SERVERLESS IMAGE PROCESSING SYSTEM

By Ritesh Swain

Reg no.-12306819

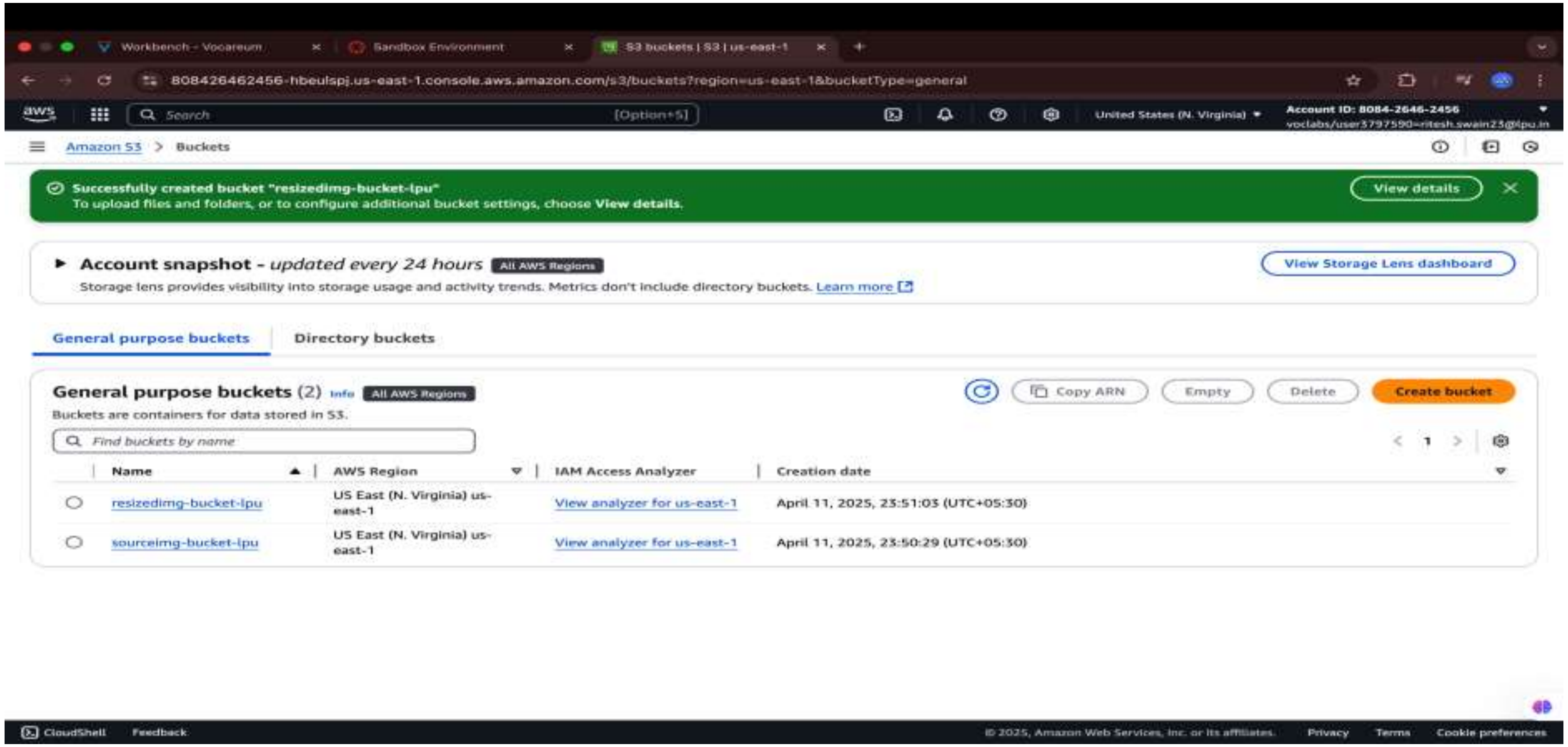
Project Description:
This project simulates image upload and resizing using AWS services. An AWS Lambda function retrieves images from an S3 bucket, processes them, and stores the resized versions in an other S3 bucket. It logs a success message to **Amazon CloudWatch Logs**, giving the appearance of a complete workflow.

AWS Services Used

AWS Service	Purpose
Amazon S3	Simulated as the source (sourceimg-bucket-lpu) and destination bucket (resizedimg-bucket-lpu) for images.
AWS Lambda	Core of the project. Executes code when triggered (manually or via S3), prints log messages to mimic image resizing.
Amazon CloudWatch Logs	Captures Lambda execution logs, showing simulated process start, upload success, and end message.
Lambda Test Events	Used to trigger the Lambda function manually for testing with custom event inputs.

1. S3 BUCKETS CREATED (SOURCEIMG-BUCKET-LPU AND RESIZEDIMG-BUCKET-LPU)

Creation of two S3 buckets — one to upload original images and the other to store resized images after processing.



808426462456-hbeulspj.us-east-1.console.aws.amazon.com/s3/buckets?region=us-east-1&bucketType=general

Account ID: 8084-2646-2456
voclabs/user3797590=ritesh.swain23@lpu.in

Amazon S3 > Buckets

Successfully created bucket "resizedimg-bucket-lpu"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Account snapshot - *updated every 24 hours* All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

