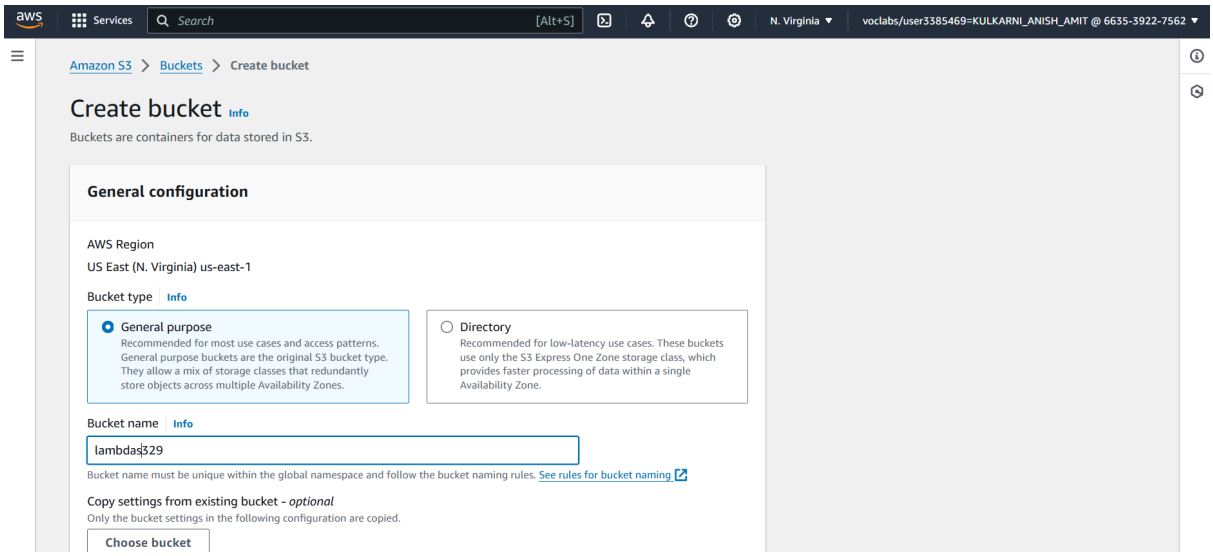
Name: Anish Kulkarni Roll No.: 29 Class: D15C AY: 2024-25

Experiment 12

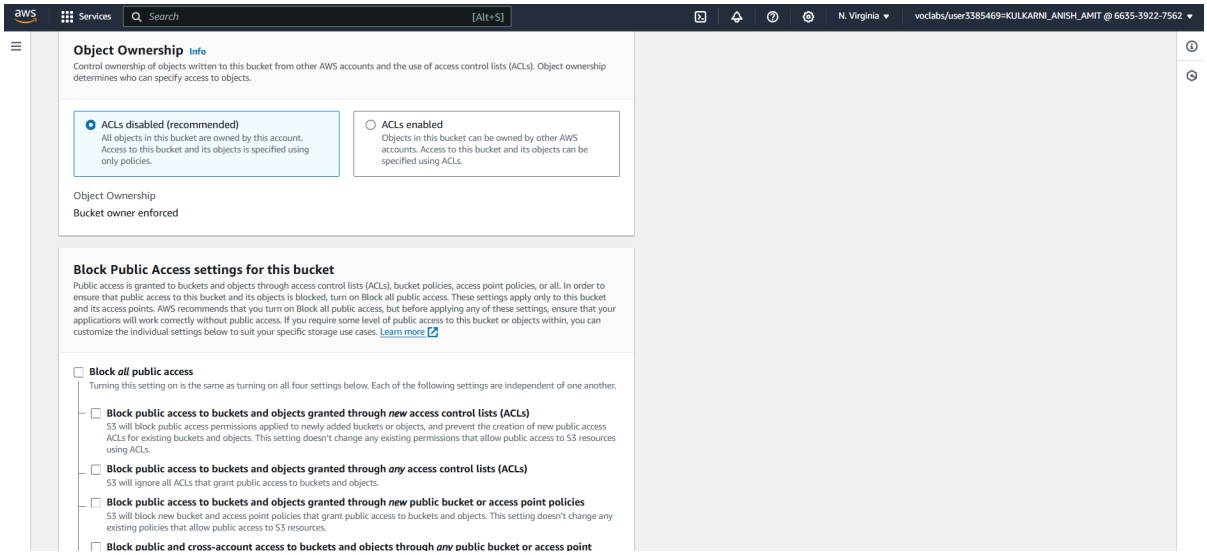
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.

