Sairam Konar D15C 27 2024-25

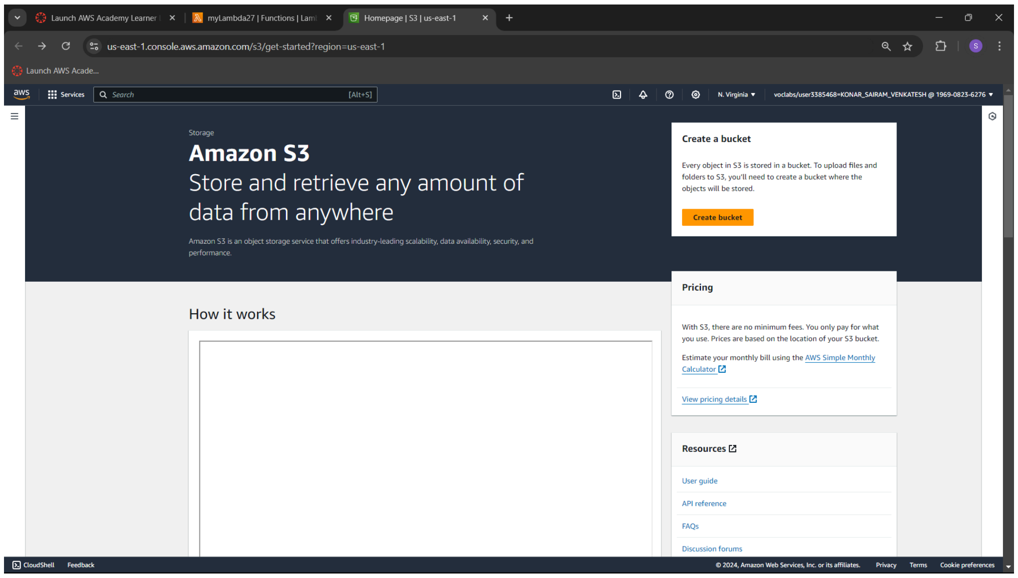
Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3.

Prerequisites:

1) AWS account (acacdemy preferable)
2) Lambda function (created in the previous experiment).

Step 1: Create a s3 bucket.

1) Search for S3 bucket in the services search. Then click on create bucket.



Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

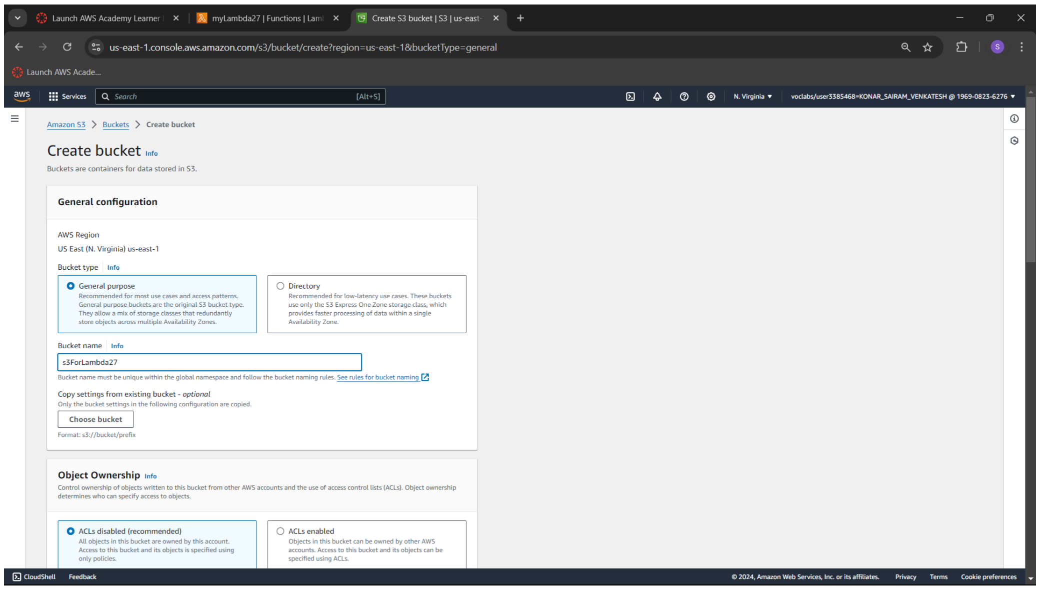
Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the AWS Simple Monthly Calculator

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



Create bucket

Buckets are containers for data stored in S3.

General configuration

AWS Region: US East (N. Virginia) us-east-1

Bucket type: General purpose

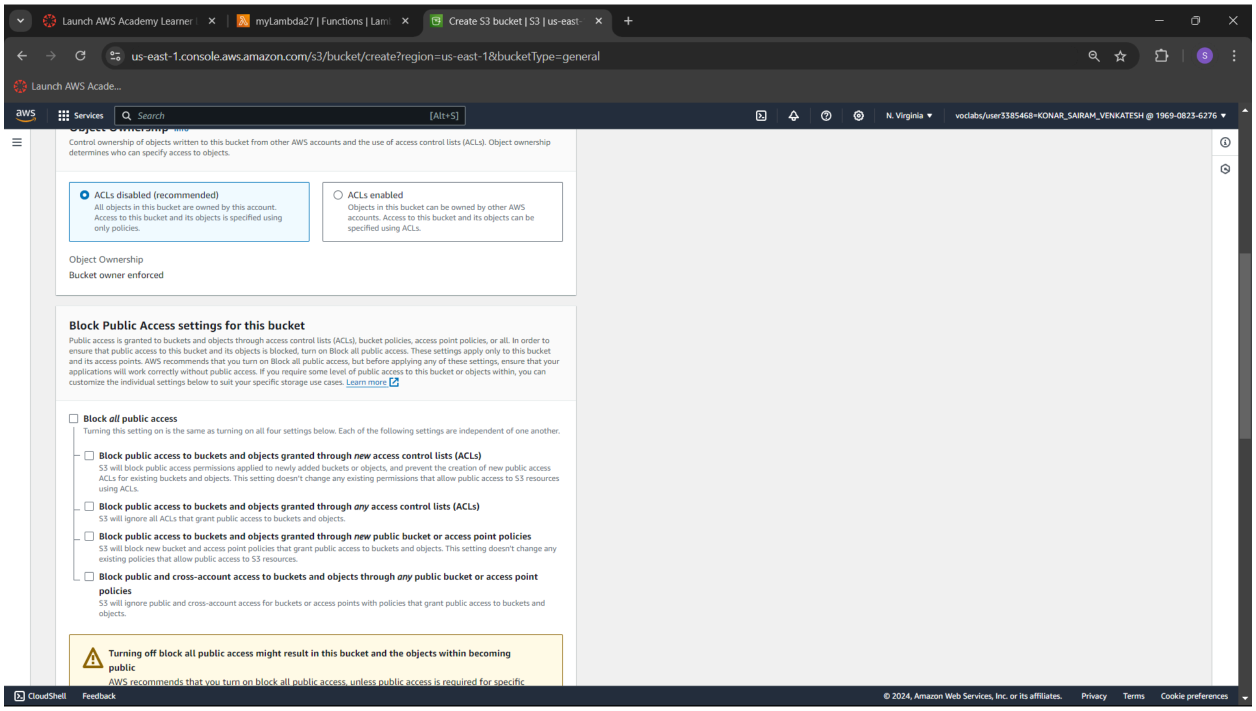
Bucket name: s3ForLambda27

Bucket name must be unique within the global namespace and follow the bucket naming rules.

Object Ownership: ACLs disabled (recommended)

3) Uncheck block all public access.

