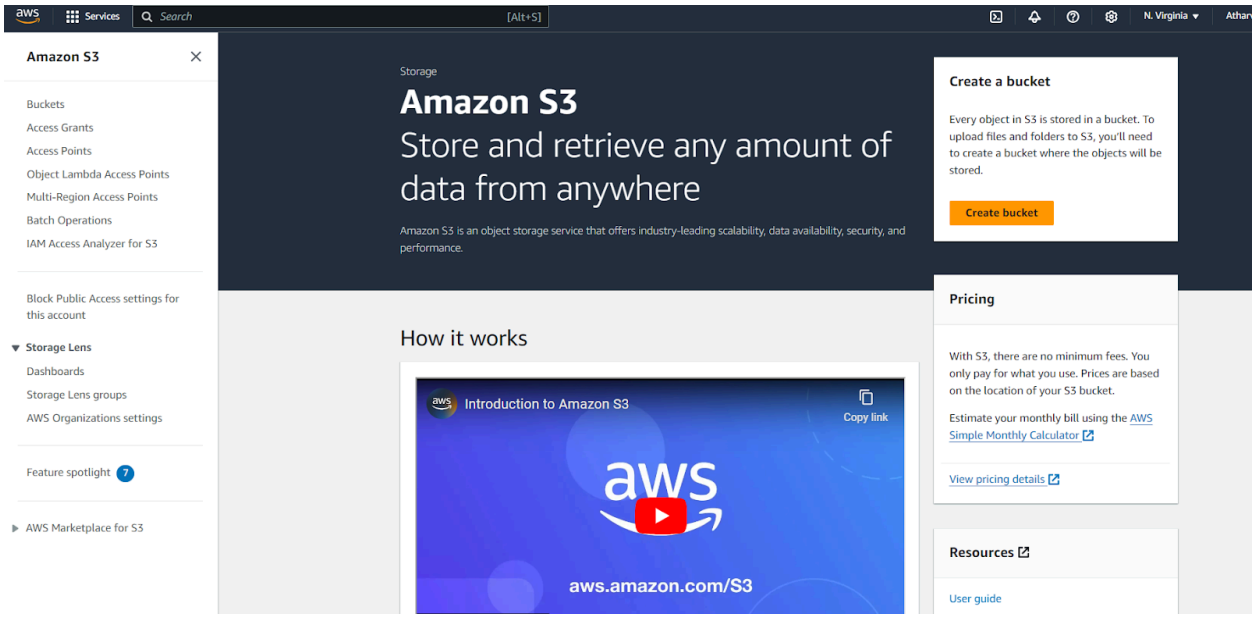
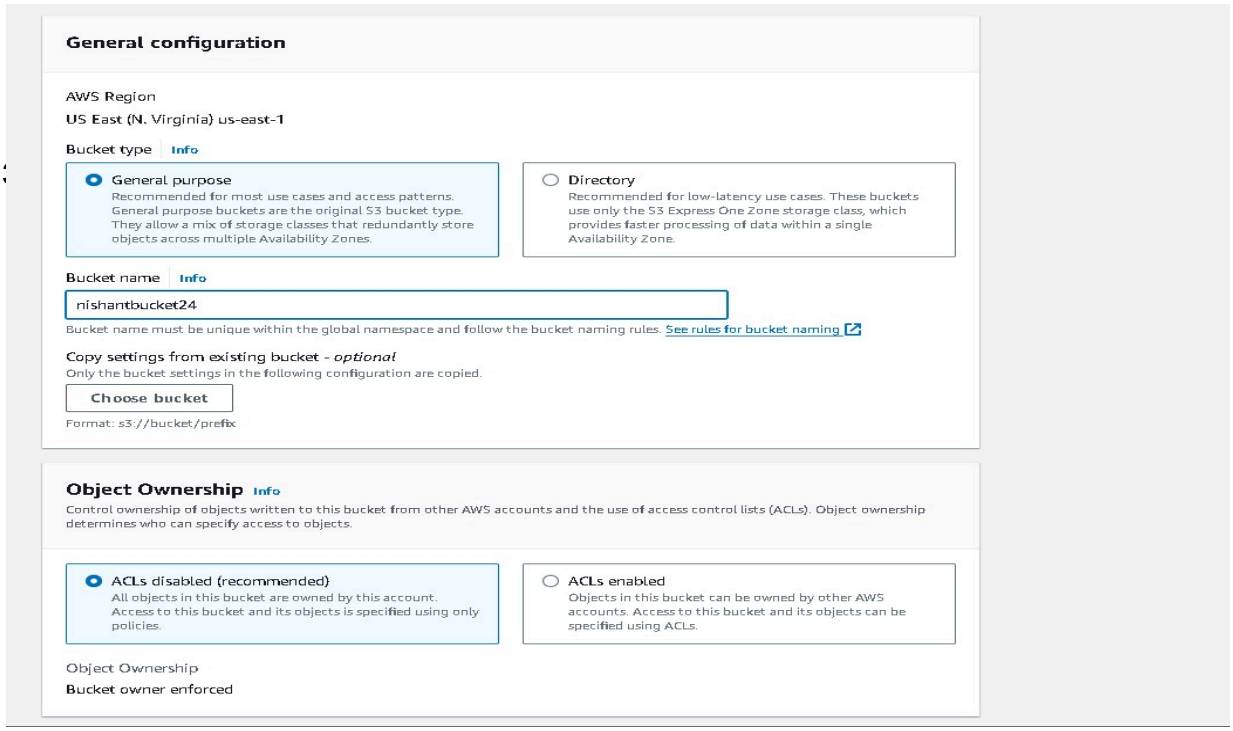
**Name : Atharva Patil** **Class : D15C** **Roll No : 39**

