



Academic Year: 2022-23

Semester: V

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Subject Lab Incharge: Prof.Manjusha K.

EXPERIMENT NO. 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.

Theory:

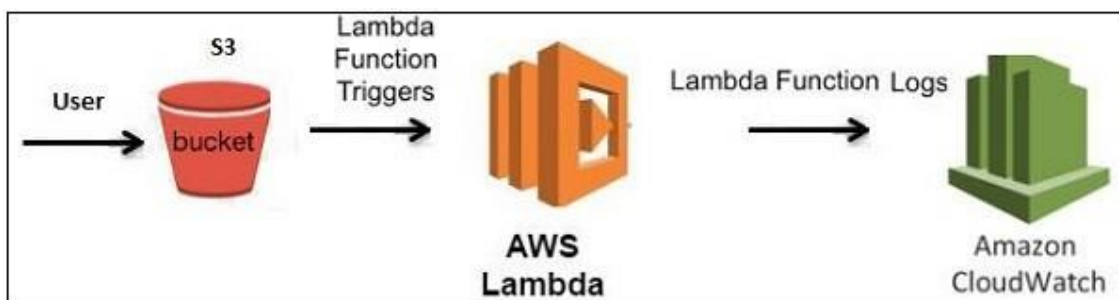
Using Lambda Function with Amazon S3

Amazon S3 service is used for file storage, where you can upload or remove files. We can trigger AWS Lambda on S3 when there are any file uploads in S3 buckets. AWS Lambda has a handler function which acts as a start point for AWS Lambda function. The handler has the details of the events. In this chapter, let us see how to use AWS S3 to trigger AWS Lambda function when we upload files in S3 bucket.

Steps for Using AWS Lambda Function with Amazon S3

To start using AWS Lambda with Amazon S3, we need the following –

- Create S3 Bucket
- Create role which has permission to work with s3 and lambda
- Create lambda function and add s3 as the trigger.





Let us see these steps with the help of an example which shows the basic interaction between Amazon S3 and AWS Lambda

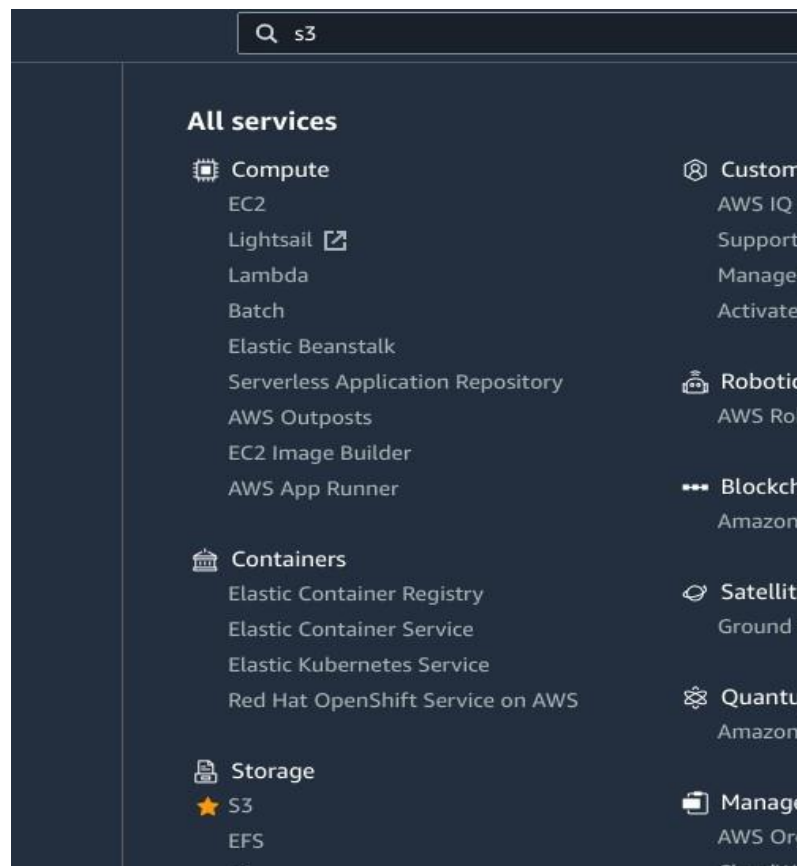
- User will upload a file in Amazon S3 bucket
- Once the file is uploaded, it will trigger AWS Lambda function in the background which will display an output in the form of a console message that the file is uploaded.
- The user will be able to see the message in Cloudwatch logs once the file is uploaded.