Sairam Konar D15C 27 2024-25



us-east-1.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

[✓ ACLs disabled (recommended)] All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled — Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

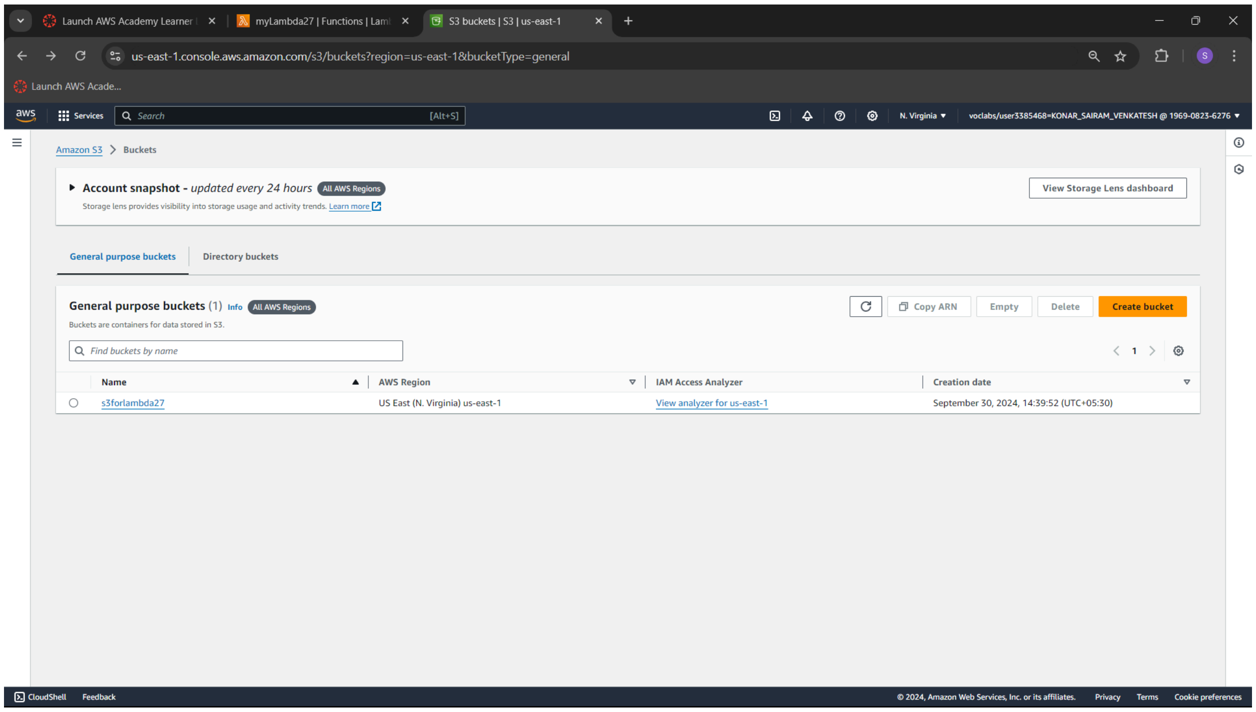
Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

- [ ] **Block *all* public access**
  Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
  - [ ] **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
    S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
  - [ ] **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
    S3 will ignore all ACLs that grant public access to buckets and objects.
  - [ ] **Block public access to buckets and objects granted through *new* public bucket or access point policies**
    S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
  - [ ] **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
    S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific

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



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

Amazon S3 > Buckets

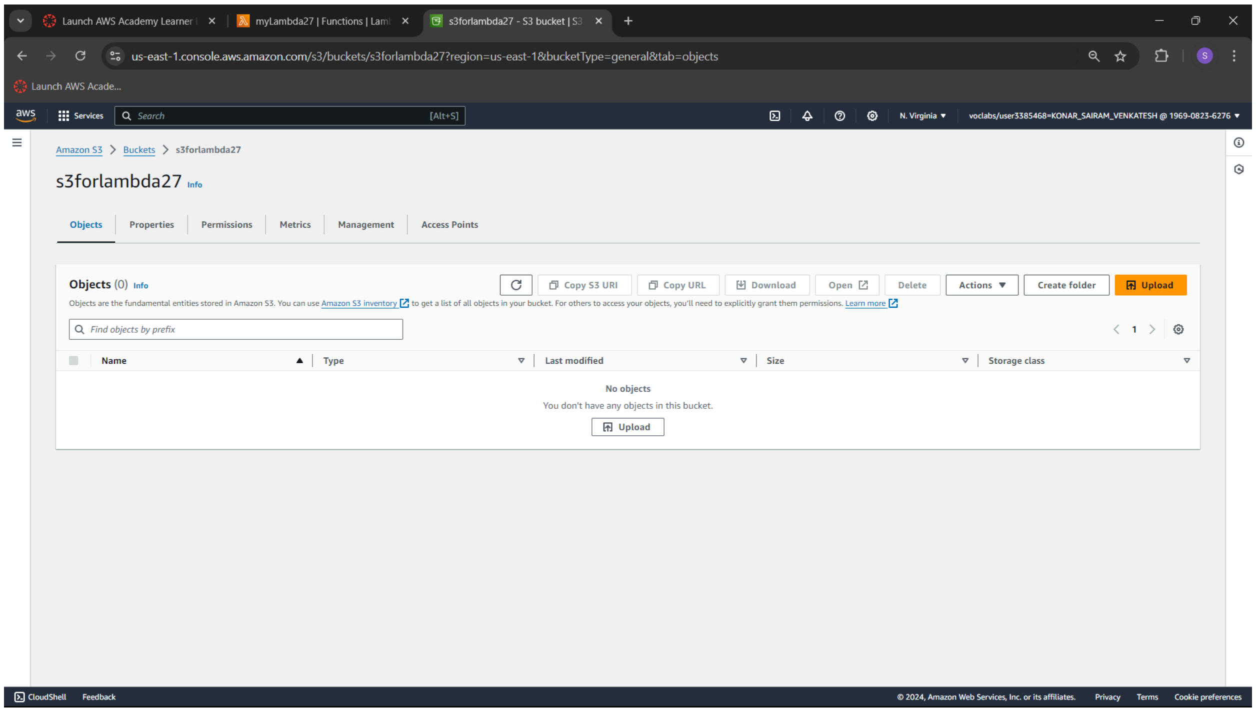
Account snapshot - *updated every 24 hours* All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (1) Info All AWS Regions
Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
s3forlambda27	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 30, 2024, 14:39:52 (UTC+05:30)

Sairam Konar D15C 27 2024-25

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



us-east-1.console.aws.amazon.com/s3/buckets/s3forlambda27?region=us-east-1&bucketType=general&tab=objects

Amazon S3 > Buckets > s3forlambda27

s3forlambda27 Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (0) Info

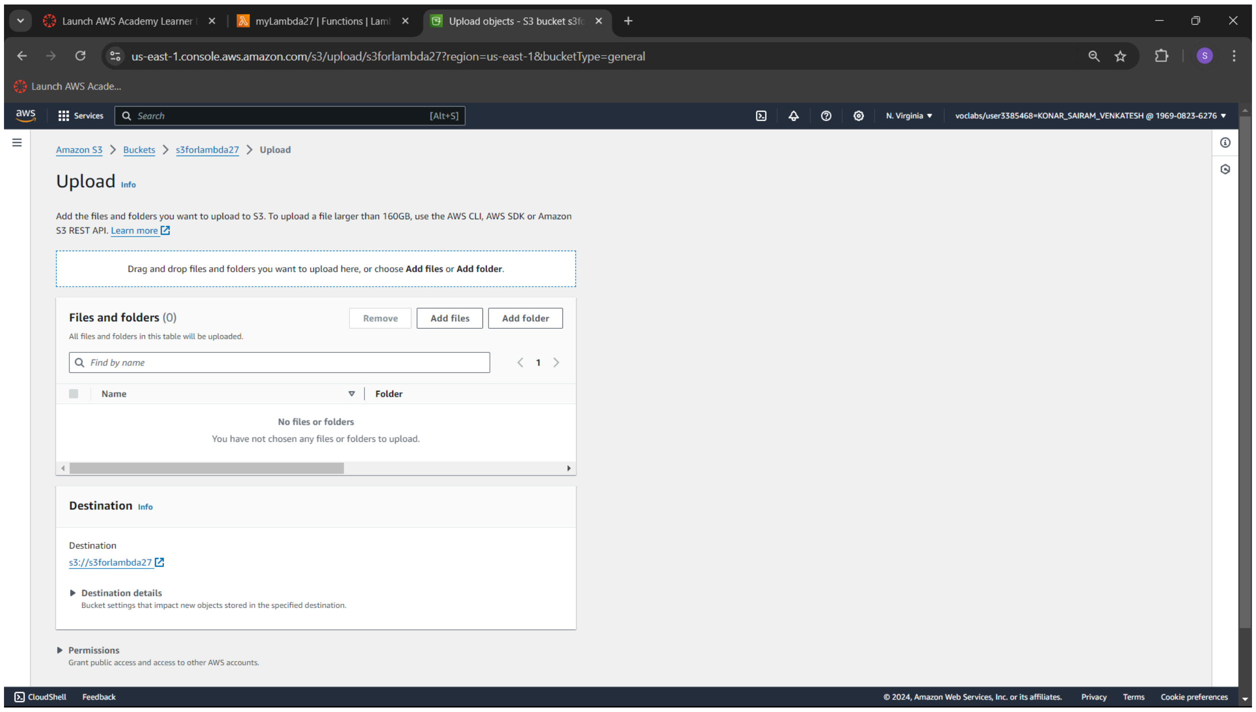
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.