**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.

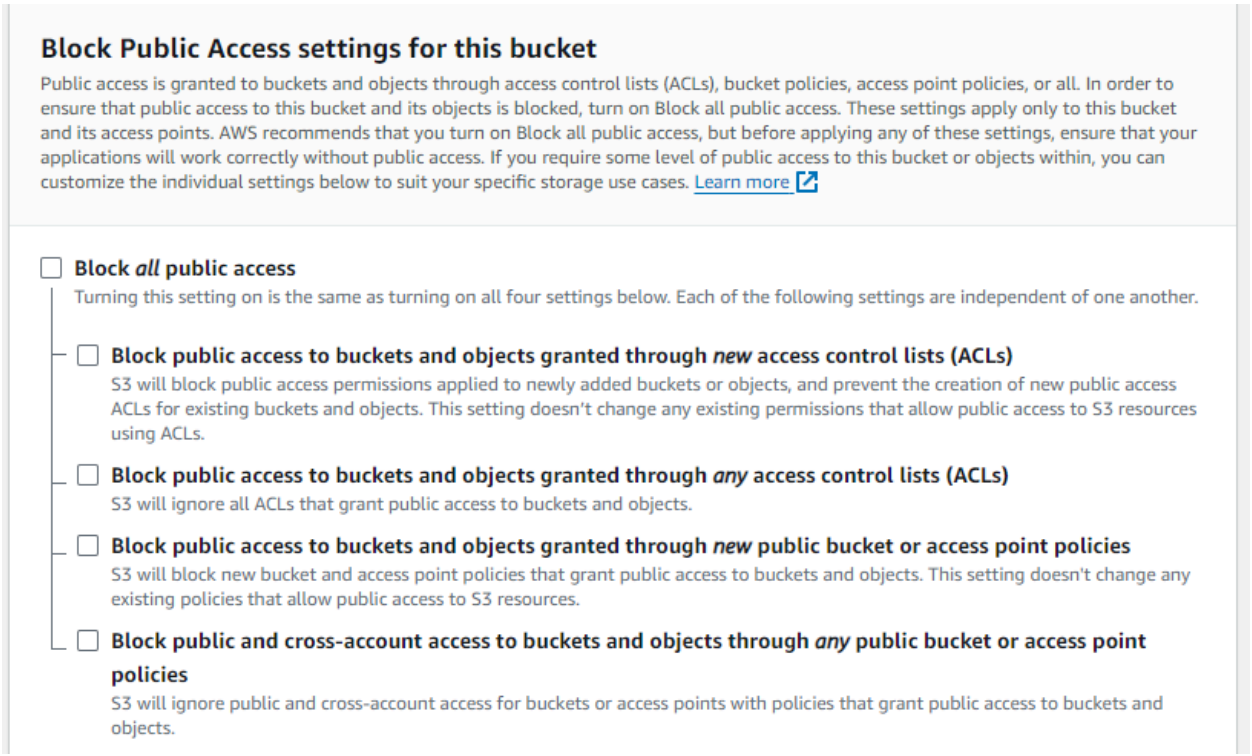
**Step 1: Create a S3 bucket. 1) Search for S3 bucket in the services search. Then click on create bucket.**



