Project Documentation: Lambda Function with S3 Trigger

# 1. Introduction

This project involves creating two S3 buckets, setting up an IAM role, and writing a Lambda function that is triggered by file uploads to the source bucket. The Lambda function processes the file, aggregates data, and saves it to the destination bucket.

# 2. Step 1: Creating S3 Buckets

In this step, we create two S3 buckets: 'sourceforgithubproject' for storing the source files and 'destinationforgithubproject' for storing the aggregated results.

Steps to create S3 buckets:

1.Go to the S3 Console in AWS.  
2. Click on 'Create Bucket'.  
3. For the first bucket, name it 'sourceforgithubproject' and set the region.  
4. For the second bucket, name it 'destinationforgithubproject' and set the region.  
5. Ensure that both buckets are created without versioning or logging enabled.

A screenshot of a computer

Description automatically generated

# 3. Step 2: Creating IAM Role

In this step, we create an IAM role with the necessary permissions for Lambda to access the S3 buckets.

Steps to create IAM Role:

1. Go to the IAM Console in AWS.  
2. Select 'Roles' and click 'Create role'.  
3. Choose 'Lambda' as the trusted entity.  
4. Attach the 'AWSLambdaBasicExecutionRole' and 'AmazonS3FullAccess' policies.  
5. Name the role 's3-lambda-cloudwatch-role' and create it.

A screenshot of a computer

Description automatically generated

# 4. Step 3: Creating the Lambda Function

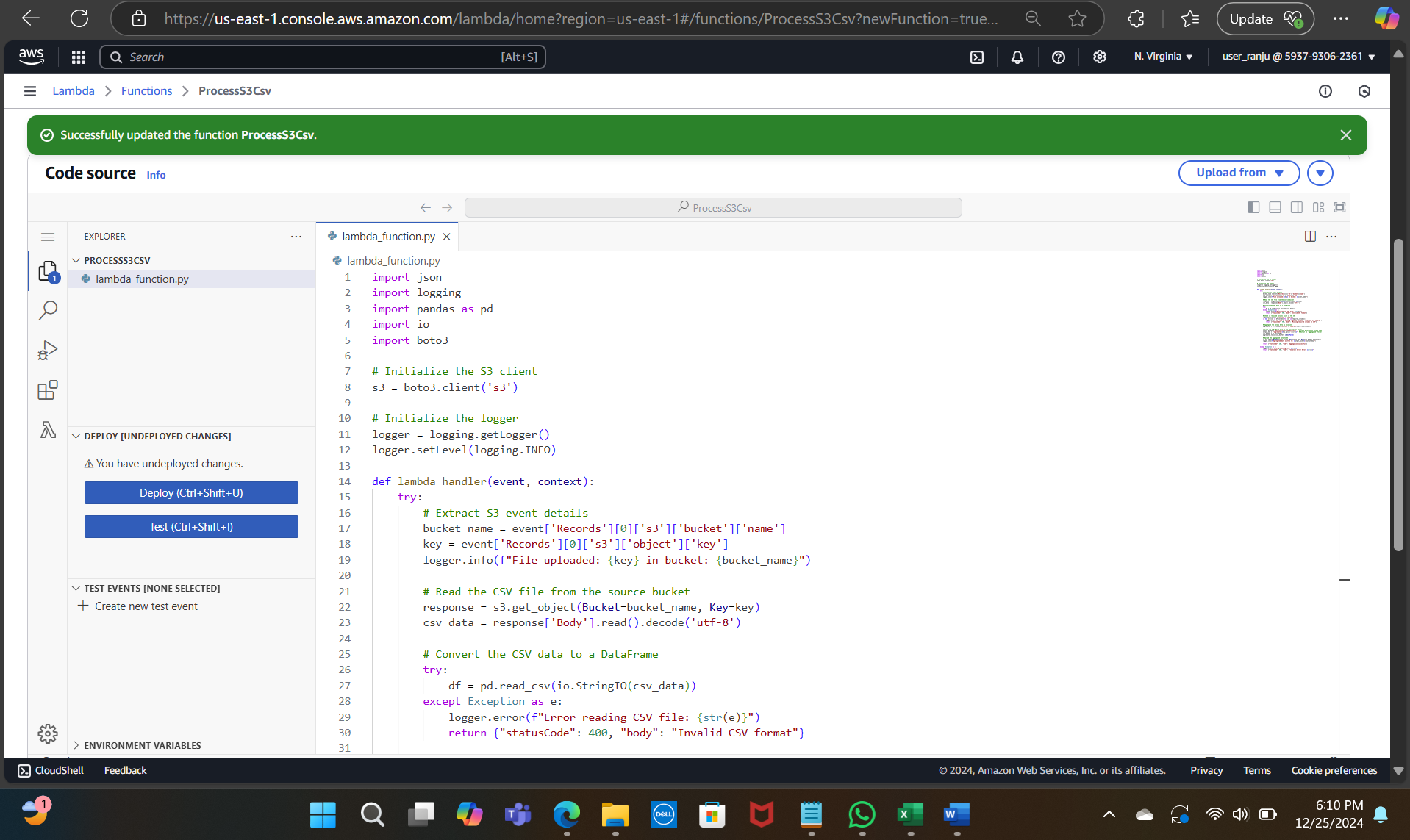
This step involves creating the Lambda function, which will be triggered by the upload of files to the 'sourceforgithubproject' S3 bucket. The function processes the data and writes the results to the 'destinationforgithubproject' S3 bucket.