Steps:

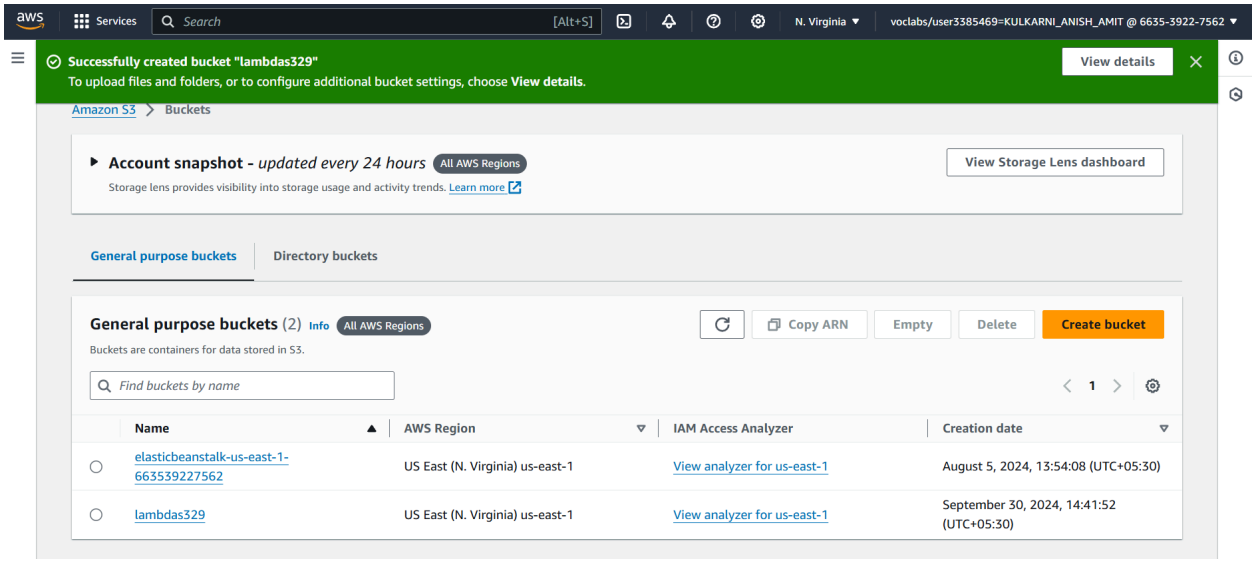
Step 1: On your AWS console, click on ‘S3’ in the services section and click on ‘Create bucket’. Give your bucket a name.



Uncheck the ‘Block all public access’ box.