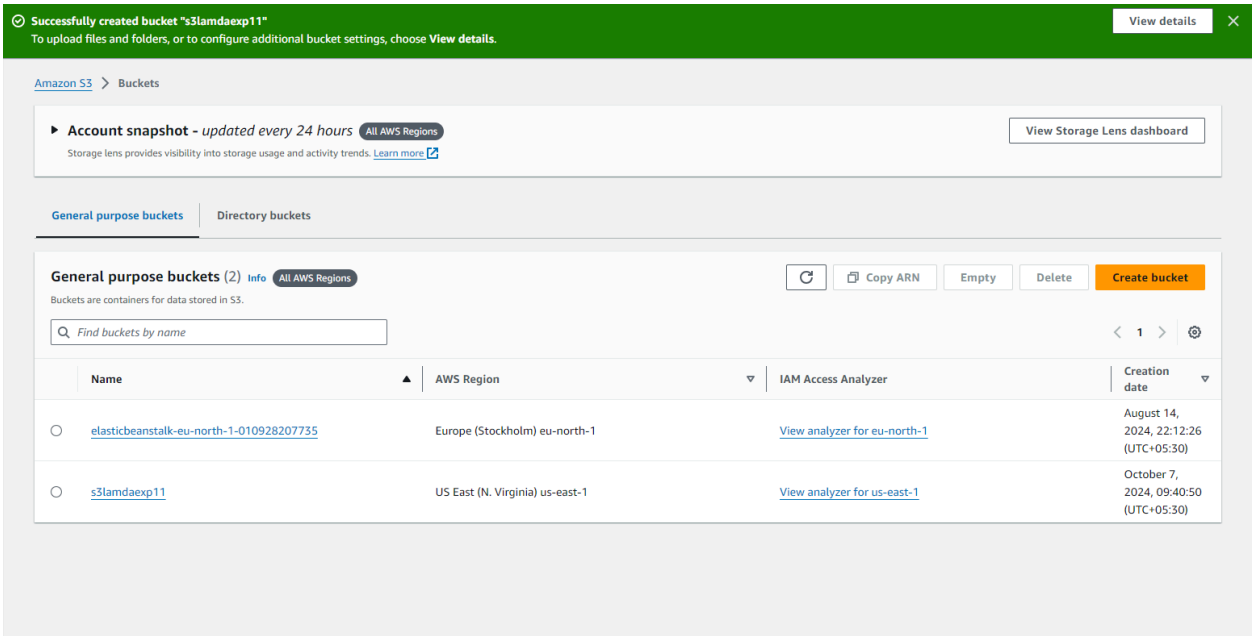

**2) Keep the bucket as a general purpose bucket. Give a name to your bucket.**



**Name : Atharva Patil** **Class : D15C** **Roll No : 39**

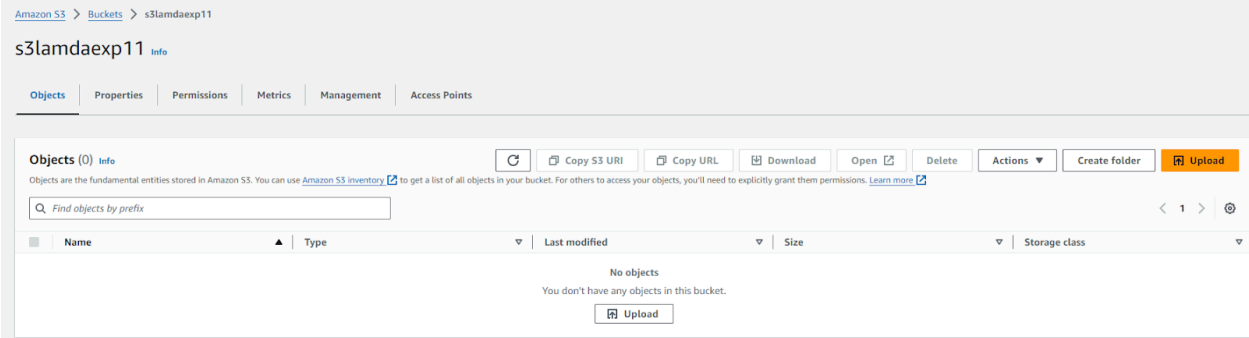


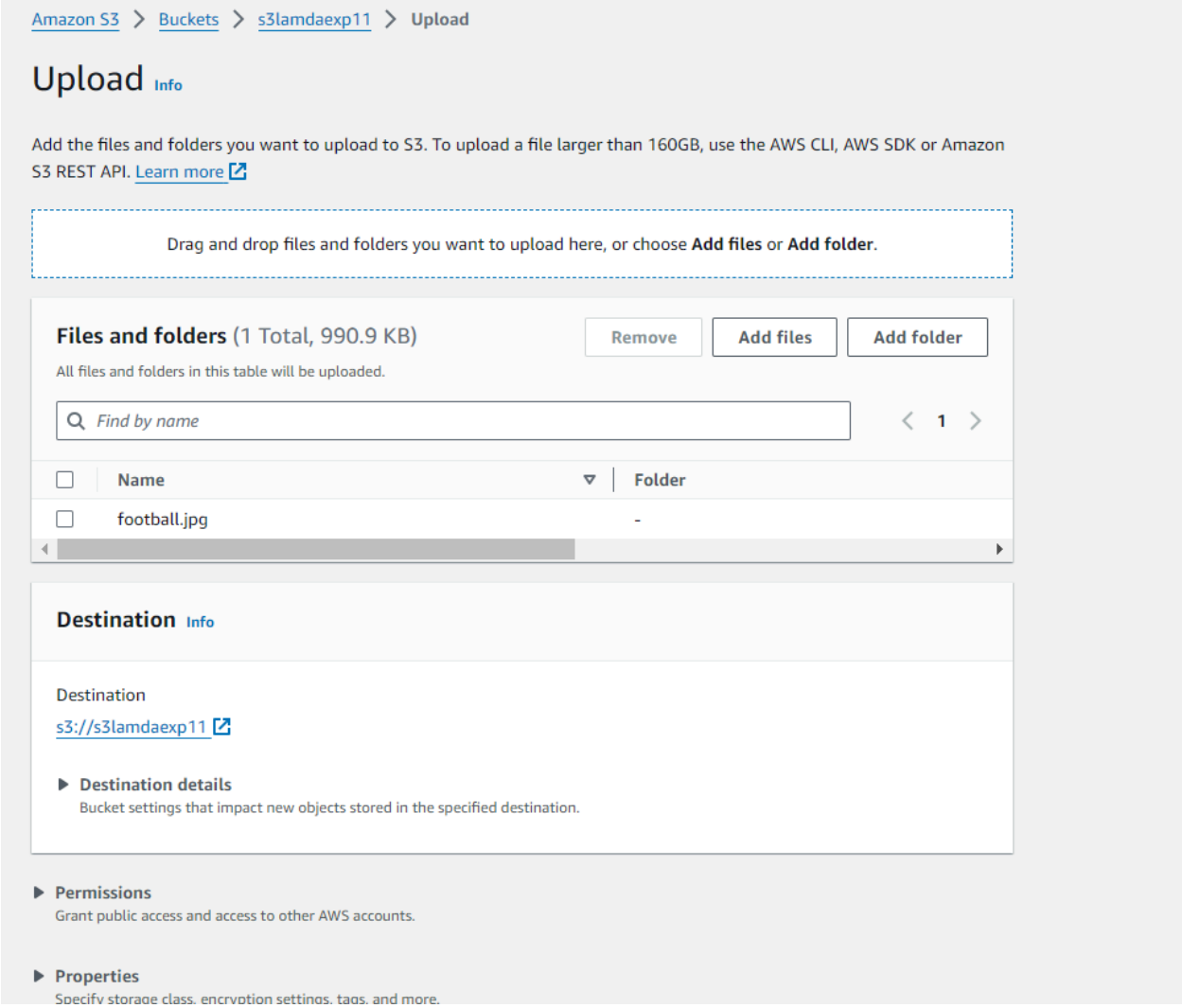

**4) Keeping all other options the same, click on create. This would create your bucket. Now click on the name of the bucket.