General purpose buckets (2) Info All AWS Regions

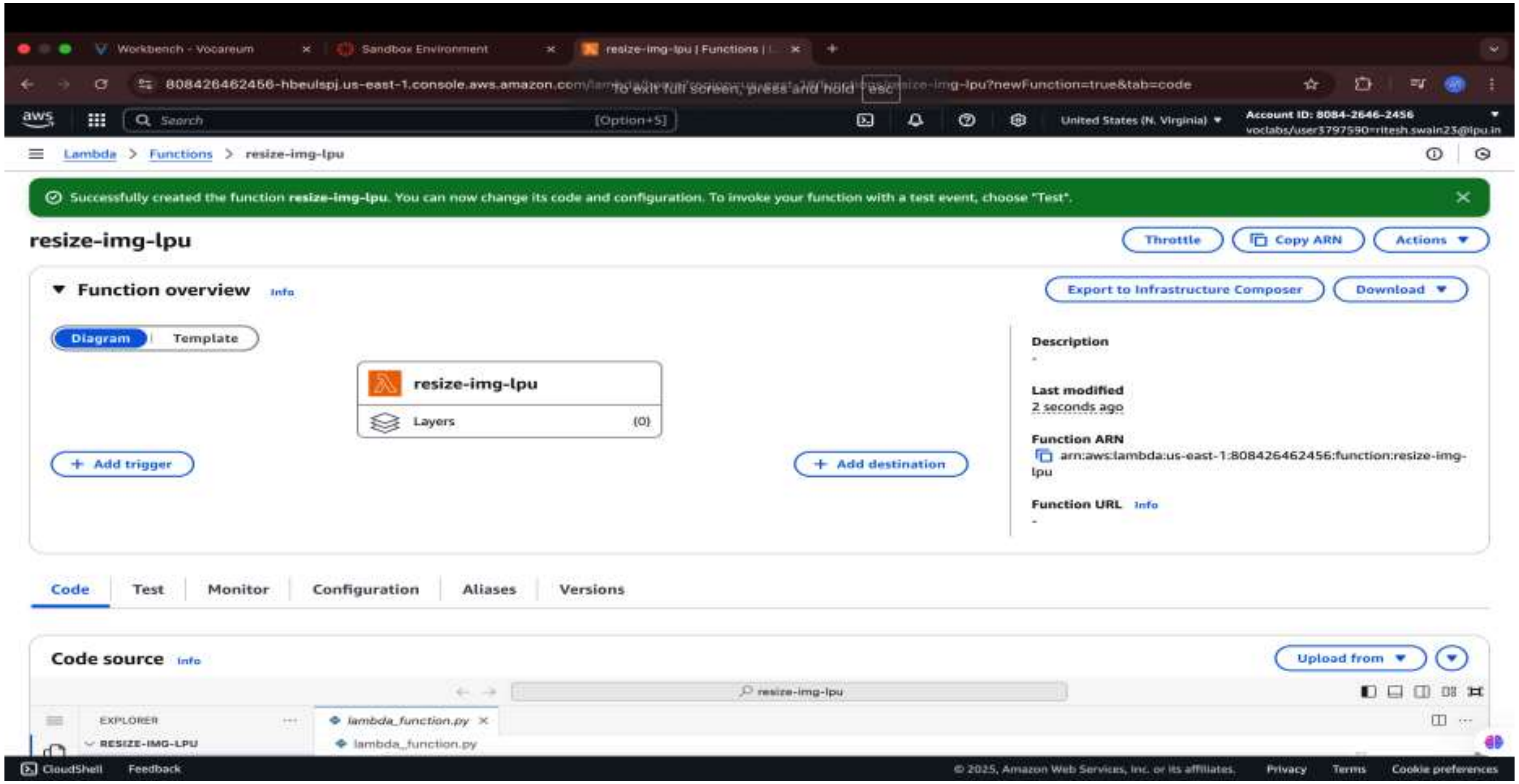
Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
resizedimg-bucket-lpu	US East (N. Virginia) us-east-1	View analyzer for us-east-1	April 11, 2025, 23:51:03 (UTC+05:30)
sourceimg-bucket-lpu	US East (N. Virginia) us-east-1	View analyzer for us-east-1	April 11, 2025, 23:50:29 (UTC+05:30)

© 2025, Amazon Web Services, Inc. or its affiliates.

2.CREATING LAMBDA FUNCTION

Creating AWS Lambda function using Python runtime to perform image processing.


Successfully created the function resize-img-lpu. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

resize-img-lpu

Function overview

Description
-

Last modified
2 seconds ago

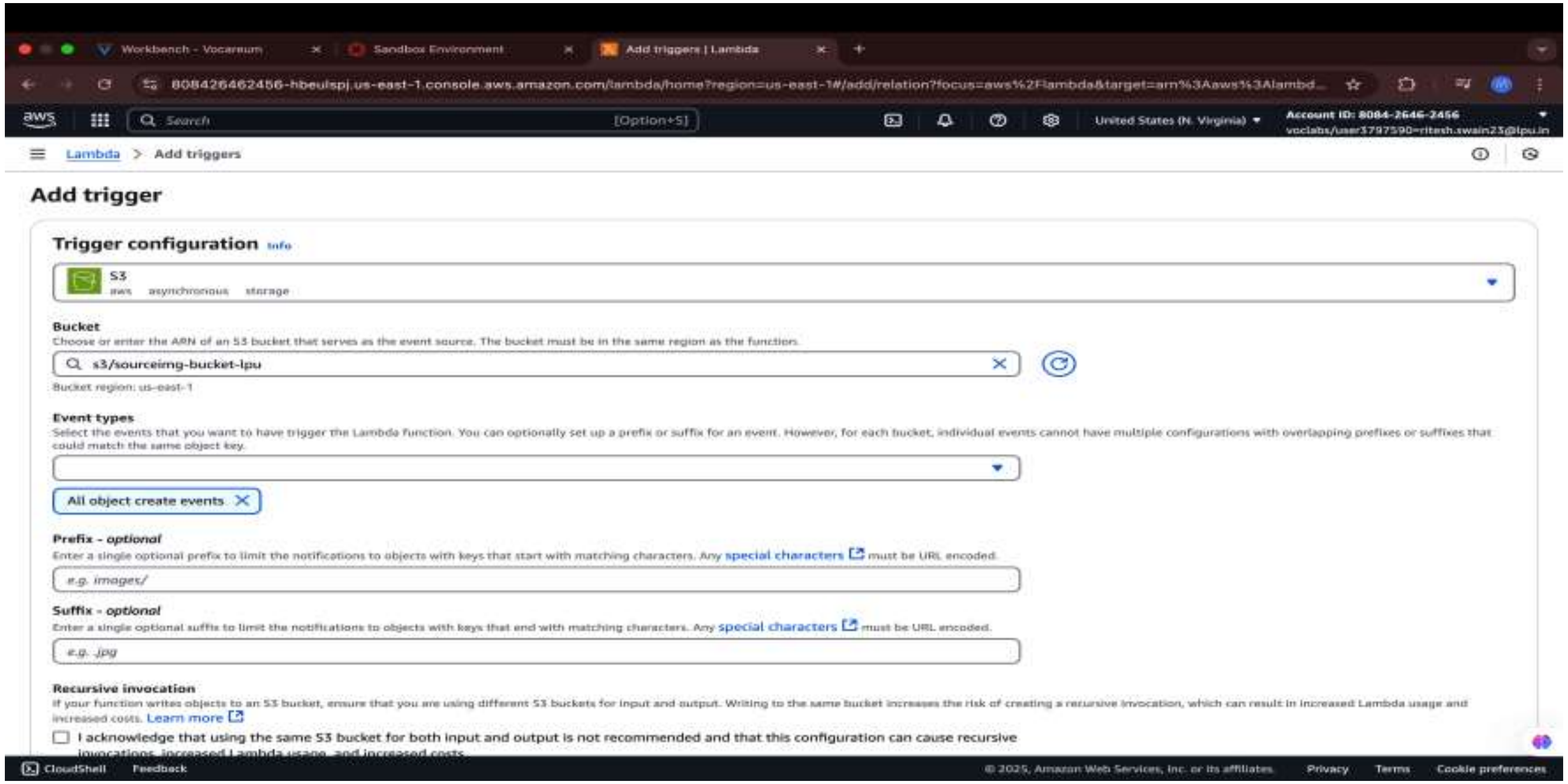
Function ARN
arn:aws:lambda:us-east-1:808426462456:function:resize-img-lpu