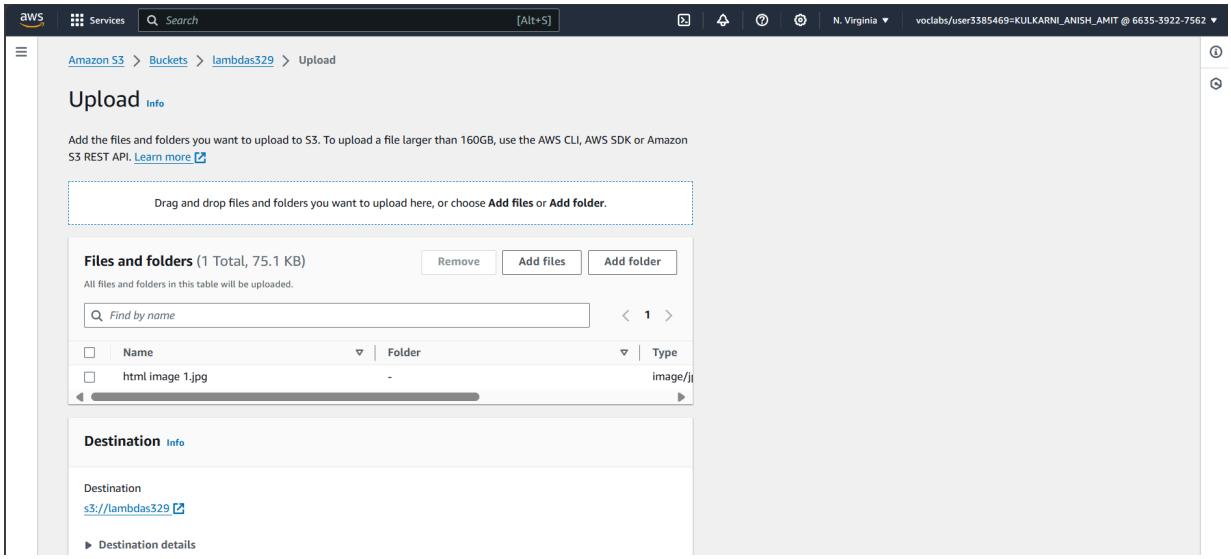


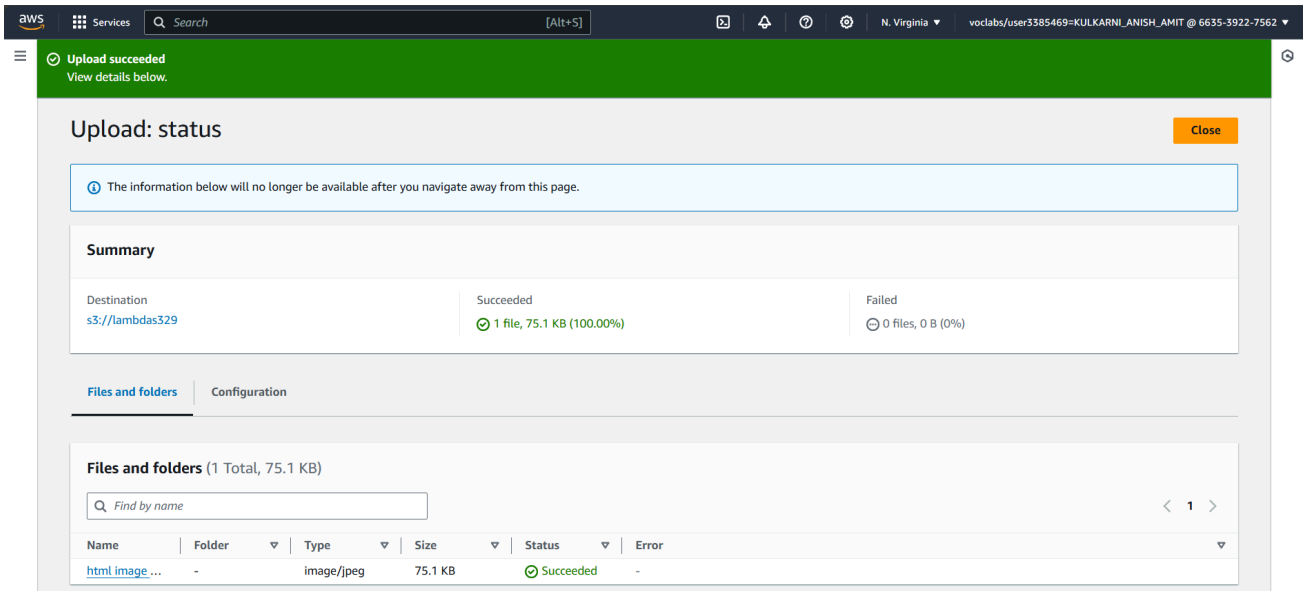
Keep all other options as default and click on ‘Create bucket’.



Your bucket is created.

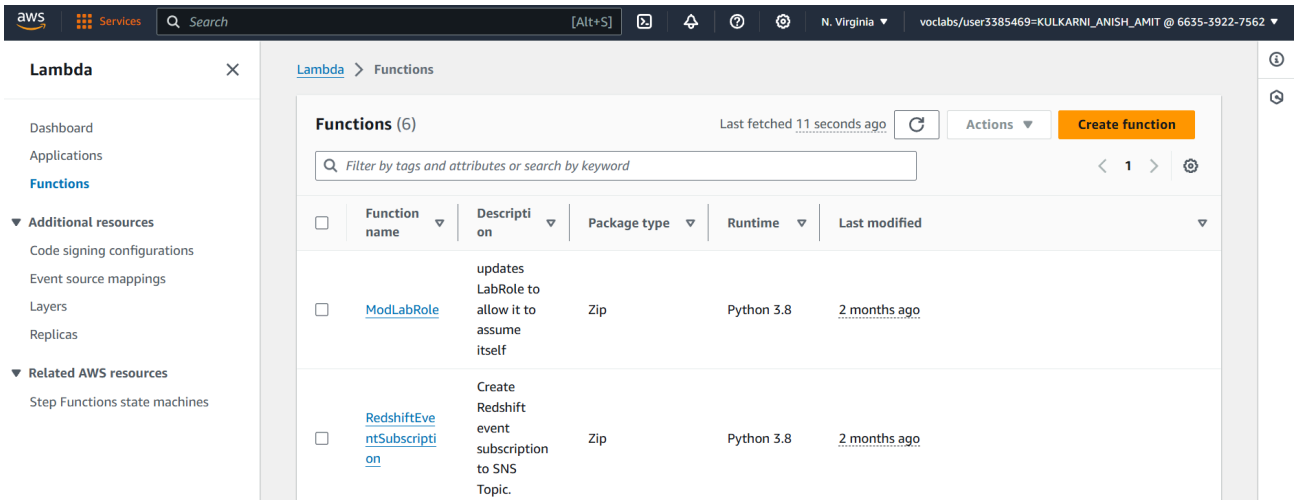
Step 2: Upload an image onto your S3 bucket by clicking on your S3 bucket, clicking on ‘Upload’, clicking on ‘Add files’, navigating to your image and selecting it.