**




**5) Here, click on upload, then add files. Select any image that you want to upload in the bucket and click on upload.**

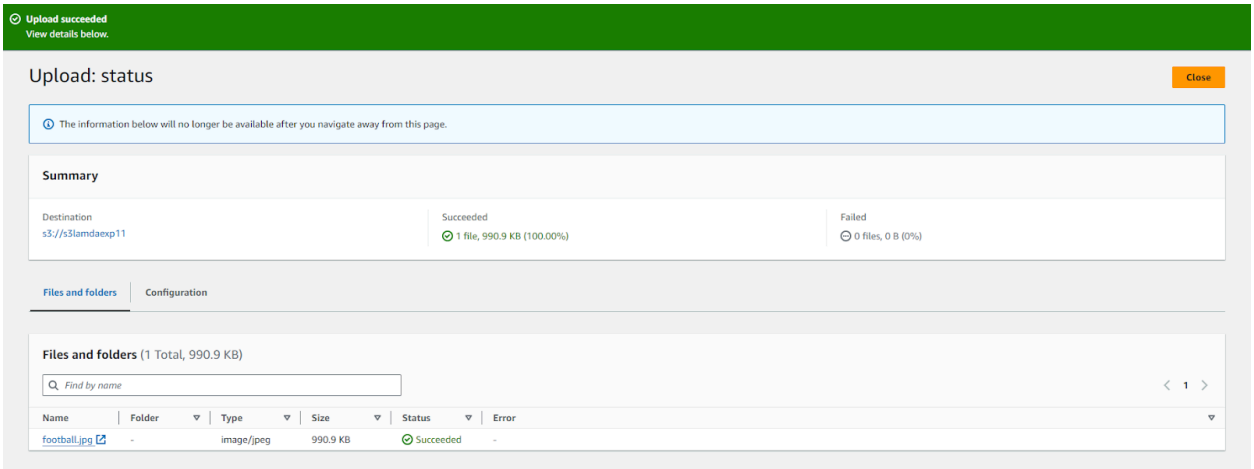
**Name : Atharva Patil** **Class : D15C** **Roll No : 39**



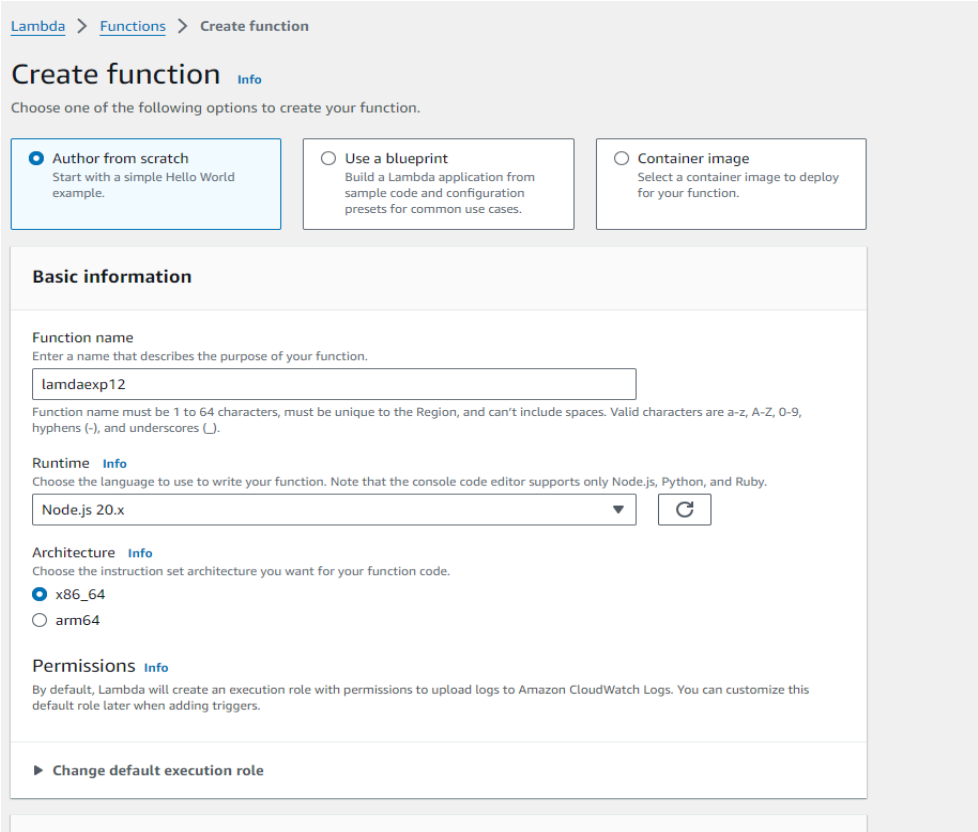

6) The image has been uploaded to the bucket.