© 2024, Amazon Web Services, Inc. or its affiliates.



us-east-1.console.aws.amazon.com/s3/upload/s3forlambda27?region=us-east-1&bucketType=general

Amazon S3 > Buckets > s3forlambda27 > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (0)

All files and folders in this table will be uploaded.

Name	Folder

No files or folders

You have not chosen any files or folders to upload.

Destination Info

Destination

s3://s3forlambda27

Destination details

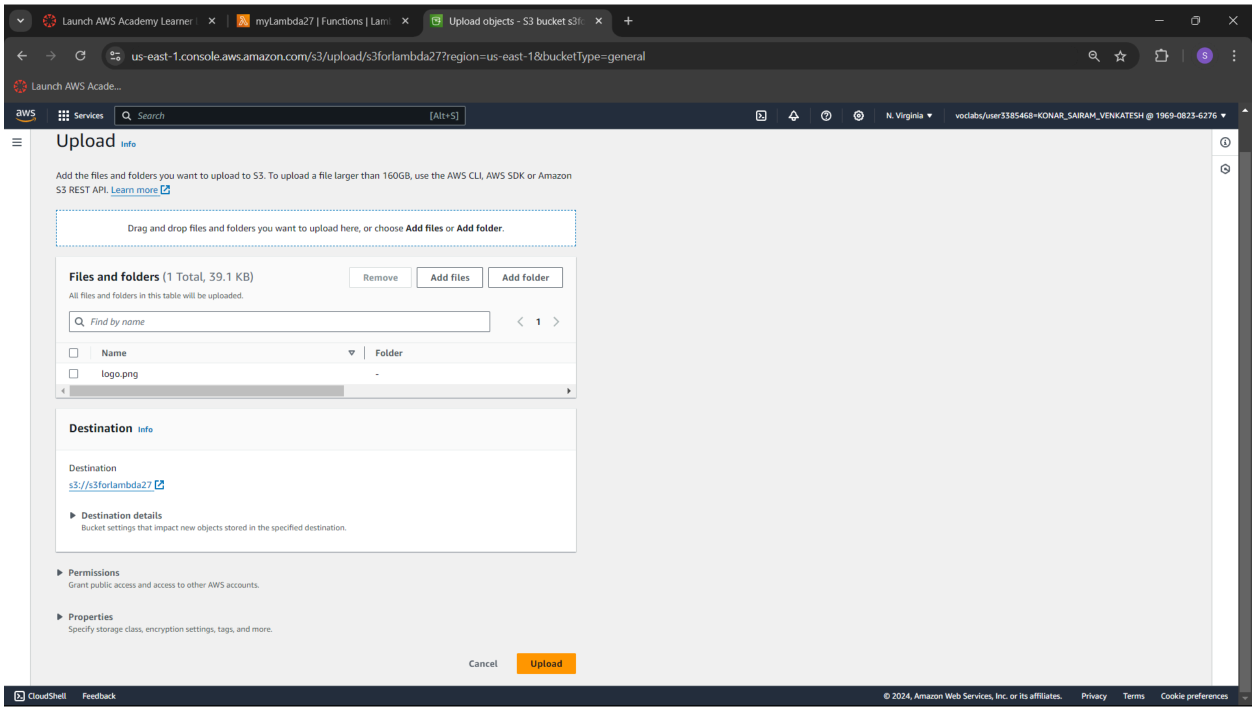
Bucket settings that impact new objects stored in the specified destination.

Permissions

Grant public access and access to other AWS accounts.

© 2024, Amazon Web Services, Inc. or its affiliates.

Sairam Konar D15C 27 2024-25


Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 39.1 KB)

All files and folders in this table will be uploaded.

Name	Folder
logo.png	-

Destination Info

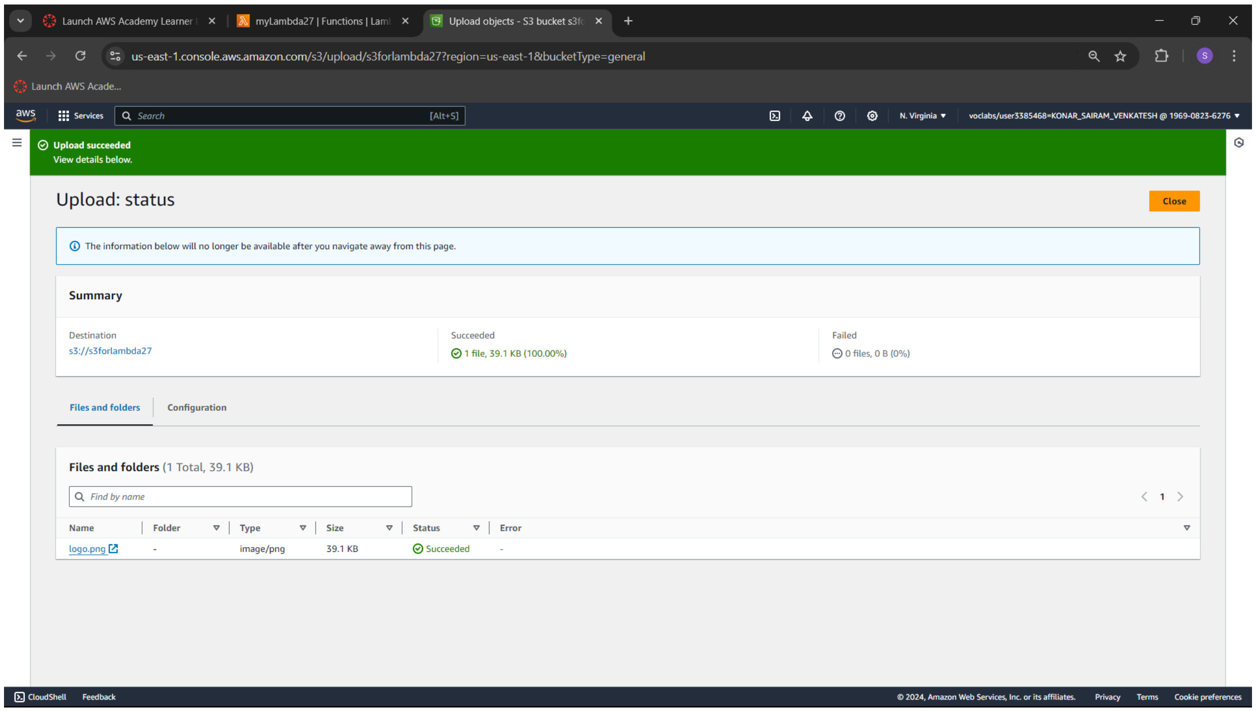
Destination
s3://s3forlambda27

Destination details
Bucket settings that impact new objects stored in the specified destination.

Permissions
Grant public access and access to other AWS accounts.

Properties
Specify storage class, encryption settings, tags, and more.

6) The image has been uploaded to the bucket.


Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://s3forlambda27	1 file, 39.1 KB (100.00%)	0 files, 0 B (0%)

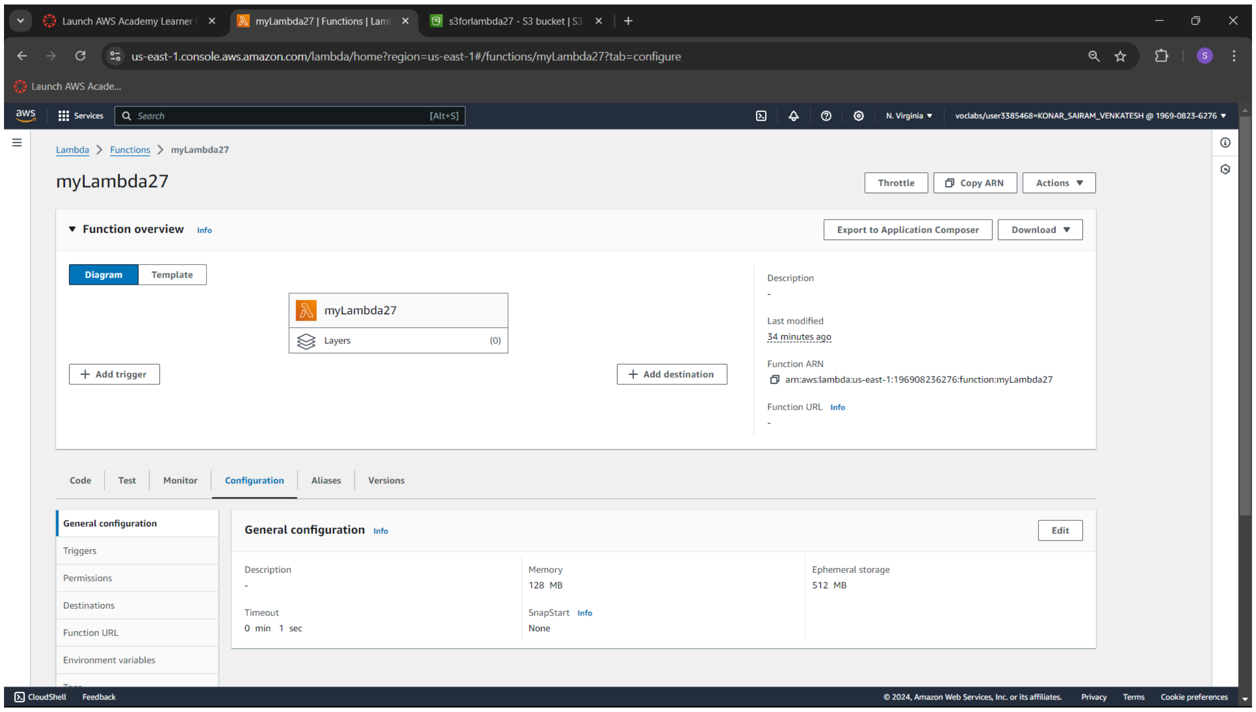
Files and folders (1 Total, 39.1 KB)

Name	Folder	Type	Size	Status	Error
logo.png	-	image/png	39.1 KB	Succeeded	-

Sairam Konar D15C 27 2024-25

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.



Launch AWS Academy Learner
myLambda27 | Functions | Lambda
s3forlambda27 - S3 bucket | S3
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/myLambda27?tab=configure
Services
Search
[Alt+S]
N. Virginia
voclabs/user3385468=KONAR_SAIRAM_VENKATESH @ 1969-0823-6276
Lambda > Functions > myLambda27
myLambda27
Throttle
Copy ARN
Actions
Function overview Info
Export to Application Composer
Download
Diagram
Template
myLambda27
Layers (0)
+ Add trigger
+ Add destination
Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Edit
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Description
-

Last modified
34 minutes ago

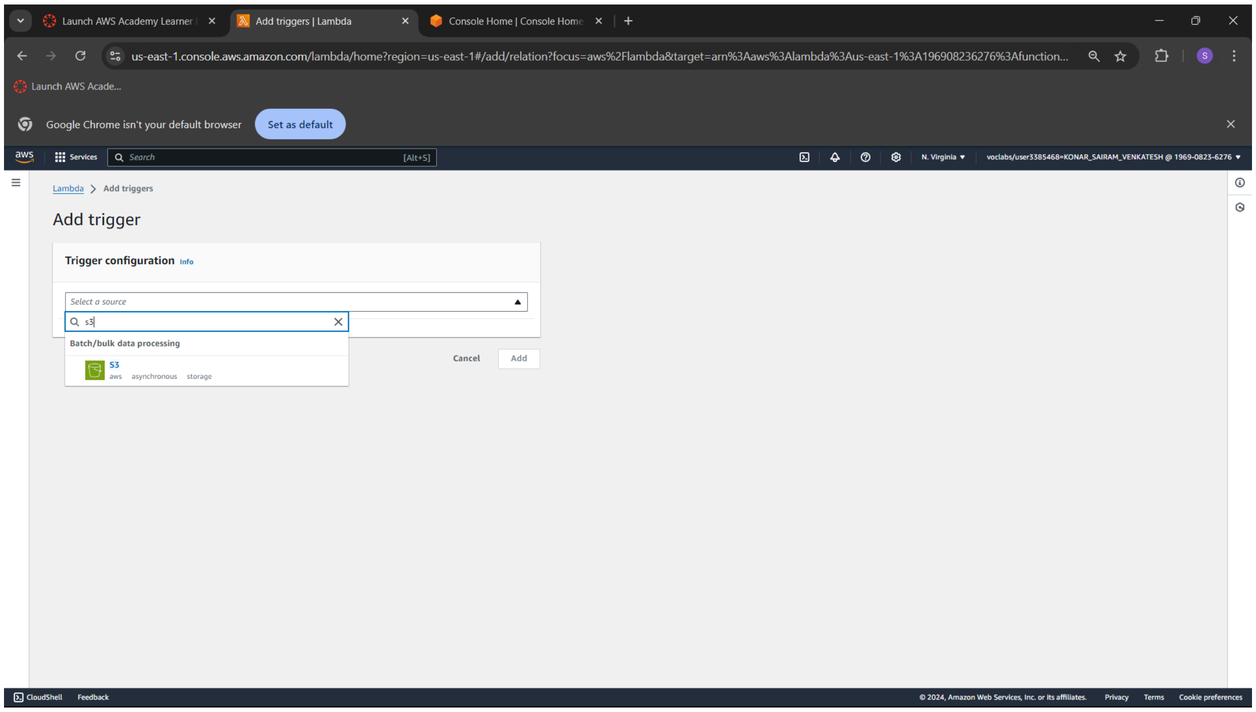
Function ARN
arn:aws:lambda:us-east-1:196908236276:function:myLambda27

Function URL Info
-

General configuration Info

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	SnapStart Info	
0 min 1 sec	None	

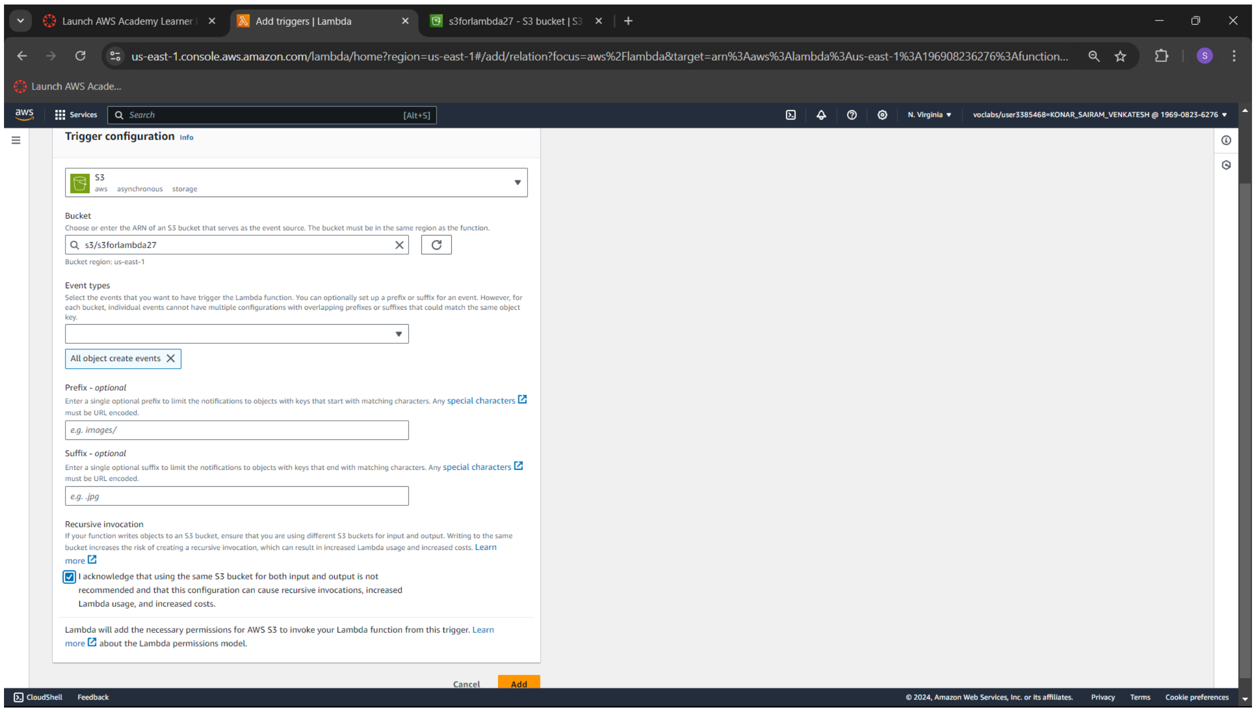
2) Under trigger configuration, search for S3 and select it.