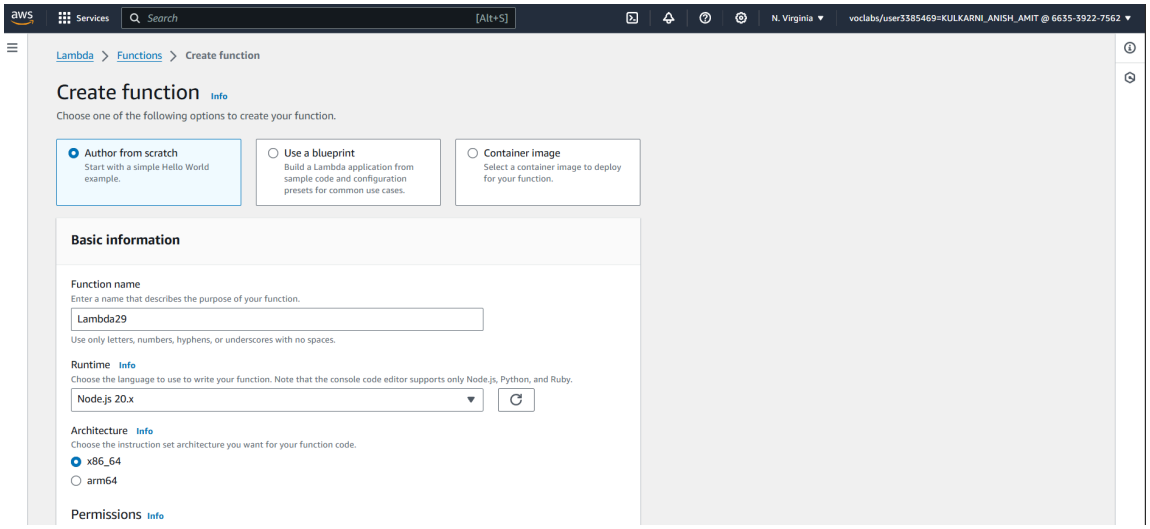


Your image gets uploaded onto the S3 bucket.

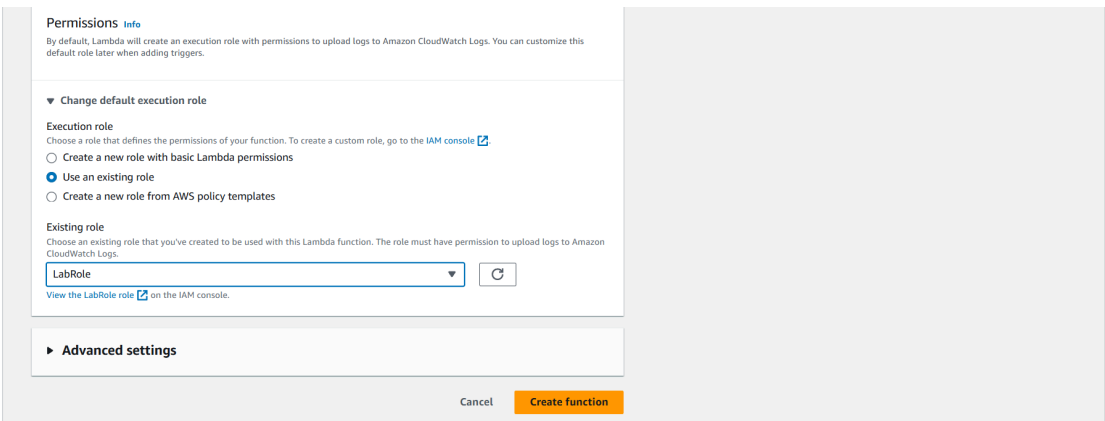
Step 3: Navigate to the AWS Lambda console using the ‘Services’ section. Click on ‘Create function’.



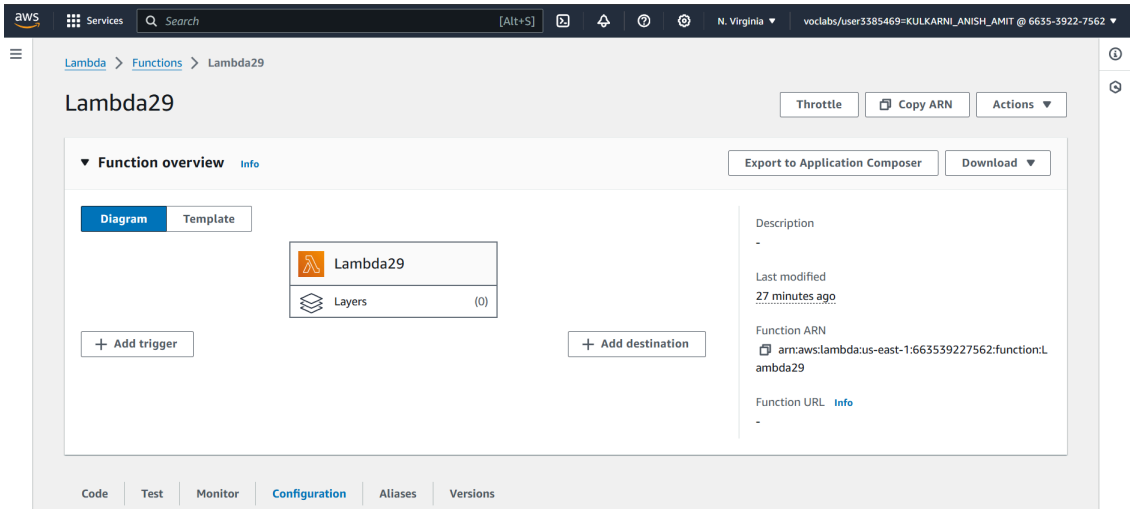
Step 4: Give your function a name and keep other settings as default.