**Name : Atharva Patil** **Class : D15C** **Roll No : 39**



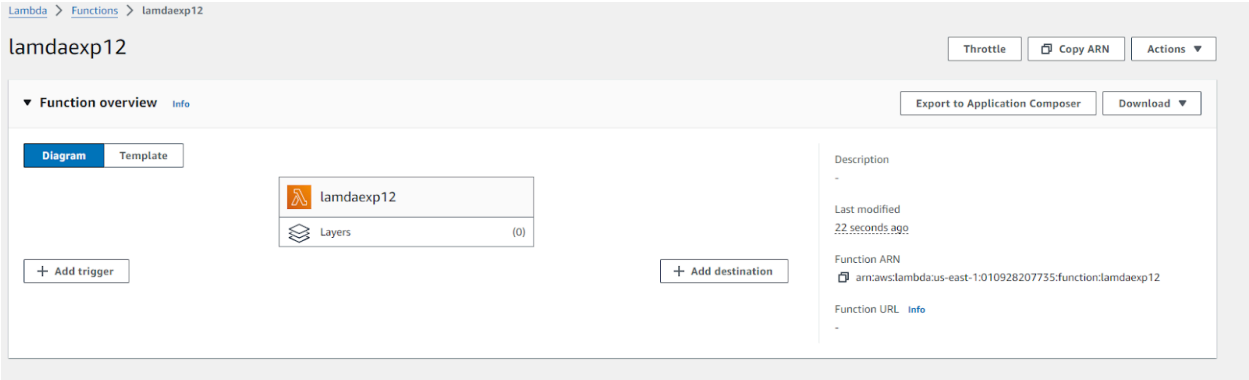

**Step 2: Configure Lambda function**

**1) Go to the lambda function you had created berfor. (Services → Lambda → Click on name of function). Here, click on add trigger**