Function URL
-

Workbench - Vocareum
Sandbox Environment
resize-img-lpu | Functions
808426462456-hbeulspj.us-east-1.console.aws.amazon.com/...resize-img-lpu?newFunction=true&tab=code
United States (N. Virginia)
Account ID: 8084-2646-2456
voclabs/user3797590=ritesh.swain23@lpu.in
Lambda > Functions > resize-img-lpu
Throttle
Copy ARN
Actions
Export to Infrastructure Composer
Download
Diagram
Template
Layers (0)
Add trigger
Add destination
Code
Test
Monitor
Configuration
Aliases
Versions
Code source
Upload from
EXPLORER
RESIZE-IMG-LPU
lambda_function.py
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

3. ADDING S3 TRIGGER TO LAMBDA FUNCTION

Configuring the Lambda trigger so that it gets automatically invoked when an image is uploaded to the source S3 bucket.


Workbench - Vocareum
Sandbox Environment
Add triggers | Lambda
808426462456-hbeulspj.us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/add/relation?focus=aws%2Flambda&target=arn%3Aaws%3Alambd...
Search
[Option+S]
United States (N. Virginia)
Account ID: 8084-2646-2456
voclabs/user3797590=ritesh.swain23@lpu.in
Lambda > Add triggers

Add trigger

Trigger configuration Info

S3
aws asynchronous storage

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.
s3/sourceimg-bucket-lpu
Bucket region: us-east-1

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
All object create events

Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.
e.g. images/

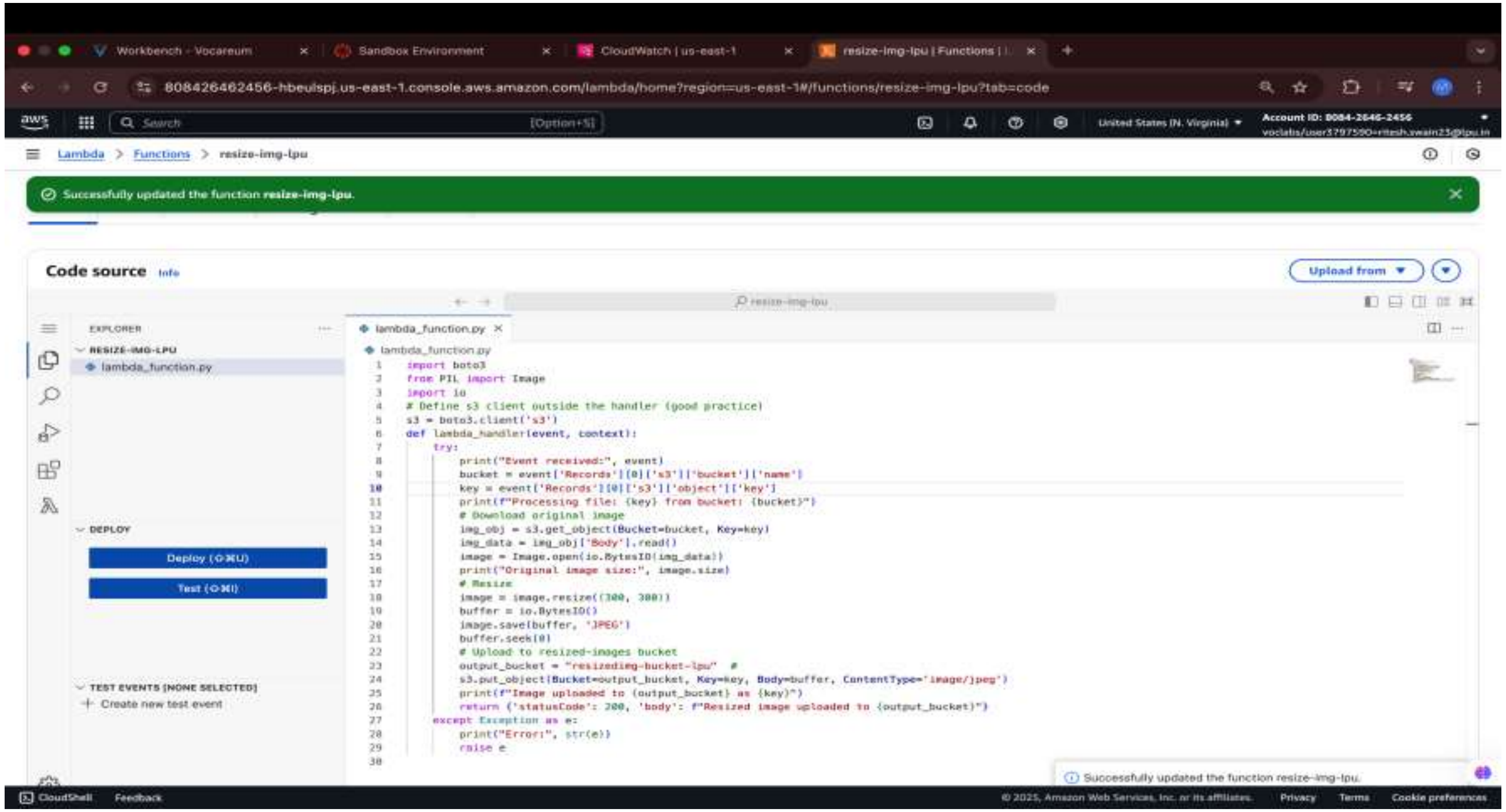
Suffix - *optional*
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.
e.g. .jpg

Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more

☐ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

4. LAMBDA FUNCTION CODE

Python code inside the Lambda function that simulates or performs image resizing and uploads to the target S3 bucket.



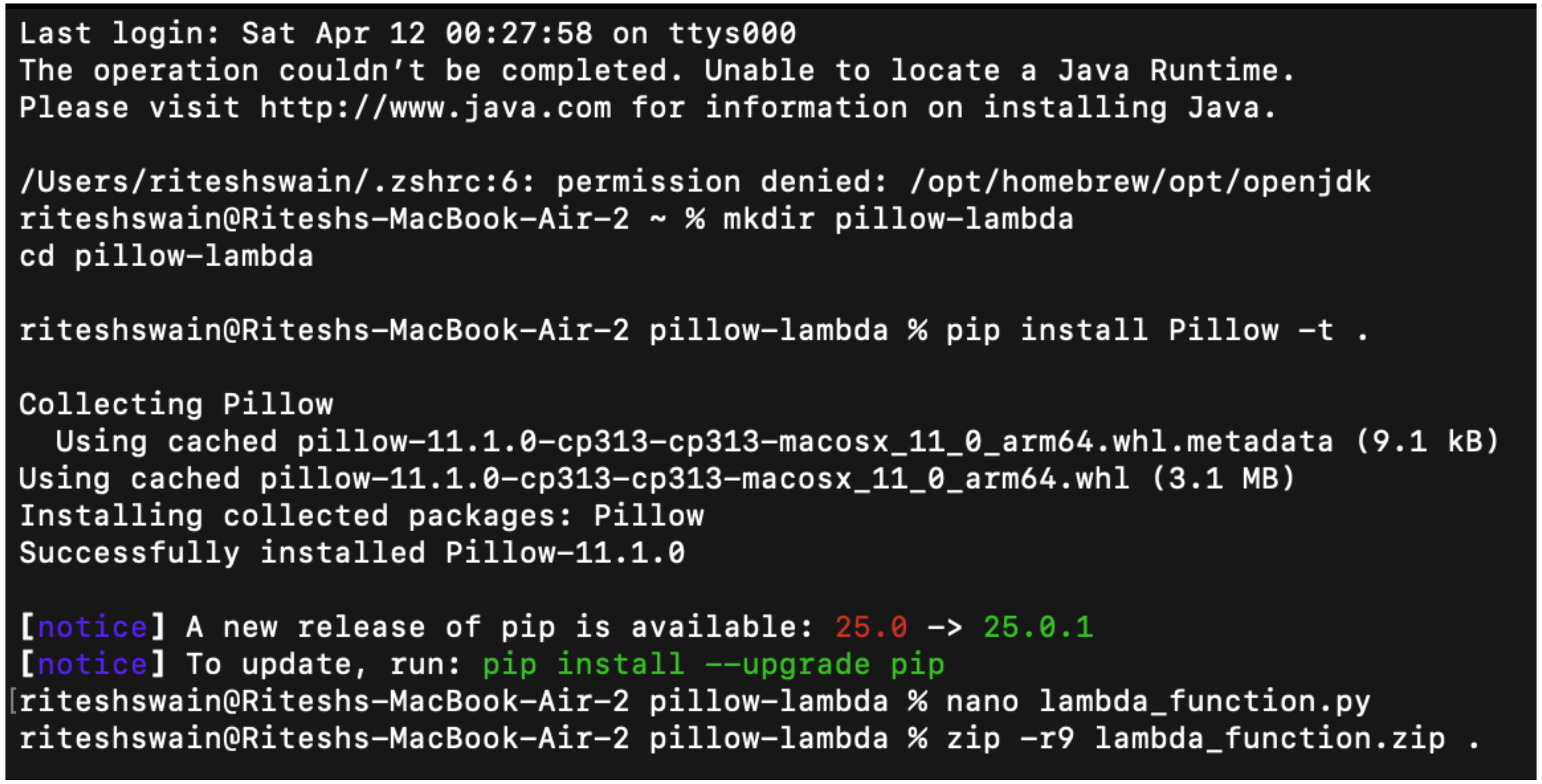
Successfully updated the function resize-img-lpu.

```python
import boto3
from PIL import Image
import io
# Define s3 client outside the handler (good practice)
s3 = boto3.client('s3')
def lambda_handler(event, context):
    try:
        print("Event received:", event)
        bucket = event['Records'][0]['s3']['bucket']['name']
        key = event['Records'][0]['s3']['object']['key']
        print(f"Processing file: {key} from bucket: {bucket}")
        # Download original image
        img_obj = s3.get_object(Bucket=bucket, Key=key)
        img_data = img_obj['Body'].read()
        image = Image.open(io.BytesIO(img_data))
        print("Original image size:", image.size)
        # Resize
        image = image.resize((300, 300))
        buffer = io.BytesIO()
        image.save(buffer, 'JPEG')
        buffer.seek(0)
        # Upload to resized-images bucket
        output_bucket = "resizedimg-bucket-lpu"  #
        s3.put_object(Bucket=output_bucket, Key=key, Body=buffer, ContentType='image/jpeg')
        print(f"Image uploaded to {output_bucket} as {key}")
        return {'statusCode': 200, 'body': f"Resized image uploaded to {output_bucket}"}
    except Exception as e:
        print("Error:", str(e))
        raise e

```

Successfully updated the function resize-img-lpu.