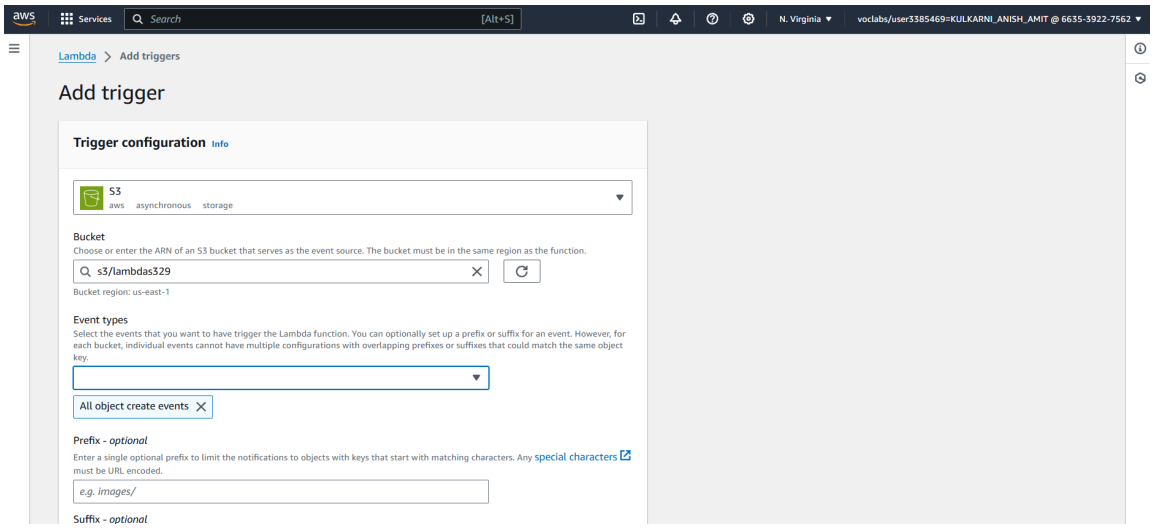
Under ‘Execution role’, choose ‘Use an existing role’ and in the dropdown box below, choose ‘LabRole’. Then, click on ‘Create function’. Your function gets created.