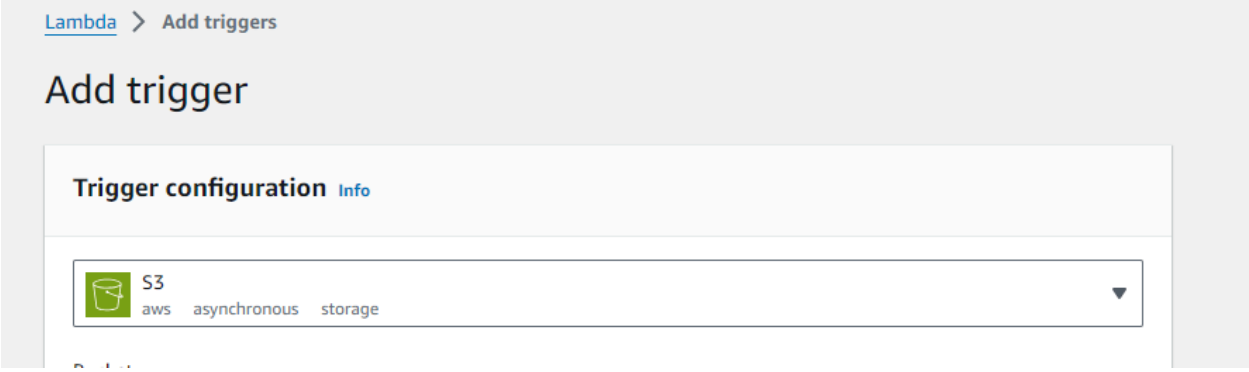


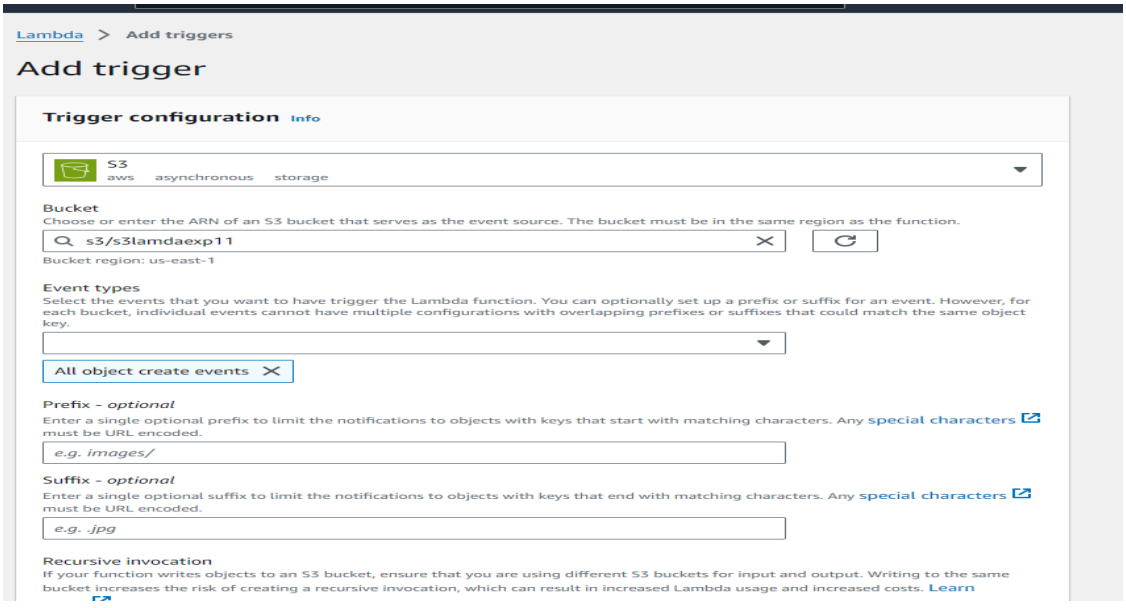

**2) Under trigger configuration, search for S3 and select it.**

**Name : Atharva Patil** **Class : D15C** **Roll No : 39**

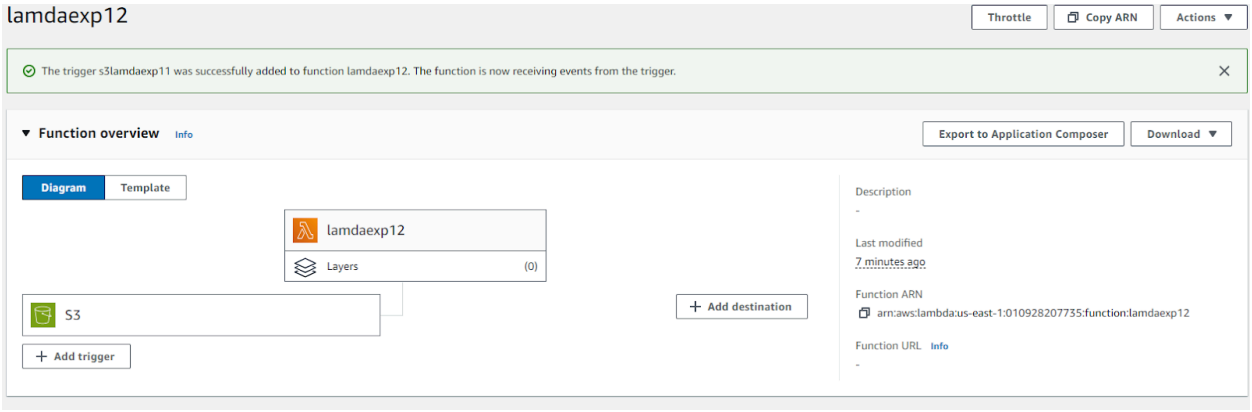



**3) Here, select the S3 bucket you created for this experiment. Acknowledge the condition given by AWS. then click on Add. This will add the S3 bucket trigger to your function**



**Name : Atharva Patil** **Class : D15C** **Roll No : 39**

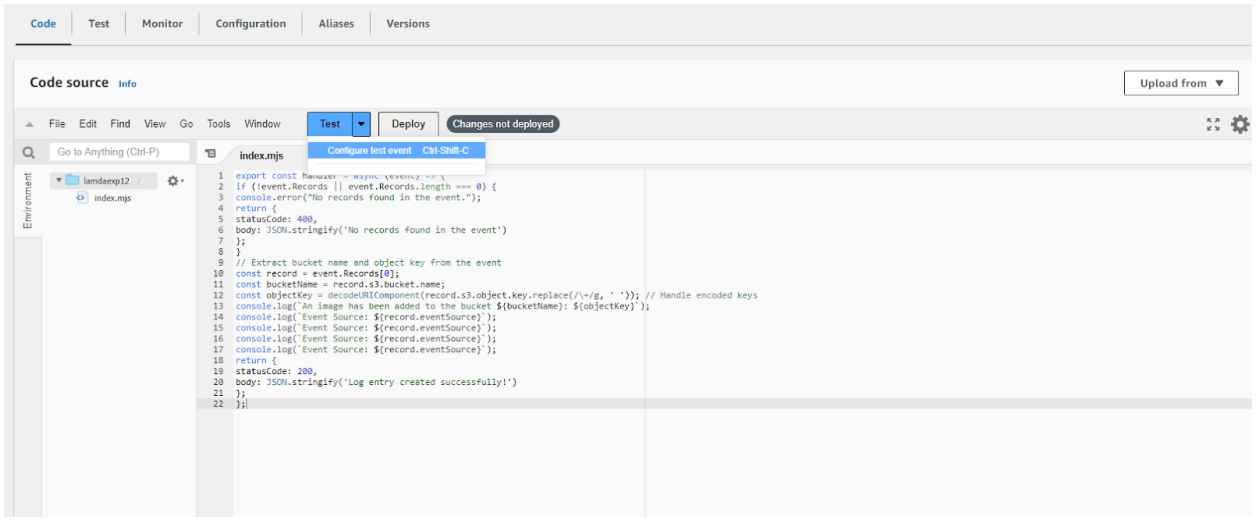



4) Scroll down to the code section of the function. Add the following javascript code to the code area by replacing the existing code

```
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!')
};
};
```

**Name : Atharva Patil** **Class : D15C** **Roll No : 39**

This JSON structure represents an S3 event notification triggered when an object is uploaded to an S3 bucket. It contains details about the event, including the bucket name (example-bucket), the object key (test/key), and metadata like the object's size, the event source (aws:s3), and the event time.