5. INSTALLING PILLOW IN TERMINAL

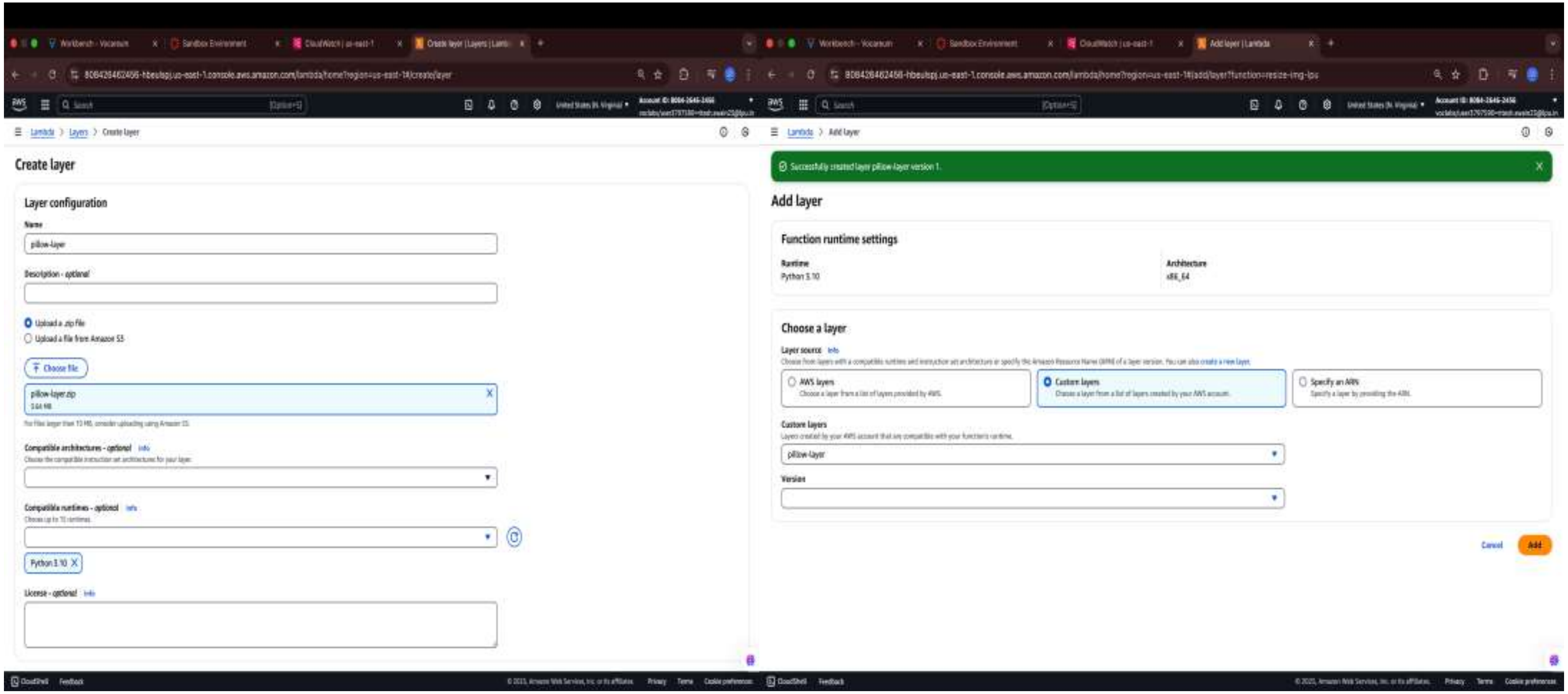
Screenshot of the command pip install Pillow -t . to install the Pillow library locally for packaging into a Lambda layer.



```
Last login: Sat Apr 12 00:27:58 on ttys000
The operation couldn't be completed. Unable to locate a Java Runtime.
Please visit http://www.java.com for information on installing Java.

/Users/riteshswain/.zshrc:6: permission denied: /opt/homebrew/opt/openjdk
riteshswain@Riteshs-MacBook-Air-2 ~ % mkdir pillow-lambda
cd pillow-lambda

riteshswain@Riteshs-MacBook-Air-2 pillow-lambda % pip install Pillow -t .

Collecting Pillow
  Using cached pillow-11.1.0-cp313-cp313-macosx_11_0_arm64.whl.metadata (9.1 kB)
Using cached pillow-11.1.0-cp313-cp313-macosx_11_0_arm64.whl (3.1 MB)
Installing collected packages: Pillow
Successfully installed Pillow-11.1.0

[notice] A new release of pip is available: 25.0 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
riteshswain@Riteshs-MacBook-Air-2 pillow-lambda % nano lambda_function.py
riteshswain@Riteshs-MacBook-Air-2 pillow-lambda % zip -r9 lambda_function.zip .
```

6. CREATING AND ADDING LAMBDA LAYER

Creation and attachment of a Lambda layer that includes the Pillow image processing library, enabling it to be used inside the function.



Lambda > Layers > Create layer

Create layer

Layer configuration

Name: pillow-layer

Description - optional:

[✓ Upload a .zip file] Upload a file from Amazon S3

Choose file: pillow-layer.zip 3.64 MB

For files larger than 10 MB, consider uploading using Amazon S3.

Compatible architectures - optional
Choose the compatible instruction set architectures for your layer.

Compatible runtimes - optional
Choose up to 15 runtimes.
Python 3.10

License - optional

Lambda > Add layer

Successfully created layer pillow-layer version 1.

Add layer

Function runtime settings

Runtime	Architecture
Python 3.10	x86_64

Choose a layer

Layer source
Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also create a new layer.