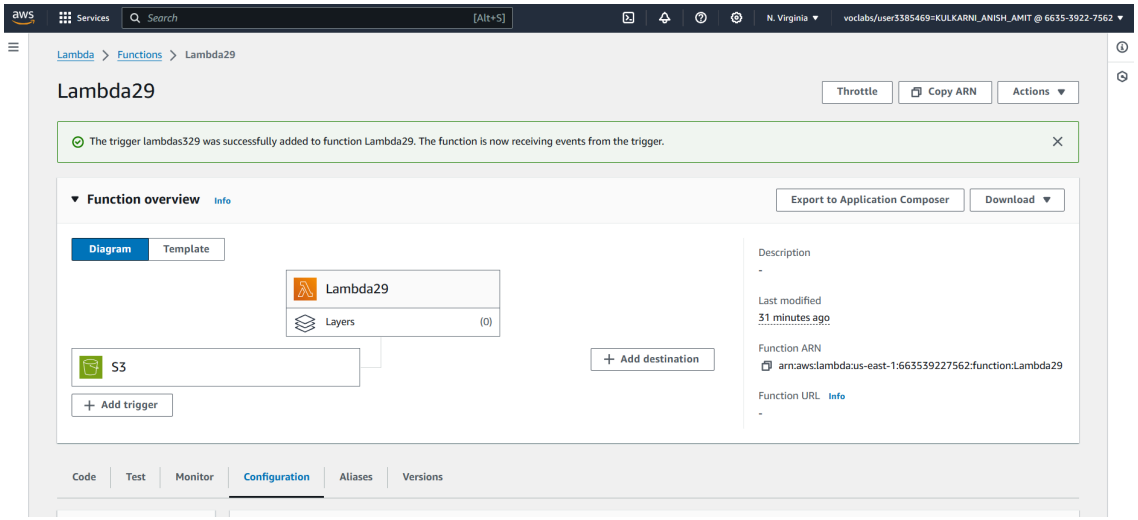


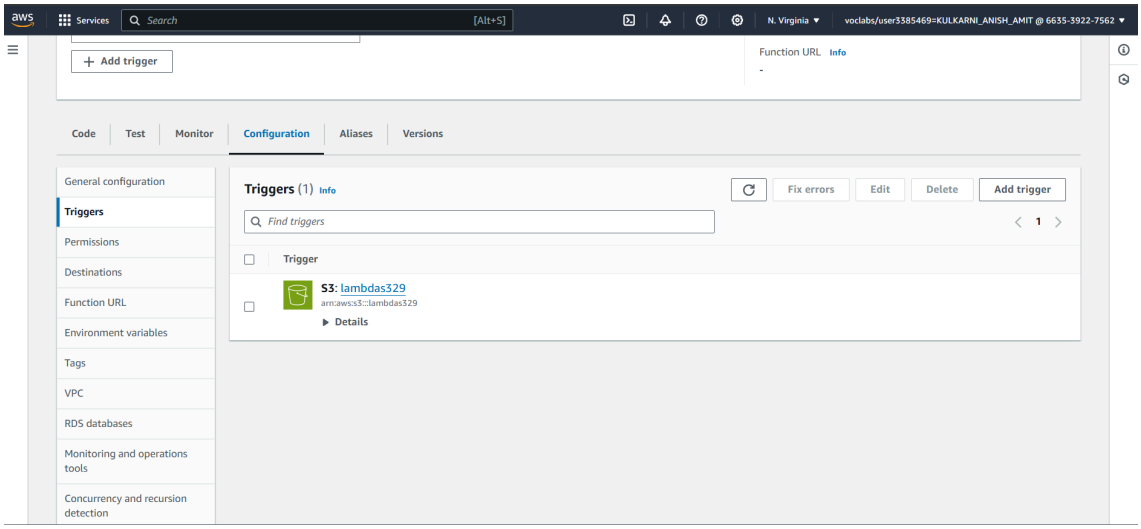
Step 5: On the page of the function you created, click on ‘Add trigger’.



Step 6: Choose ‘Trigger configuration’ as S3 and select the name of your bucket in the dropdown box below it. Keep other options as default and click on ‘Add’.



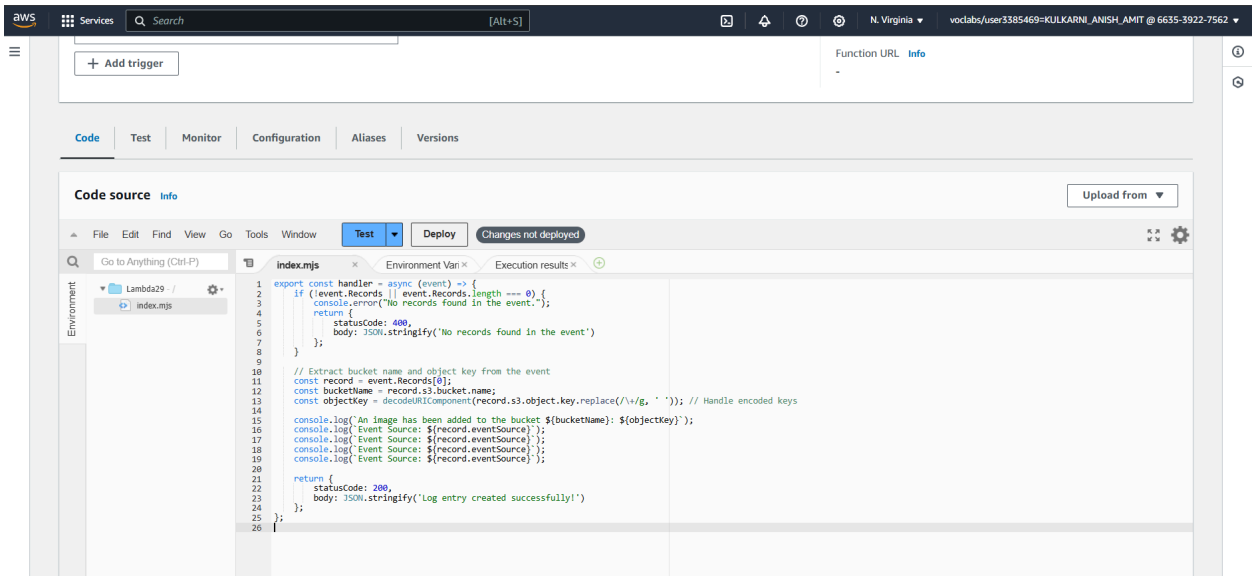




The trigger gets successfully added to your function.

