**Task 1:** How to convert excel file to csv file from one s3 bucket to another s3 bucket using aws lambda.

To convert an Excel file to a CSV file from one S3 bucket to another S3 bucket using AWS Lambda with Python, you can use the **pandas** library to handle the conversion. Here's a step-by-step guide to achieving this:

1. **Setup AWS Lambda Environment**:
   * Create a new Lambda function in the AWS Management Console or use an existing one.
   * Ensure that the Lambda function has appropriate IAM permissions to read from the source S3 bucket and write to the destination S3 bucket.
2. **Install Required Libraries**:
   * Install the **pandas** library along with other dependencies locally. You'll need to package these dependencies with your Lambda function.
3. **Write Lambda Function Code**:
   * Write the Python code for your Lambda function to convert the Excel file to a CSV file.
4. **Package Dependencies**:
   * Package your Lambda function along with its dependencies (including **pandas**) into a ZIP file. You can do this using tools like **pip** or by manually including the necessary files.
5. **Upload ZIP to Lambda**:
   * Upload the ZIP file containing your Lambda function code and dependencies to AWS Lambda.
6. **Configure Lambda Triggers**:
   * Configure the Lambda function to trigger upon an event in the source S3 bucket (e.g., new Excel file upload).
7. **Test Lambda Function**:
   * Test the Lambda function by uploading an Excel file to the source S3 bucket and verifying that it converts to a CSV file in the destination S3 bucket.

Here's an example Lambda function code to convert an Excel file to a CSV file and move it from one S3 bucket to another:

**Code:::::::::**

import boto3

import pandas as pd

import io

def lambda\_handler(event, context):

# Initialize S3 client

s3 = boto3.client('s3')

# Extract source and destination bucket and key from the event

source\_bucket = event['Records'][0]['s3']['bucket']['name']

source\_key = event['Records'][0]['s3']['object']['key']

destination\_bucket = 'destination\_bucket\_name'

destination\_key = source\_key.replace('.xlsx', '.csv') # Replace file extension with .csv

# Download Excel file from source bucket

response = s3.get\_object(Bucket=source\_bucket, Key=source\_key)

excel\_data = response['Body'].read()

# Convert Excel data to CSV

excel\_df = pd.read\_excel(io.BytesIO(excel\_data))

csv\_buffer = io.StringIO()

excel\_df.to\_csv(csv\_buffer, index=False)

csv\_data = csv\_buffer.getvalue()

# Upload CSV file to destination bucket

s3.put\_object(Bucket=destination\_bucket, Key=destination\_key, Body=csv\_data)

return {

'statusCode': 200,

'body': f'Excel file "{source\_key}" converted to CSV and moved to "{destination\_key}"'

}

**:::::::::**

Json test code:

{

"Records": [

{

"eventVersion": "2.1",

"eventSource": "aws:s3",

"awsRegion": "us-east-1",

"eventName": "ObjectCreated:Put",

"s3": {

"bucket": {

"name": "shyam-source-bucket-1"

},

"object": {

"key": "example.xlsx"

}

}

}

]

}