Launch AWS Academy Learner
Add triggers | Lambda
Console Home | Console Home
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/add/relation?focus=aws%2Flambda&target=arn%3Aaws%3Alambda%3Aus-east-1%3A196908236276%3Afunction...
Google Chrome isn't your default browser
Set as default
Services
Search
[Alt+S]
N. Virginia
voclabs/user3385468=KONAR_SAIRAM_VENKATESH @ 1969-0823-6276
Lambda > Add triggers
Add trigger
Trigger configuration Info
Select a source
s3
Batch/bulk data processing
S3
aws asynchronous storage
Cancel
Add
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Sairam Konar D15C 27 2024-25

3) Here, select teh 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.



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/add/relation?focus=aws%2Flambda&target=arn%3Aaws%3Alambda%3Aus-east-1%3A196908236276%3Afunction...

Trigger configuration

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/s3forlambda27
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

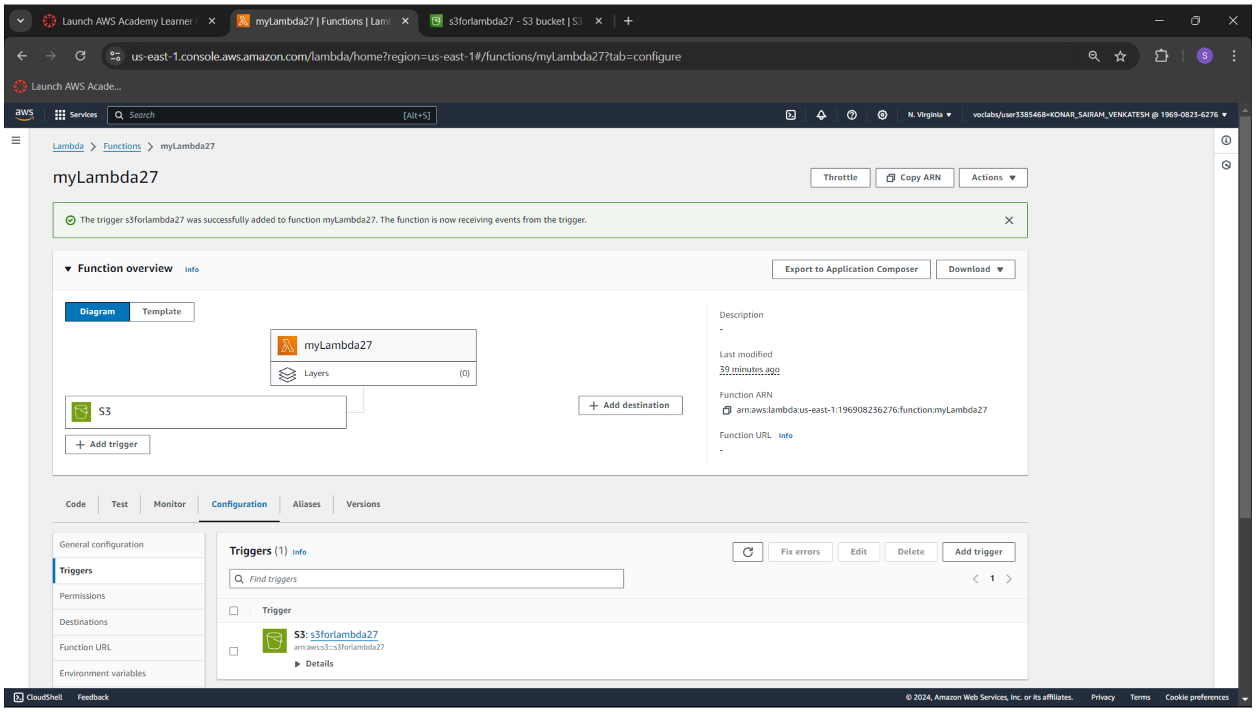
Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.
e.g. images/

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.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/myLambda27?tab=configure

Lambda > Functions > myLambda27

myLambda27

The trigger s3forlambda27 was successfully added to function myLambda27. The function is now receiving events from the trigger.

Function overview

Description: -
Last modified: 39 minutes ago
Function ARN: arn:aws:lambda:us-east-1:196908236276:function:myLambda27
Function URL: -

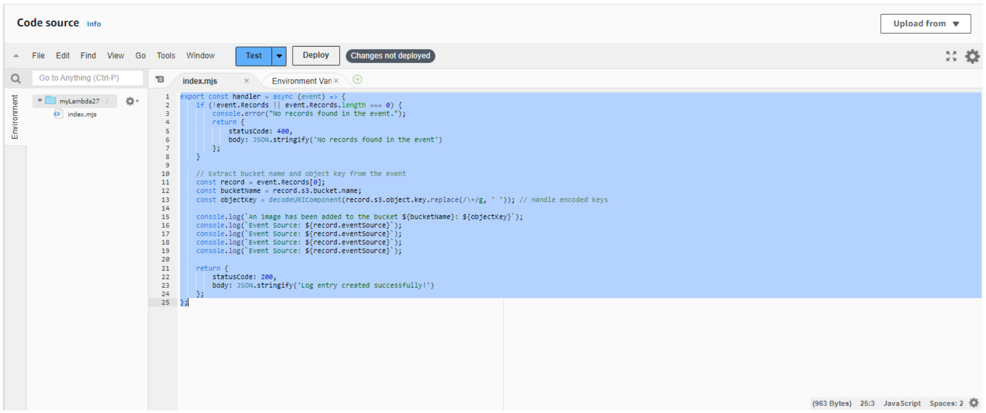
Triggers (1)