Steps to create Lambda Function:

1.Go to the Lambda Console in AWS.  
2. Click 'Create function'.  
3. Choose 'Author from Scratch' and set the function name, such as 'ProcessS3Csv'.  
4. Select 'Python 3.8' or higher as the runtime.  
5. Choose the IAM role created in the previous step (' s3-lambda-cloudwatch-role' ').

A screenshot of a computer

Description automatically generated  
6. In the 'Function code' section, write the Python code for processing the CSV files:



7. After pasting the code, click 'Deploy' to save and deploy the Lambda function.

# 5. Step 4: Configuring S3 Event Notification to Trigger Lambda

Instead of configuring the trigger directly in the Lambda function's configuration, we used S3 event notifications to invoke the Lambda function. Here’s how to set it up:

Steps:

1. Navigate to S3 Console: Go to the S3 service and open the sourceforgithubproject bucket.

2. Go to Properties: Under the bucket, select Properties.

3. Set Up Event Notifications:

* + Scroll down to Event notifications and click Create event notification.
  + Name: NewFileUploadTrigger.
  + Event types: All object create events.
  + Destination: Choose Lambda function.

1. **Choose Lambda Function**: Select the Lambda function that processes the file from the dropdown.
2. **Save Configuration**: Click **Save** to finish the setup.

A screenshot of a computer

Description automatically generated

# 5. Uploading CSV File to the Source Bucket

# The next step was to upload the CSV file githubprojectdata\_employee.csv to the sourceforgithubproject bucket.

A screenshot of a computer

Description automatically generated

**The Lambda function was automatically triggered upon the upload of this file.**

**6. Verifying Lambda Function Execution in CloudWatch Logs**

After the file upload, we verified the execution of the Lambda function by checking CloudWatch Logs. The log confirmed that the Lambda function executed successfully, without any errors, and aggregated the data as expected.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# 7. Verifying Output in Destination Bucket

# Upon confirming successful execution, we navigated to the destinationforgithubproject bucket.We found the aggregated/ folder containing the processed file githubprojectdata\_employee.csv.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# 8. Downloading the Aggregated CSV File

# We downloaded the aggregated CSV file from the destinationforgithubproject/aggregated/ folder for further review and validation.

A screenshot of a computer

Description automatically generated

# 9. Conclusion

By following the steps above, we have successfully created an S3-triggered Lambda function that processes CSV files, aggregates data by location, and stores the results in the destination S3 bucket. This solution can be scaled to process multiple files as needed.