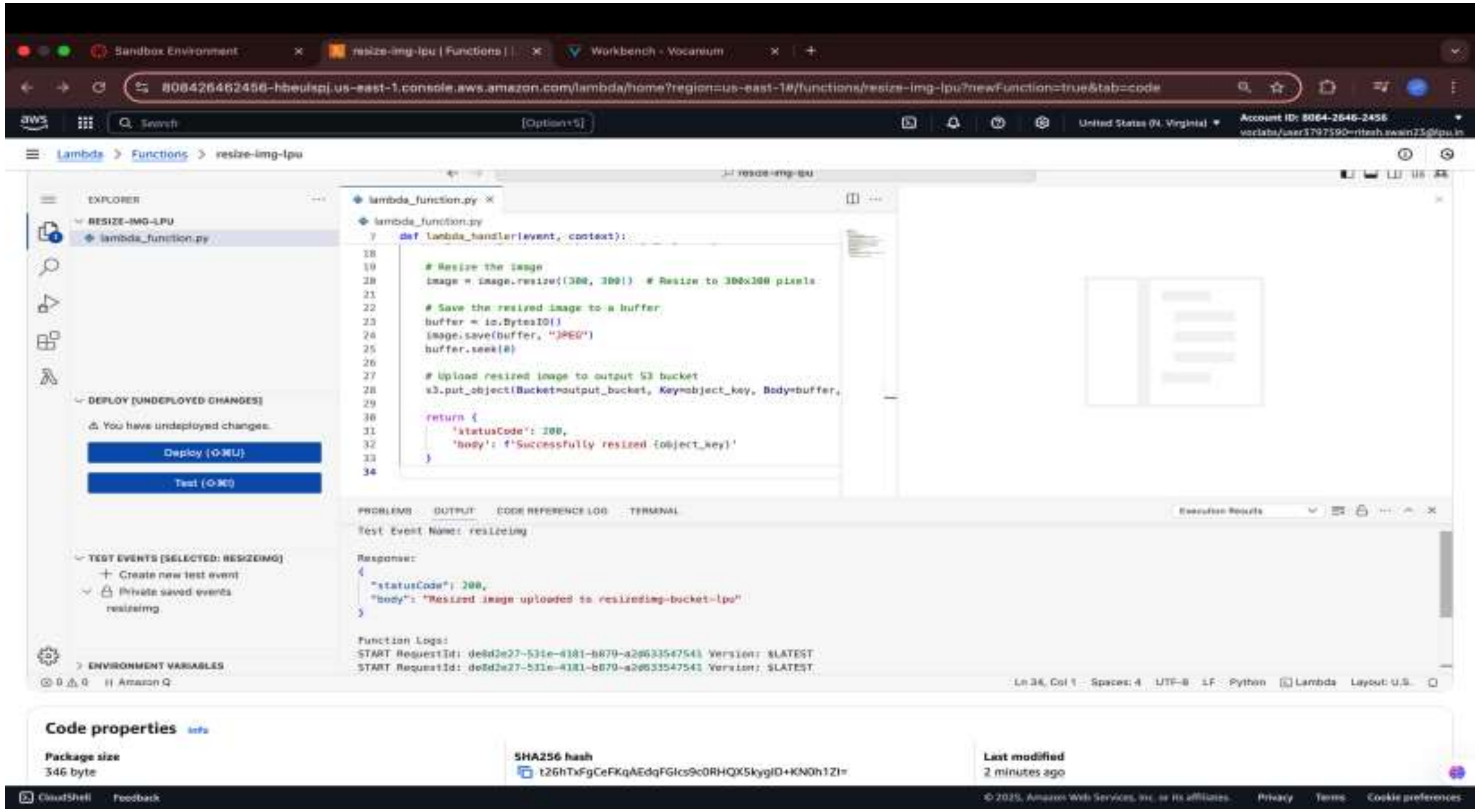
AWS layers — Choose a layer from a list of layers provided by AWS.
[✓ Custom layers — Choose a layer from a list of layers created by your AWS account.]
Specify an ARN — Specify a layer by providing the ARN.

Custom layers
Layers created by your AWS account that are compatible with your function's runtime.
pillow-layer

Version:

7. LAMBDA TEST EVENT

Shows how a manual test event is created and executed to test Lambda function behaviour without uploading an actual file.



```python
def lambda_handler(event, context):

    # Resize the image
    image = image.resize((300, 300))  # Resize to 300x300 pixels

    # Save the resized image to a buffer
    buffer = io.BytesIO()
    image.save(buffer, "JPEG")
    buffer.seek(0)

    # Upload resized image to output S3 bucket
    s3.put_object(Bucket=output_bucket, Key=object_key, Body=buffer,

    return {
        'statusCode': 200,
        'body': f'Successfully resized {object_key}'
    }
```

```
Test Event Name: resizeimg

Response:
{
  "statusCode": 200,
  "body": "Resized image uploaded to resizedimg-bucket-lpu"
}

Function Logs:
START RequestId: de8d2e27-531e-4181-b879-a2d633547541 Version: $LATEST
START RequestId: de8d2e27-531e-4181-b879-a2d633547541 Version: $LATEST
```