Creating S3 Bucket

Let us start first by creating a s3 bucket in AWS console using the steps given below –

Step 1

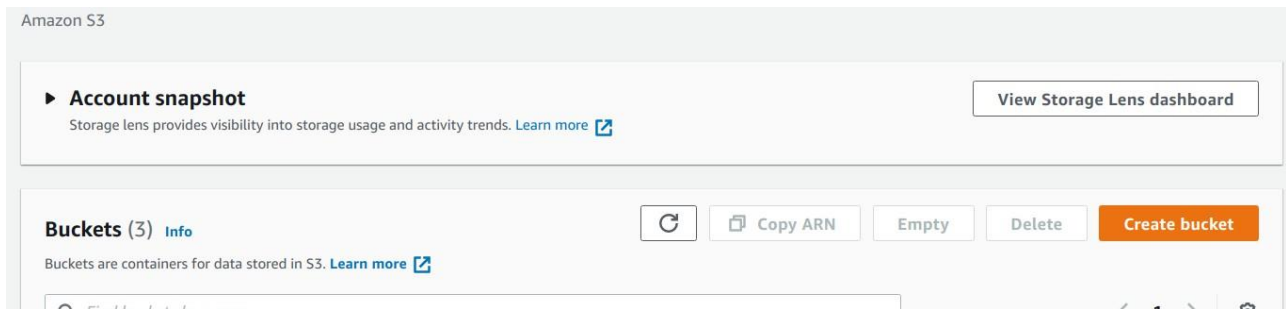
Go to Amazon services and click **S3** in storage section as highlighted in the image given below –