**Name : Atharva Patil** **Class : D15C** **Roll No : 39**

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - *optional*

hello-world

**Event JSON** Format JSON

```
{
  "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",
```

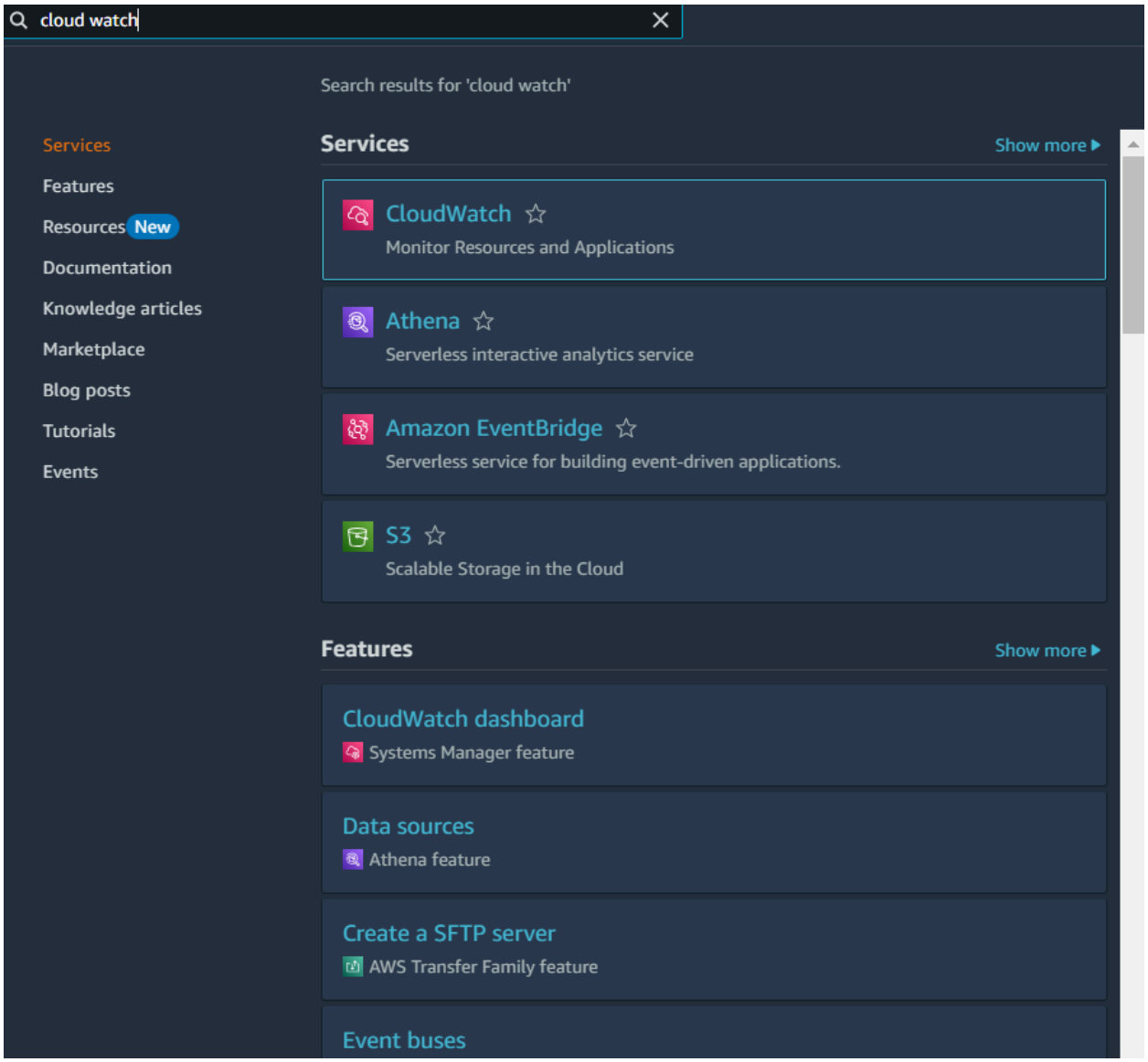
1:1 JSON Spaces: 2

Cancel Invoke Save

**Name : Atharva Patil** **Class : D15C** **Roll No : 39**

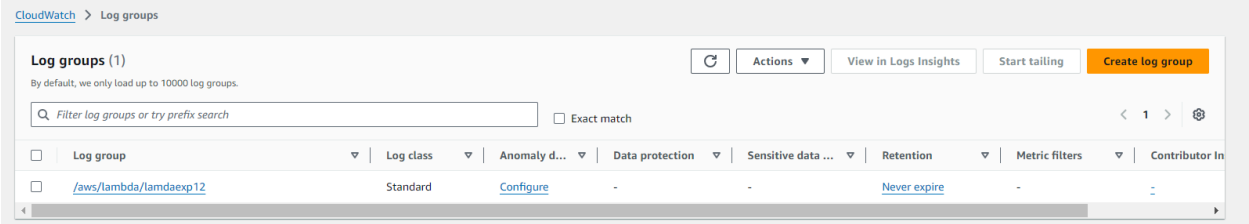
**Step 3: Check the logs**

**1) To check the logs explicitly, search foe CloudWatch on services and open it in a new tab**

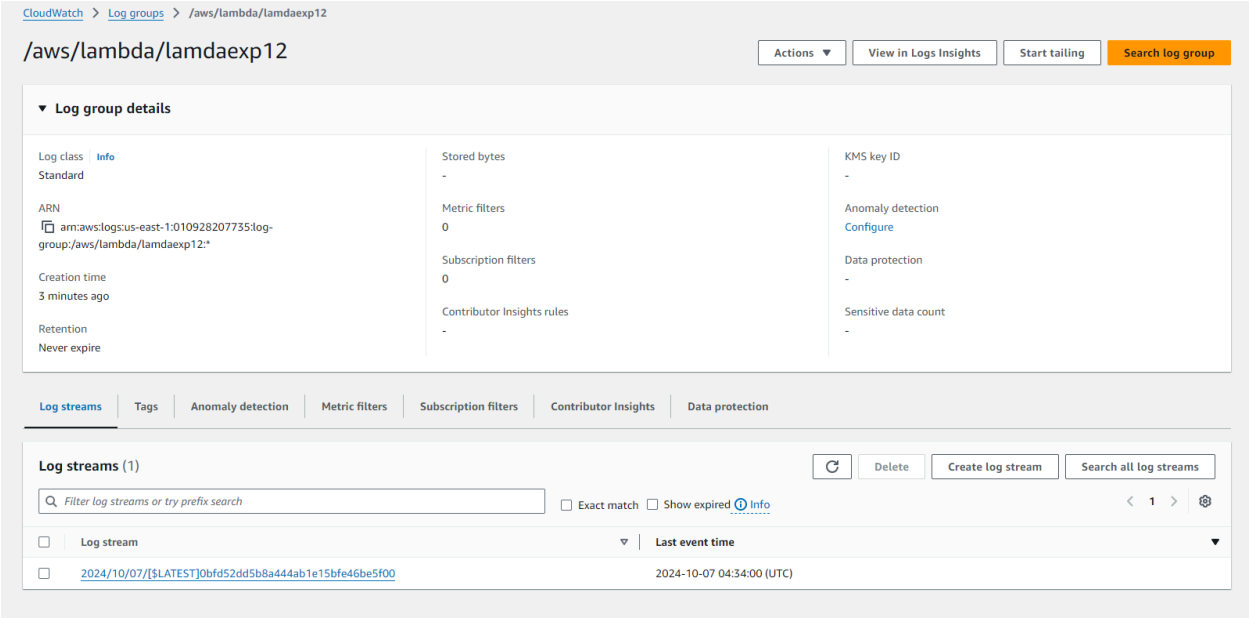



**2) Here, Click on Logs → Log Groups. Select the log that has the lambda function name you just ran.**

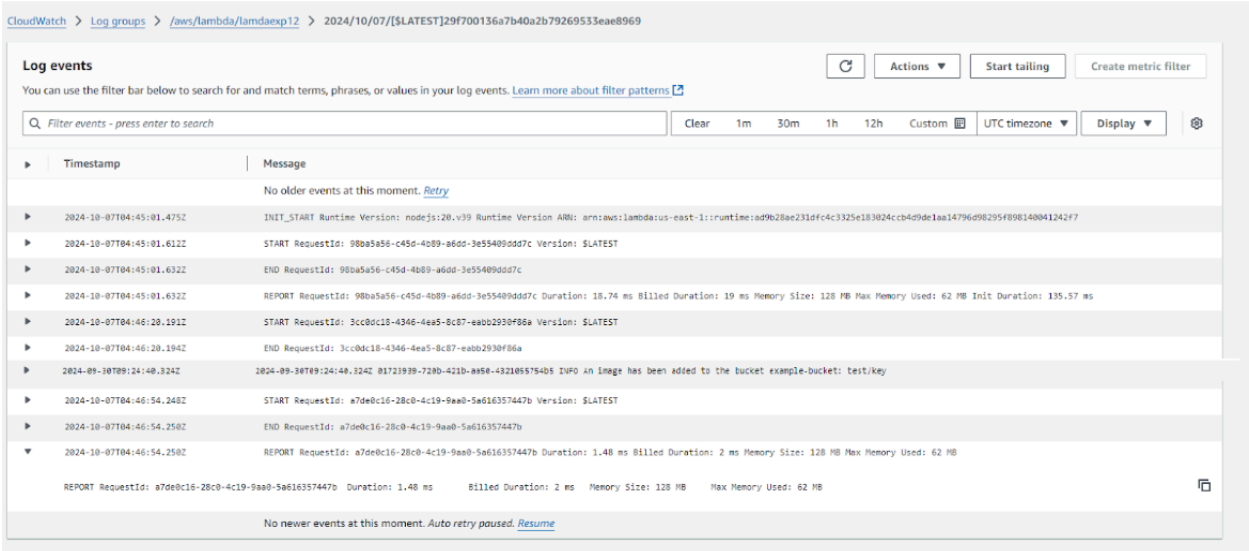
**Name : Atharva Patil** **Class : D15C** **Roll No : 39**




**3) Here, under Log streams, select the log stream you want to check.**




**4) Here again, we can see that 'An image has been added to the bucket'.**



**Name : Atharva Patil** **Class : D15C** **Roll No : 39**

**Conclusion:**

In this experiment, In addition to demonstrating the integration of AWS Lambda with S3, this experiment showcases the scalability and flexibility of serverless architectures. By leveraging these services, we can build applications that respond in real-time to changes in data, such as the addition of files to S3 buckets, without the need for managing underlying server infrastructure. This not only enhances efficiency but also reduces operational costs, allowing developers to focus on building features rather than maintaining systems. Furthermore, the ability to log and monitor events through CloudWatch opens opportunities for further automation and analytics, paving the way for more complex workflows and data processing solutions.