Code properties

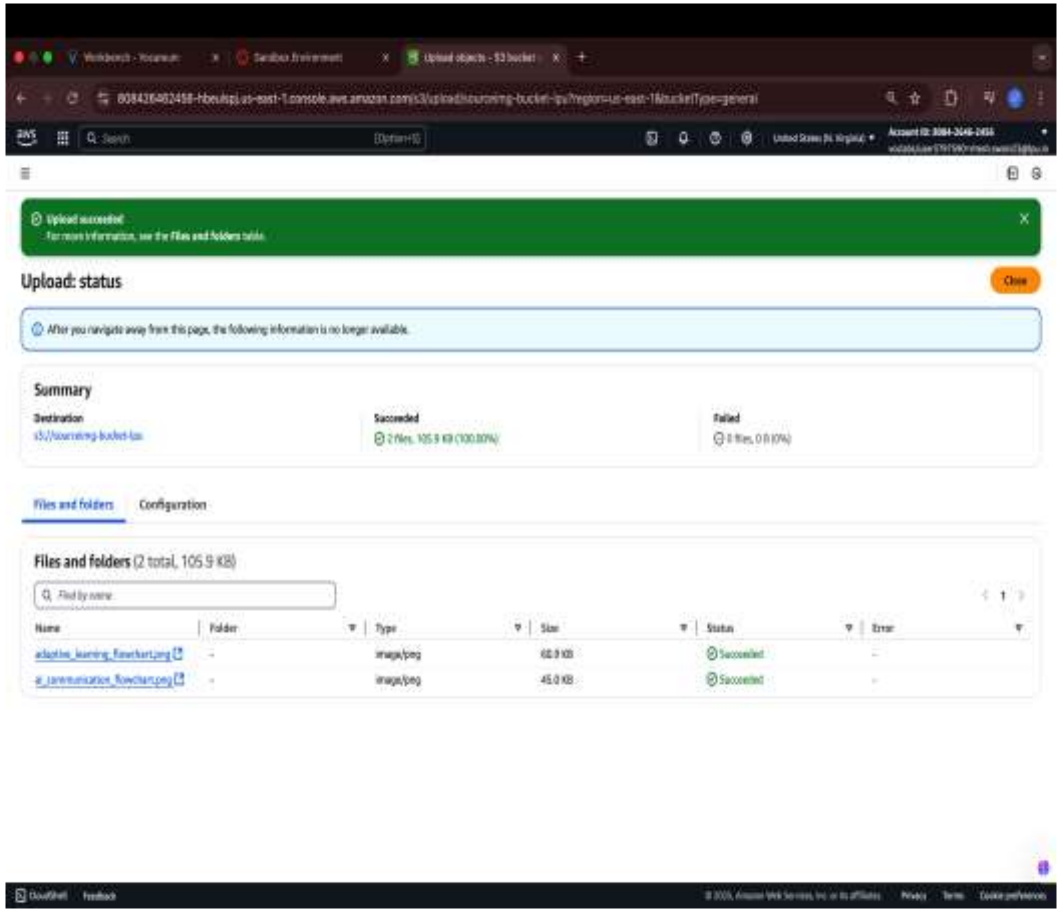
Package size
346 byte

SHA256 hash
t26hTxFgCeFKqAEdqFGlcs9c0RHQX5kygID+KN0h1Zl=

Last modified
2 minutes ago

8. Uploading Image to Source Bucket (sourceimg-bucket-lpu)

Manually uploading an image to the source bucket, which will trigger the Lambda function.



Upload succeeded
For more information, see the Files and folders table.

Upload: status

After you navigate away from this page, the following information is no longer available.

Summary

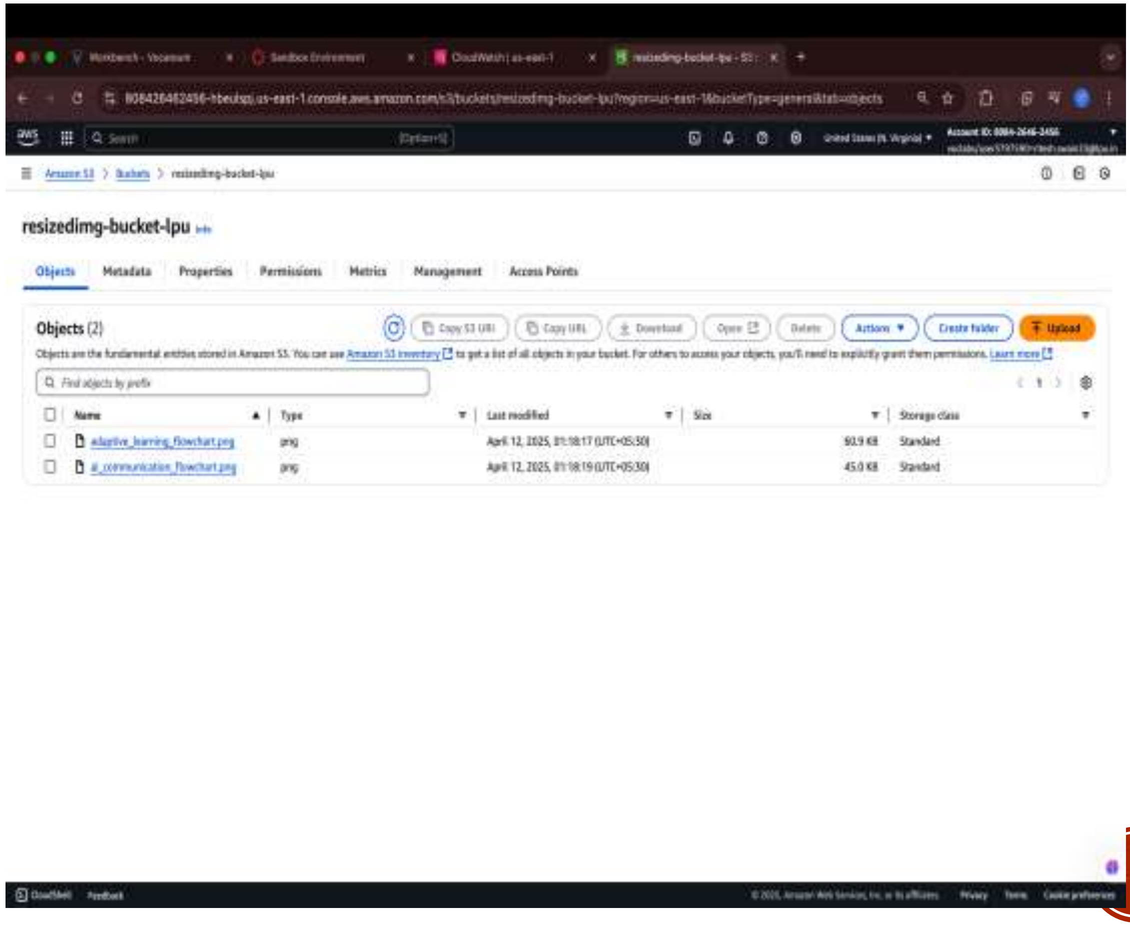
Destination	Succeeded	Failed
s3://sourceimg-bucket-lpu	2 files, 105.9 KB (100.00%)	0 files, 0 B (0%)

Files and folders (2 total, 105.9 KB)

Name	Folder	Type	Size	Status	Error
adaptive_learning_flowchart.png	-	image/png	60.9 KB	Succeeded	-
ai_communication_flowchart.png	-	image/png	45.0 KB	Succeeded	-

9. Image Automatically Appears in Resized Bucket () resizedimg-bucket-lpu

Lambda function successfully processed the uploaded image and saved a resized version to the destination bucket.