Trigger
S3: s3forlambda27 arn:aws:s3:::s3forlambda27

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

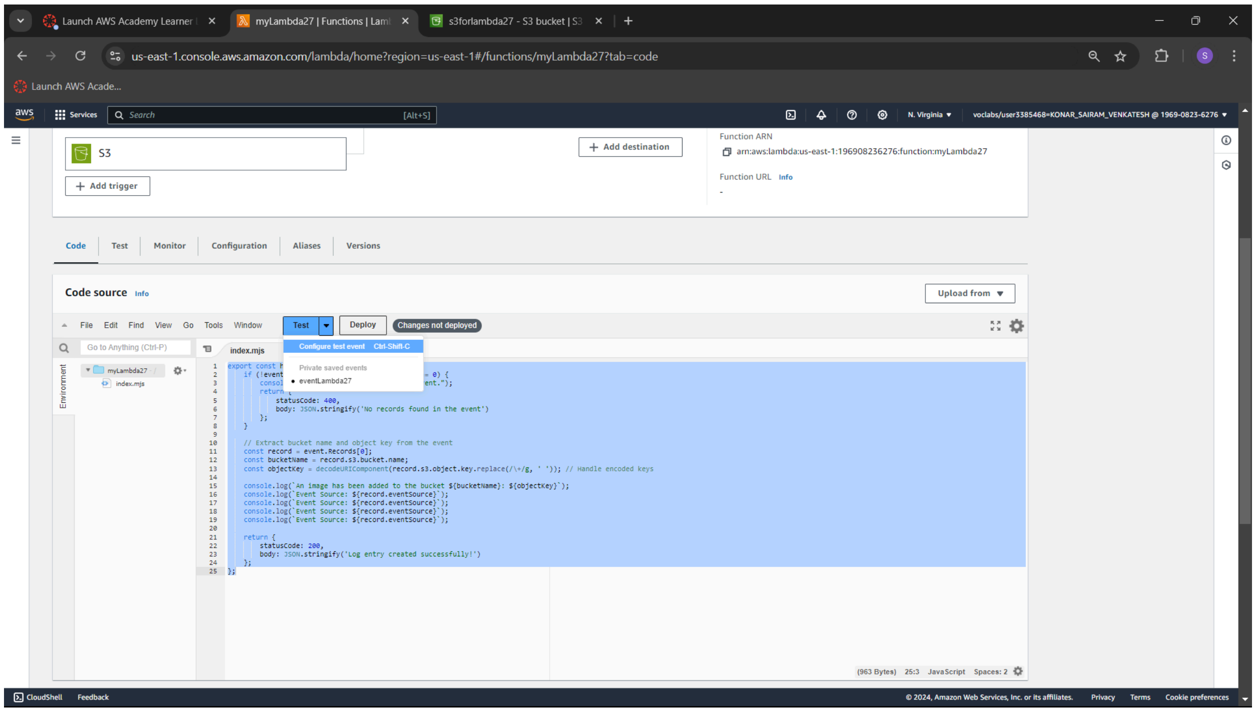
```
export const handler = async (event) => {
  if (!event.Records || event.Records.length === 0) {
    console.error("No records found in the event.");
    return {
      statusCode: 400,
      body: JSON.stringify('No records found in the event')
    };
  }

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

This code checks for records in the event, extracts the bucket name and object key, logs the details, and returns a success message if an image is added to the bucket.



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

5) Now, click on the dropdown near test, then click on configure test event.



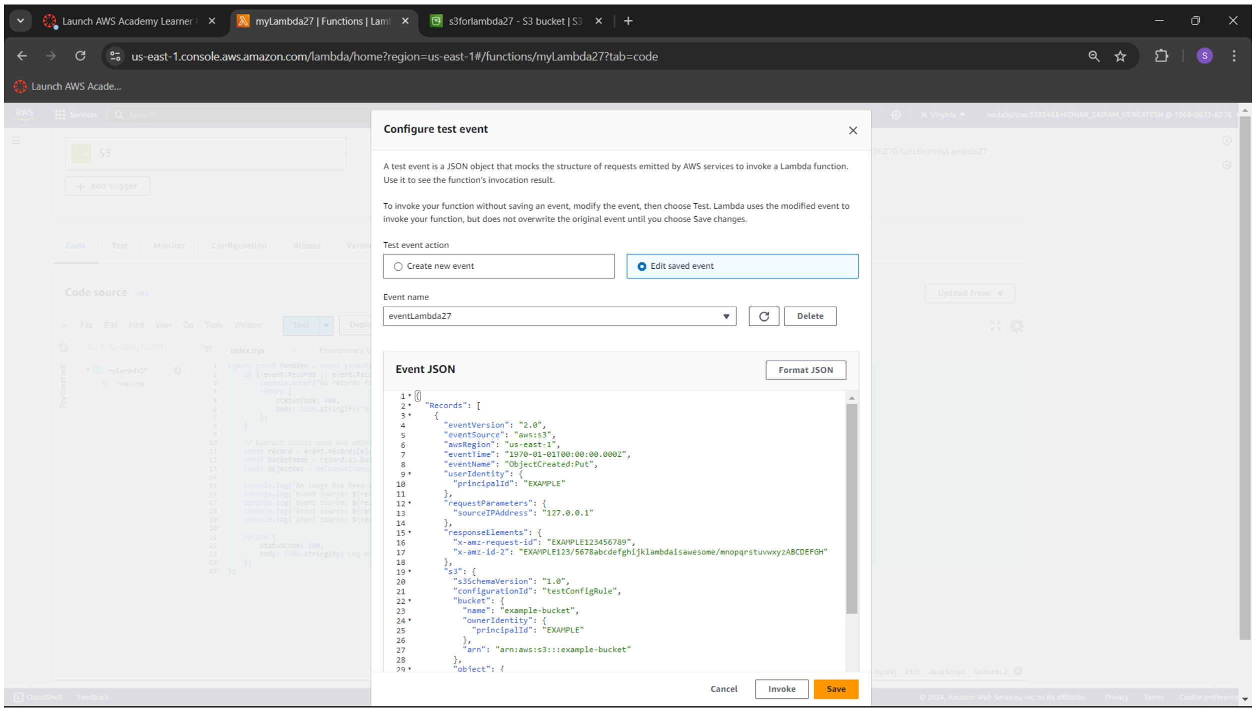
Function ARN
arn:aws:lambda:us-east-1:196908236276:function:myLambda27

```
export const h
    if (!event                                  = 0) {
        consol                                  ent.");
        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!')
    };
};
```

6) Here, select edit saved event. Select the event taht you had created before. Under Event JSON, 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"
        }
      }
    }
  ]
}
```

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.


Launch AWS Academy Learner
myLambda27 | Functions | Lam
s3forlambda27 - S3 bucket | S3
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/myLambda27?tab=code
Launch AWS Acade...

Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action: Create new event [✓ Edit saved event]