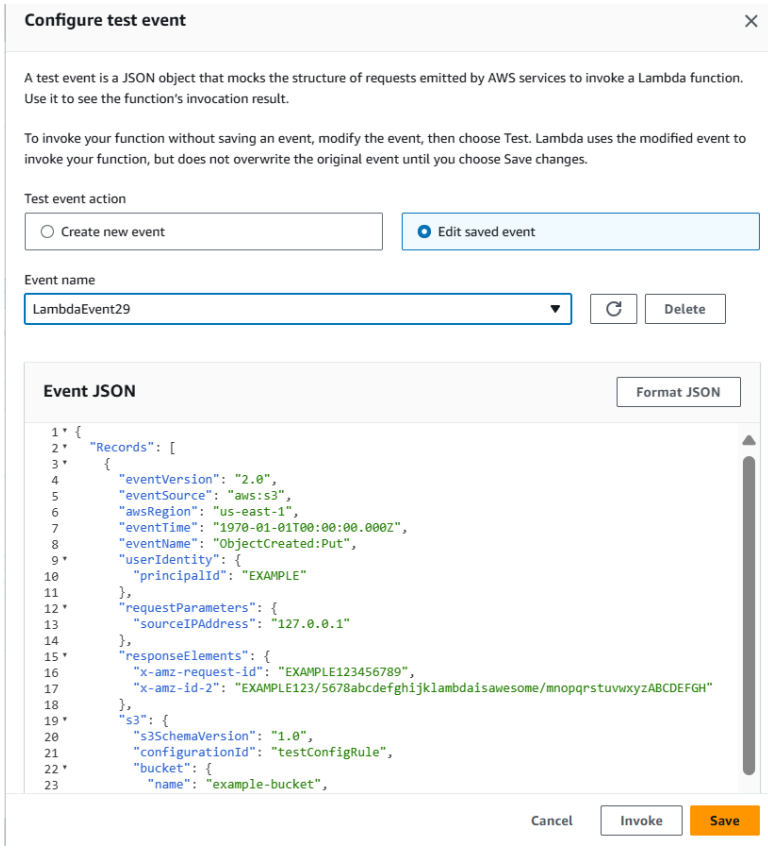
Step 7: In the ‘Code source’ section of your function, paste the following javascript code instead of the existing code:-

```
export const handler = async (event) => {
    if (!event.Records || event.Records.length === 0) {
        console.error("No records found in the event.");
        return {
            statusCode: 400,
            body: JSON.stringify('No records found in the event')
        };
    }

    // Extract bucket name and object key from the event
    const record = event.Records[0];
    const bucketName = record.s3.bucket.name;
    const objectKey = decodeURIComponent(record.s3.object.key.replace(/\+/g, ' ')); // Handle encoded keys

    console.log(`An image has been added to the bucket ${bucketName}: ${objectKey}`);
    console.log(`Event Source: ${record.eventSource}`);
    console.log(`Event Source: ${record.eventSource}`);
    console.log(`Event Source: ${record.eventSource}`);
    console.log(`Event Source: ${record.eventSource}`);

    return {
        statusCode: 200,
        body: JSON.stringify('Log entry created successfully!')
    };
};
```



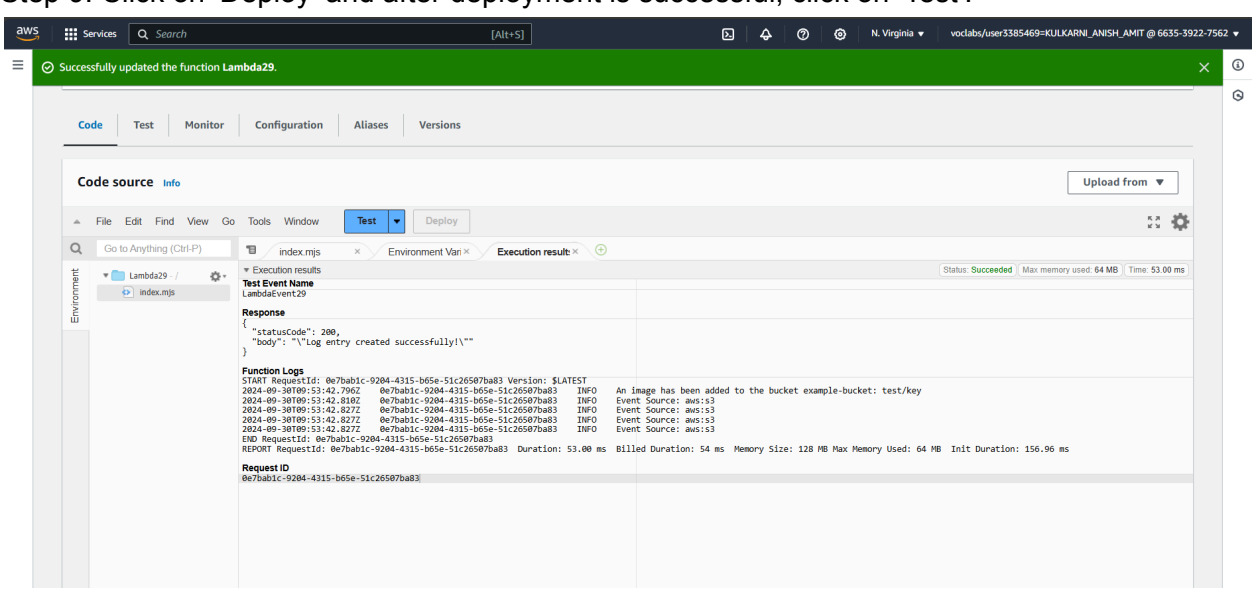
Step 8: Click on the arrow next to the ‘Test’ button and click on ‘Configure test event’. In the popup box that appears, if you have an existing event, enter the name of your event or create a new event and in the ‘Event JSON’ section, paste the following code:-


```
{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789",
        "x-amz-id-2":
"EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "example-bucket",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::example-bucket"
        },
        "object": {
          "key": "test%2Fkey",
          "size": 1024,
          "eTag": "0123456789abcdef0123456789abcdef",
          "sequencer": "0A1B2C3D4E5F678901"
        }
      }
    }
  ]
}
```