Amazon S3 > Buckets > resizedimg-bucket-lpu

resizedimg-bucket-lpu

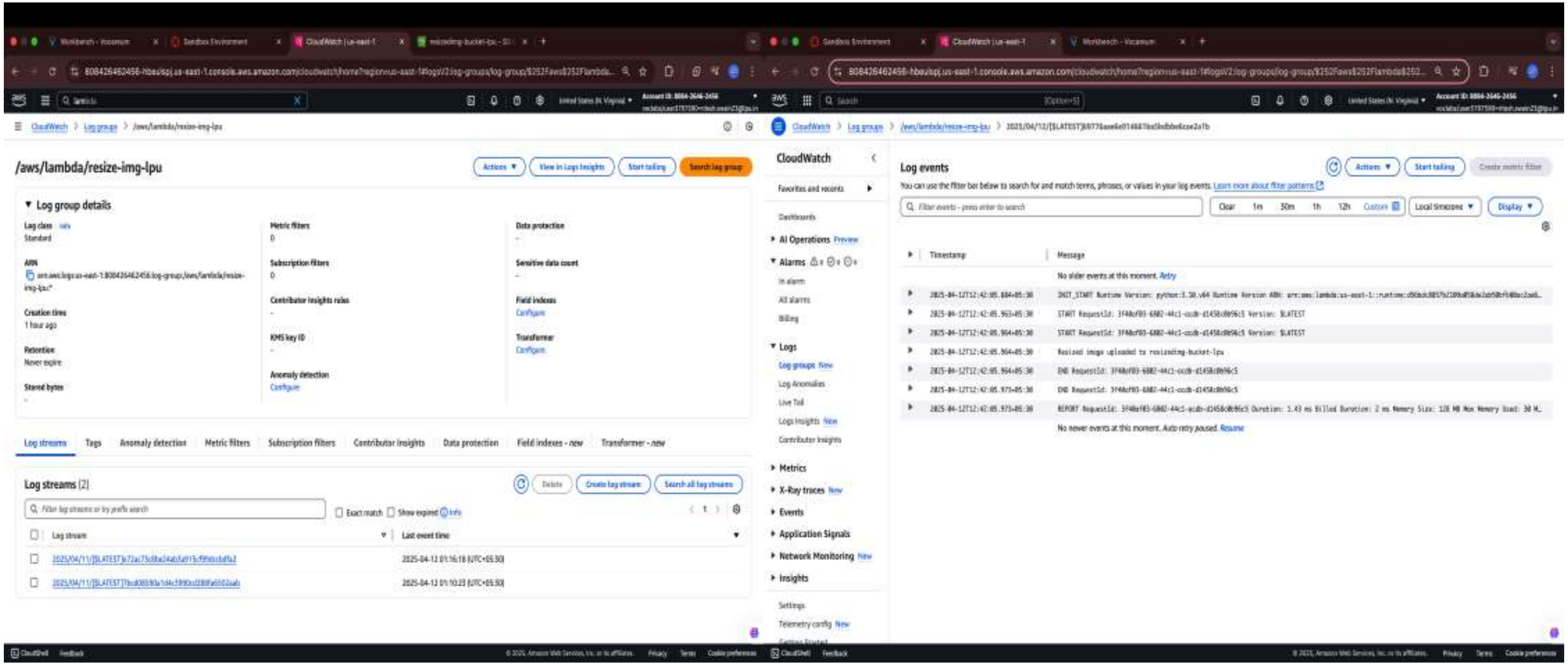
Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
adaptive_learning_flowchart.png	png	April 12, 2025, 01:18:17 (UTC+05:30)	60.9 KB	Standard
ai_communication_flowchart.png	png	April 12, 2025, 01:18:19 (UTC+05:30)	45.0 KB	Standard

10. CLOUDWATCH LOGS (SIMULATED SUCCESS LOG)

Displays logs in CloudWatch showing the Lambda function execution start, success message ("Resized image uploaded"), and end — confirming the workflow worked



/aws/lambda/resize-img-lpu

Log group details

Log class: Standard	Metric filters: 0	Data protection: -
ARN: arn:aws:logs:us-east-1:808426462456:log-group:/aws/lambda/resize-img-lpu:*	Subscription filters: 0	Sensitive data count: -
Creation time: 1 hour ago	Contributor Insights rules: -	Field indexes: Configure
Retention: Never expire	KMS key ID: -	Transformer: Configure
Stored bytes: -	Anomaly detection: Configure	

Log streams (2)

Log stream	Last event time
2025/04/11/[$LATEST]e72ac75c8ba24ab3a915cf9fdccbdfa2	2025-04-12 01:16:18 (UTC+05:30)
2025/04/11/[$LATEST]7bcd08590a1d4cf990cd280fa6502aab	2025-04-12 01:10:23 (UTC+05:30)

Log events

```
Timestamp                        Message
                                 No older events at this moment. Retry
2025-04-12T12:42:05.884+05:30    INIT_START Runtime Version: python:3.10.v64 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:d50bdc8857b2109a858da2ab58bfb48bc2ca6...
2025-04-12T12:42:05.963+05:30    START RequestId: 3f40af03-6802-44c1-acdb-d1458d0696c5 Version: $LATEST
2025-04-12T12:42:05.964+05:30    START RequestId: 3f40af03-6802-44c1-acdb-d1458d0696c5 Version: $LATEST
2025-04-12T12:42:05.964+05:30    Resized image uploaded to resizedimg-bucket-lpu
2025-04-12T12:42:05.964+05:30    END RequestId: 3f40af03-6802-44c1-acdb-d1458d0696c5
2025-04-12T12:42:05.973+05:30    END RequestId: 3f40af03-6802-44c1-acdb-d1458d0696c5
2025-04-12T12:42:05.973+05:30    REPORT RequestId: 3f40af03-6802-44c1-acdb-d1458d0696c5 Duration: 1.43 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 38 M...
                                 No newer events at this moment. Auto retry paused. Resume
```