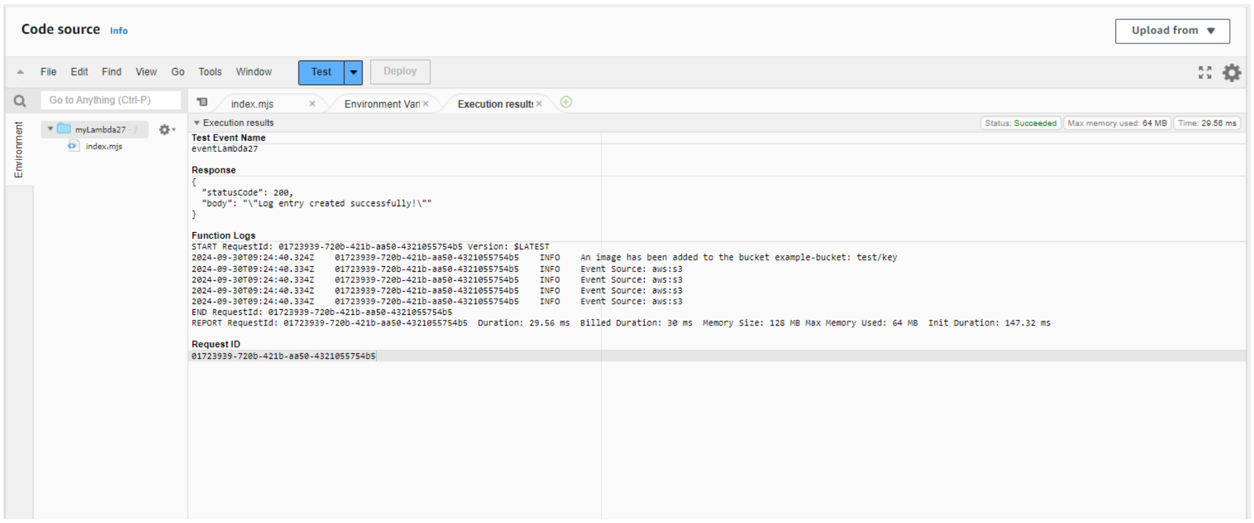
Event name: eventLambda27

Event 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": {
```

Save the changes. Then deploy the code changes by clicking on deploy.

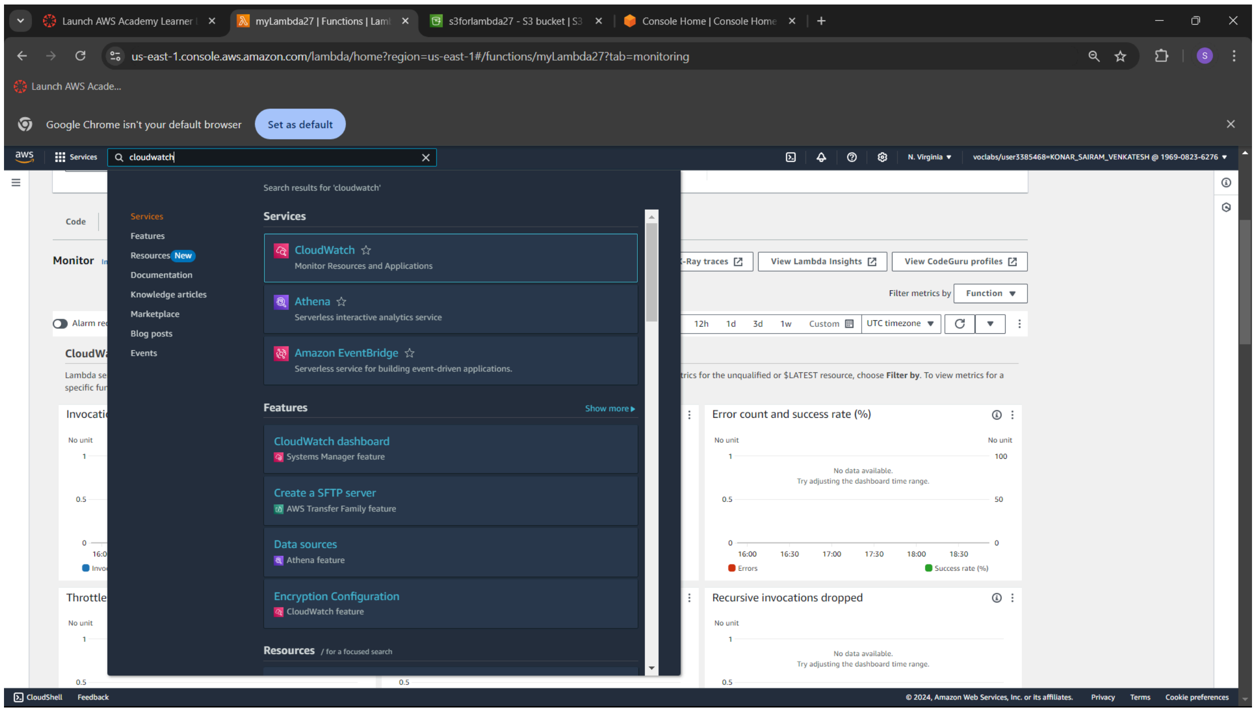
7) After deploying, click on Test. The console output shows that ‘an image has been added to the bucket’
   The JSON response shows that the log entry was created successfully.



```
Execution results
Test Event Name
eventLambda27

Response
{
  "statusCode": 200,
  "body": "\"Log entry created successfully!\""
}

Function Logs
START RequestId: 01723939-720b-421b-aa50-4321055754b5 Version: $LATEST
2024-09-30T09:24:40.324Z    01723939-720b-421b-aa50-4321055754b5    INFO    An image has been added to the bucket example-bucket: test/key
2024-09-30T09:24:40.334Z    01723939-720b-421b-aa50-4321055754b5    INFO    Event Source: aws:s3
2024-09-30T09:24:40.334Z    01723939-720b-421b-aa50-4321055754b5    INFO    Event Source: aws:s3
2024-09-30T09:24:40.334Z    01723939-720b-421b-aa50-4321055754b5    INFO    Event Source: aws:s3
2024-09-30T09:24:40.334Z    01723939-720b-421b-aa50-4321055754b5    INFO    Event Source: aws:s3
END RequestId: 01723939-720b-421b-aa50-4321055754b5
REPORT RequestId: 01723939-720b-421b-aa50-4321055754b5  Duration: 29.56 ms  Billed Duration: 30 ms  Memory Size: 128 MB Max Memory Used: 64 MB  Init Duration: 147.32 ms