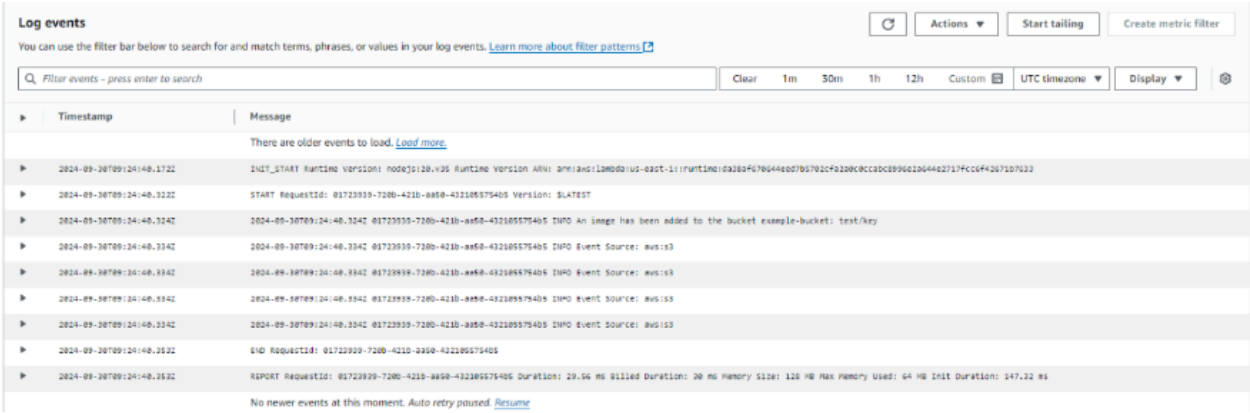


Then, click on 'Save'. Your function gets successfully updated.

Step 9: Click on 'Deploy' and after deployment is successful, click on 'Test'.



Running the test gives the above output which displays that 'An Image has been added to the bucket' and that the log entry was successfully created.



Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

Actions | Start tailing | Create metric filter

Filter events - press enter to search | Clear | 1m | 30m | 1h | 12h | Custom | UTC timezone | Display

Timestamp	Message
	There are older events to load. Load more.
2024-09-30T09:24:40.172Z	INIT_START Runtime Version: nodejs:20.v35 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:da38bf6706444ed7b6702cf0200c0ccab0c8596210644e2717fcc6f4267107633
2024-09-30T09:24:40.322Z	START RequestId: 01723939-720b-421b-aa60-432105575405 Version: $LATEST
2024-09-30T09:24:40.324Z	2024-09-30T09:24:40.324Z 01723939-720b-421b-aa60-432105575405 INFO An image has been added to the bucket example-bucket: test/key
2024-09-30T09:24:40.334Z	2024-09-30T09:24:40.334Z 01723939-720b-421b-aa60-432105575405 INFO Event Source: aws:s3
2024-09-30T09:24:40.334Z	2024-09-30T09:24:40.334Z 01723939-720b-421b-aa60-432105575405 INFO Event Source: aws:s3
2024-09-30T09:24:40.334Z	2024-09-30T09:24:40.334Z 01723939-720b-421b-aa60-432105575405 INFO Event Source: aws:s3
2024-09-30T09:24:40.334Z	2024-09-30T09:24:40.334Z 01723939-720b-421b-aa60-432105575405 INFO Event Source: aws:s3
2024-09-30T09:24:40.353Z	END RequestId: 01723939-720b-421b-aa60-432105575405
2024-09-30T09:24:40.353Z	REPORT RequestId: 01723939-720b-421b-aa60-432105575405 Duration: 29.56 ms Billed Duration: 30 ms Memory Size: 128 MB Max Memory Used: 64 MB Init Duration: 147.32 ms
	No newer events at this moment. Auto retry paused. Resume

Conclusion: In this experiment, we learned how to create a Lambda function which logs “An image has been added” once we add an image to our specific S3 bucket. We first created an S3 bucket and uploaded an image to it. We then created a Lambda function and added an S3 trigger to it and selected the S3 bucket we created. Then, we configured the ‘Code section’ of our Lambda function and a test event for our Lambda function. On running the test event, we observed that it logged important information about the event such as the bucket name and object key and also verified that an image had been added to the S3 bucket. Also, a log of our Lambda function was also created which confirmed that the image had been successfully added to the bucket.