## Lab 6-01: Serverless AI Workflow Automation

|  |
| --- |
| **Introduction**  Serverless AI workflow automation in AWS allows you to connect multiple AI services without managing servers. You can use AWS Lambda to run code automatically when new data arrives, and connect it with services like Amazon S3, Amazon Comprehend, or Amazon Rekognition to analyze or process that data. This saves time, reduces manual effort, and is cost-effective since you only pay for what you use.  **Challenge**  Imagine your company receives hundreds of text files daily from customers containing product feedback. It takes hours for employees to read them and identify whether the feedback is positive, negative, or neutral. You decide to automate this process using AWS services. Every time a text file is uploaded to an S3 bucket, a Lambda function will automatically run and use Amazon Comprehend to analyze the sentiment of the text and store the result in another S3 bucket.  **Lab Diagram**  **C:\Users\Binary Computers\Downloads\9807c95a-8956-4476-9513-9a702473deb5.jpg**  **Solution**  **Step 1: Open the AWS Management Console**   1. Go to <https://aws.amazon.com>. 2. Sign in to your AWS account with your credentials. 3. In the search bar, type S3 and open the Amazon S3 service.     **Step 2: Create an Input Bucket**   1. Click Create bucket.      1. Name it something like text-feedback-input.      1. Keep all default settings and click Create bucket.      1. This bucket will store the incoming text files.     **Step 3: Create an Output Bucket**   1. Click Create bucket again.      1. Name it text-feedback-output.      1. Keep the default settings and create it.      1. This bucket will store sentiment-analysis results.     **Step 4: Open AWS Lambda**   1. In the AWS Console search bar, type Lambda.      1. Choose the Create function.      1. Click Author from scratch.      1. Function name: TextFeedbackAnalyzer.      1. Runtime: Python 3.12 (or latest).      1. Click the Create function.     **Step 5: Add the Code**   1. In the Code source section, delete the default code and copy and paste this simple example:   **import json**  **import boto3**  **comprehend = boto3.client('comprehend')**  **s3 = boto3.client('s3')**  **def lambda\_handler(event, context):**  **bucket\_name = event['Records'][0]['s3']['bucket']['name']**  **file\_name = event['Records'][0]['s3']['object']['key']**  **text\_file = s3.get\_object(Bucket=bucket\_name, Key=file\_name)**  **text\_data = text\_file['Body'].read().decode('utf-8')**  **response = comprehend.detect\_sentiment(Text=text\_data, LanguageCode='en')**  **sentiment = response['Sentiment']**  **result = {'File': file\_name, 'Sentiment': sentiment}**  **s3.put\_object(**  **Bucket='text-feedback-output',**  **Key=f"{file\_name}\_result.json",**  **Body=json.dumps(result)**  **)**  **return {"status": "Success", "Sentiment": sentiment}**     1. Click Deploy to save your code.     **Step 6: Add an S3 Trigger**   1. Scroll up to Function overview → click + Add trigger.      1. Choose S3 as the trigger source.      1. Select your text-feedback-input bucket.      1. Event type: All object create events.      1. Check Enable trigger, then click Add.     **Step 7: Test the Workflow**   1. Go back to the S3 Console → open your text-feedback-input bucket.      1. Upload a text file (e.g., feedback1.txt) containing a short message like:   **I love this product! It works perfectly.**     1. Wait a few seconds.      1. Open your text-feedback-output bucket.      1. You will see a new file, such as feedback1.txt\_result.json.      1. Open it, and it will show something like:   **{"File": "feedback1.txt", "Sentiment": "POSITIVE"}**    **Note**   * You have successfully built a serverless AI workflow using S3 + Lambda + Comprehend |