Request ID
01723939-720b-421b-aa50-4321055754b5
```

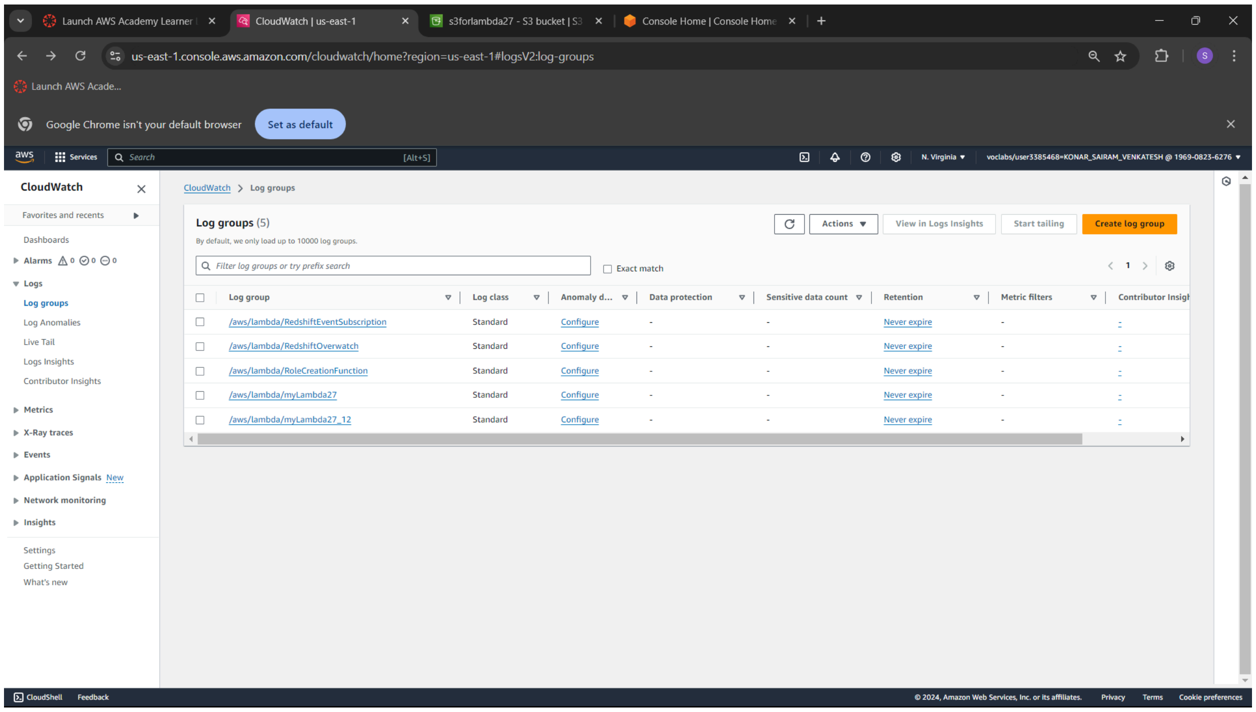
Step 3: Check the logs

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


Launch AWS Academy Learner
myLambda27 | Functions | Lam
s3forlambda27 - S3 bucket | S3
Console Home | Console Home
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/myLambda27?tab=monitoring
Google Chrome isn't your default browser
Set as default
Services
cloudwatch
N. Virginia
voclabs/user3385468=KONAR_SAIRAM_VENKATESH @ 1969-0823-6276
Search results for 'cloudwatch'
Services
Features
Resources New
Documentation
Knowledge articles
Marketplace
Blog posts
Events
CloudWatch
Monitor Resources and Applications
Athena
Serverless interactive analytics service
Amazon EventBridge
Serverless service for building event-driven applications.
Features
Show more
CloudWatch dashboard
Systems Manager feature
Create a SFTP server
AWS Transfer Family feature
Data sources
Athena feature
Encryption Configuration
CloudWatch feature
Resources / for a focused search
View Lambda Insights
View CodeGuru profiles
Filter metrics by
Function
12h 1d 3d 1w Custom
UTC timezone
Error count and success rate (%)
No data available.
Try adjusting the dashboard time range.
Errors
Success rate (%)
Recursive invocations dropped
Code
Monitor
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Sairam Konar D15C 27 2024-25

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



us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:log-groups

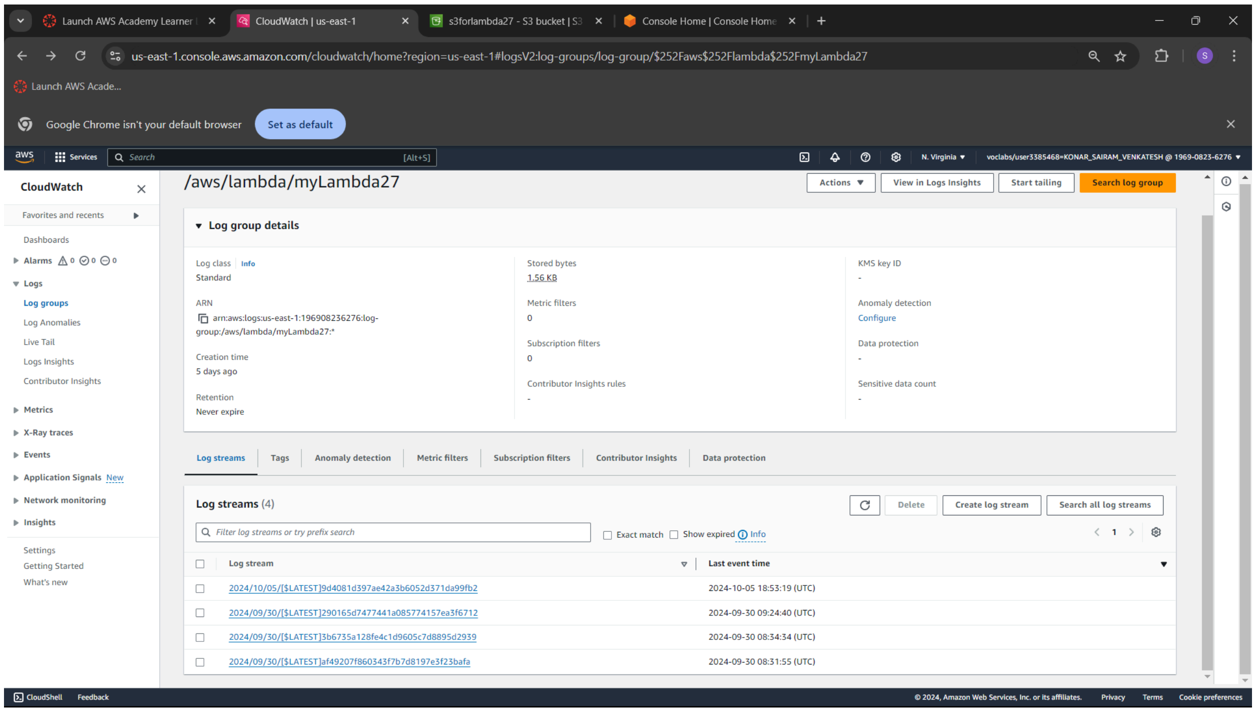
CloudWatch > Log groups

Log groups (5)

By default, we only load up to 10000 log groups.

Log group	Log class	Anomaly d...	Data protection	Sensitive data count	Retention	Metric filters	Contributor Insigh
/aws/lambda/RedshiftEventSubscription	Standard	Configure	-	-	Never expire	-	-
/aws/lambda/RedshiftOverwatch	Standard	Configure	-	-	Never expire	-	-
/aws/lambda/RoleCreationFunction	Standard	Configure	-	-	Never expire	-	-
/aws/lambda/myLambda27	Standard	Configure	-	-	Never expire	-	-
/aws/lambda/myLambda27_12	Standard	Configure	-	-	Never expire	-	-

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



us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:log-groups/log-group/$252Faws$252Flambda$252FmyLambda27

/aws/lambda/myLambda27

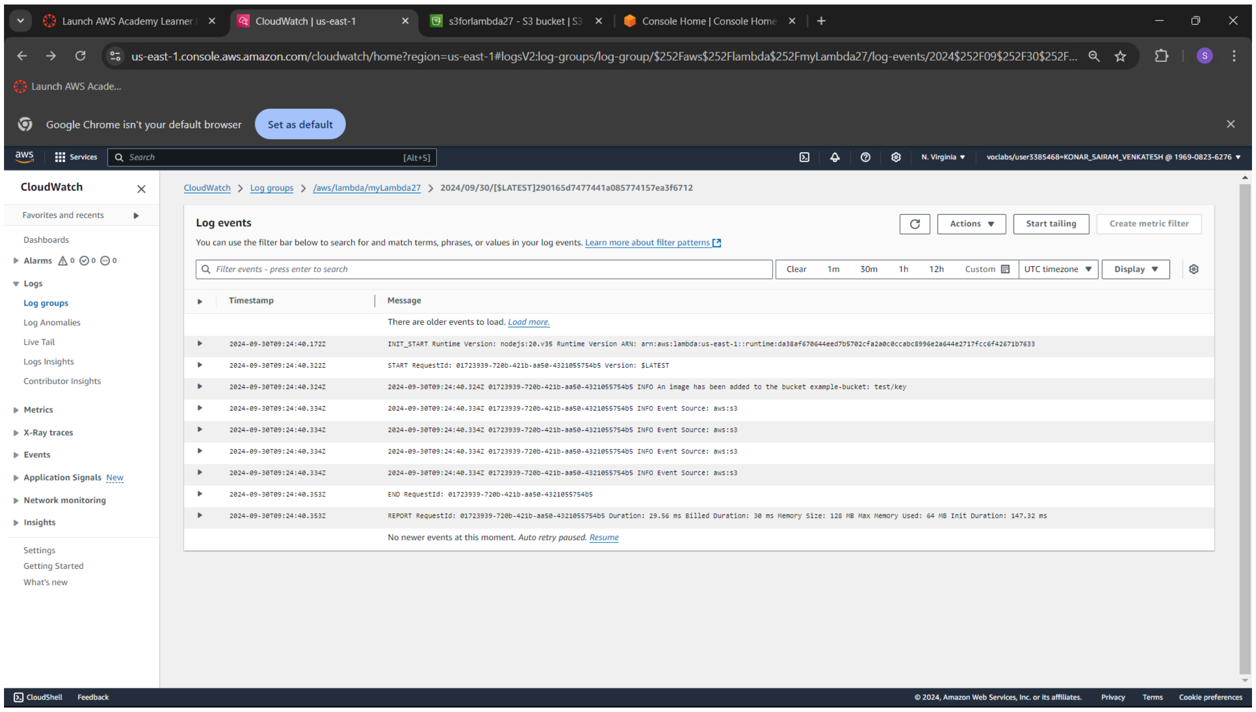
Log group details

Log class: Standard	Stored bytes: 1.56 KB	KMS key ID: -
ARN: arn:aws:logs:us-east-1:196908236276:log-group:/aws/lambda/myLambda27:*	Metric filters: 0	Anomaly detection: Configure
Creation time: 5 days ago	Subscription filters: 0	Data protection: -
Retention: Never expire	Contributor Insights rules: -	Sensitive data count: -

Log streams (4)

Log stream	Last event time
2024/10/05/[$LATEST]9d4081d397ae42a3b6052d371da99fb2	2024-10-05 18:53:19 (UTC)
2024/09/30/[$LATEST]290165d7477441a085774157ea3f6712	2024-09-30 09:24:40 (UTC)
2024/09/30/[$LATEST]3b6735a128fe4c1d9605c7d8895d2939	2024-09-30 08:34:34 (UTC)
2024/09/30/[$LATEST]af49207f860343f7b7d8197e3f23bafa	2024-09-30 08:31:55 (UTC)

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



Log events

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

```
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:da38af670644eed7b5702cfa2a0c0ccabc8996e2a644e2717fcc6f42671b7633
2024-09-30T09:24:40.322Z   START RequestId: 01723939-720b-421b-aa50-4321055754b5 Version: $LATEST
2024-09-30T09:24:40.324Z   2024-09-30T09:24:40.324Z 01723939-720b-421b-aa50-4321055754b5 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-aa50-4321055754b5 INFO Event Source: aws:s3
2024-09-30T09:24:40.334Z   2024-09-30T09:24:40.334Z 01723939-720b-421b-aa50-4321055754b5 INFO Event Source: aws:s3
2024-09-30T09:24:40.334Z   2024-09-30T09:24:40.334Z 01723939-720b-421b-aa50-4321055754b5 INFO Event Source: aws:s3
2024-09-30T09:24:40.334Z   2024-09-30T09:24:40.334Z 01723939-720b-421b-aa50-4321055754b5 INFO Event Source: aws:s3
2024-09-30T09:24:40.353Z   END RequestId: 01723939-720b-421b-aa50-4321055754b5
2024-09-30T09:24:40.353Z   REPORT RequestId: 01723939-720b-421b-aa50-4321055754b5 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 successfully created a Lambda function that logs a message when an image is added to a specific S3 bucket. By configuring an S3 bucket trigger for the Lambda function, we demonstrated how AWS services can work together to automate tasks. The function logged important details about the event, including the bucket name and object key. After deploying the function and testing with a sample event, we verified the logs in CloudWatch, confirming that the Lambda function correctly detected and logged the addition of an image to the bucket. This experiment highlighted the seamless integration between AWS Lambda and S3 for event-driven processes.