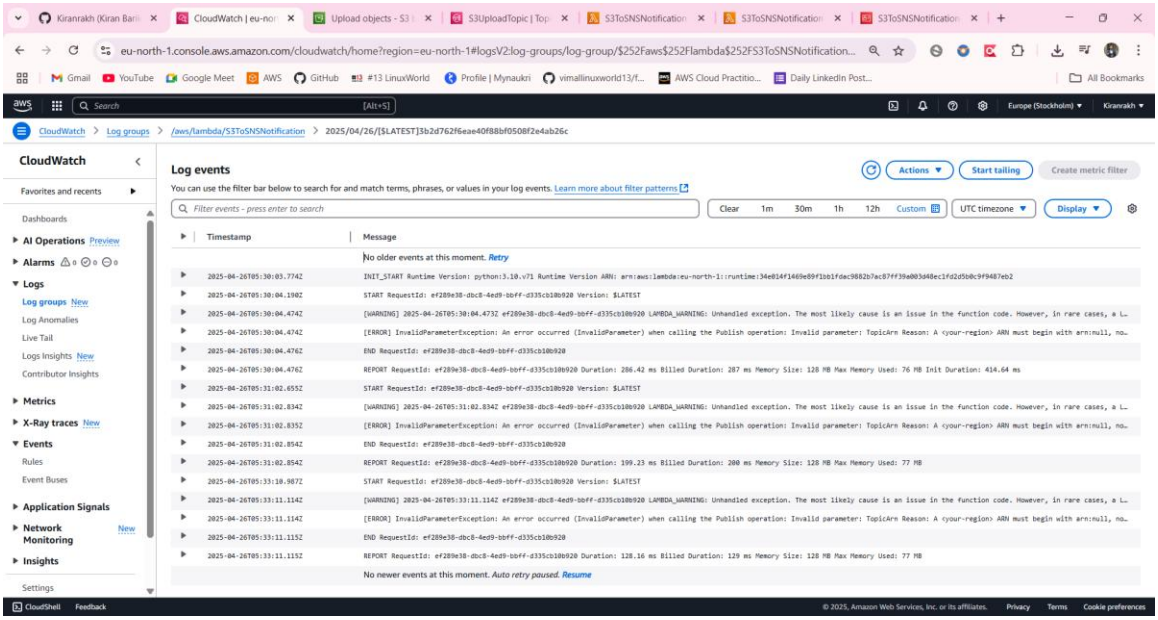
6 TEST IT

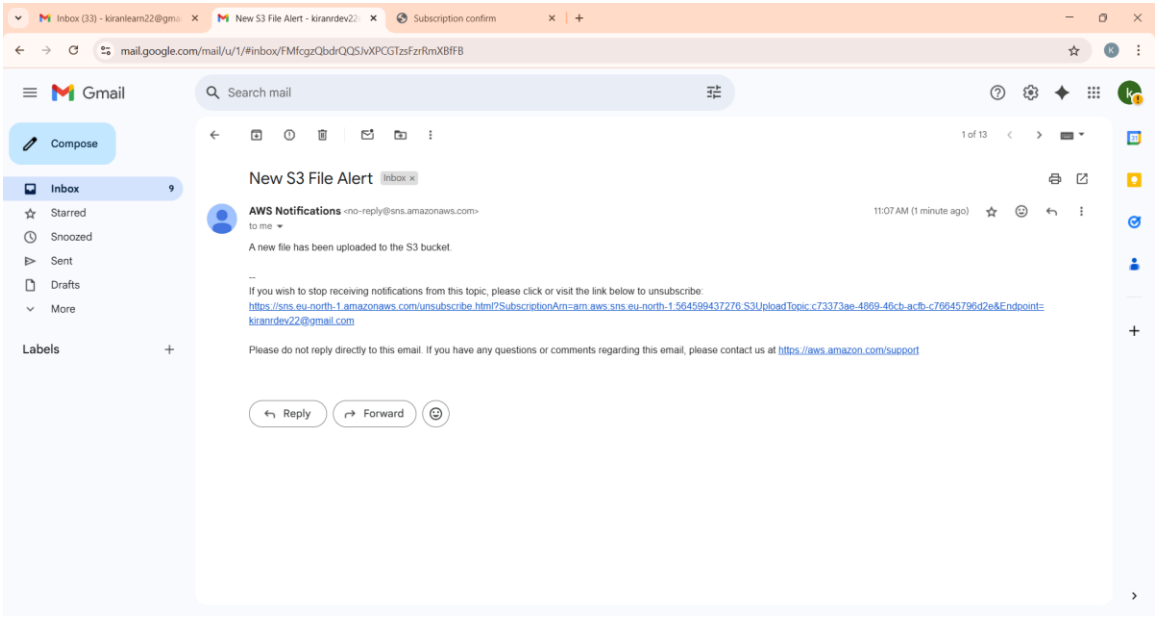
- 1. Go to S3 → Your bucket → Upload any file (e.g., test3.txt)
- 2. Lambda should be triggered automatically
- File upload to S3:



7 VERIFY IN CLOUDWATCH

- 1. Go to Services → CloudWatch → Log groups
- 2. Find log group for S3ToSNSNotification → Open logs
- 3. Check print statements and SNS execution logs
- CloudWatch logs output:





FINAL USE CASE

- ✔ Automatic email notifications for every file uploaded to S3
- ✔ Use cases: automation pipelines, audit alerts, monitoring
- ✔ Fully managed serverless stack using AWS-native services

EXTRAS / CUSTOMIZATION TIPS

- - You can rename function, bucket, topic, and handler
- - Edit handler in Lambda if function name is changed
- - Customize message in Lambda code
- - Keep all resources in the same region

IAM ROLES & KEYS

- - Lambda role must have AmazonSNSFullAccess
- - S3 automatically gets permission to trigger Lambda during setup
- - No access keys required due to IAM roles