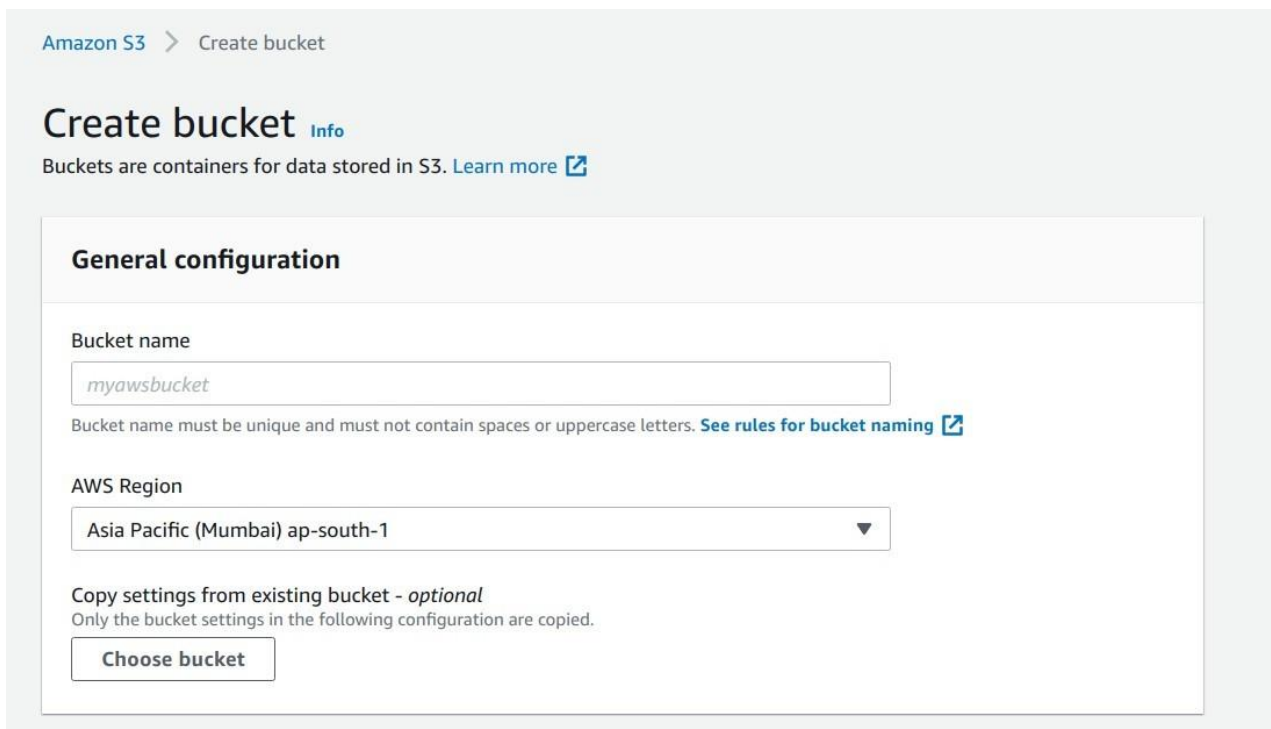
Step 2

Click **S3** storage and **Create bucket** which will store the files uploaded.



Step 3

Once you click Create bucket button, you can see a screen as follows –





Step 4

Enter the details Bucket name, Select the Region and click Create button at the bottom left side. Thus, we have created bucket with name :

<input type="radio"/>	lambdawiths3	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	August 3, 2021, 11:22:23 (UTC+05:30)
-----------------------	--------------	----------------------------------	-------------------------------	--------------------------------------

Step 5

Now, click the bucket name and it will ask you to upload files as shown below –

lambdawiths3 [Info](#)

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

Objects (0)

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
No objects					

Thus, we are done with bucket creation in S3.

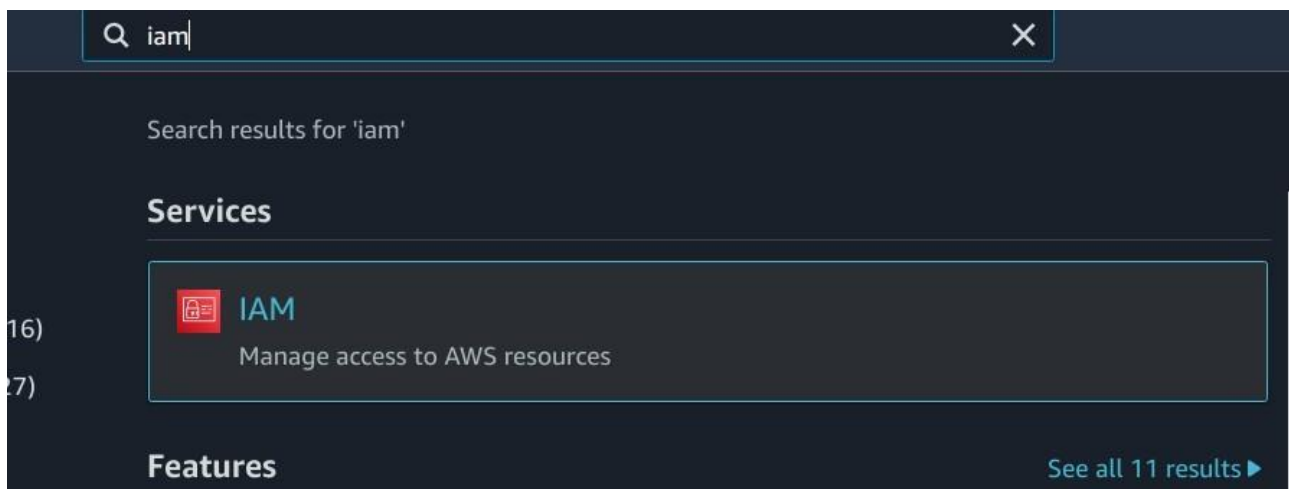


Create Role that Works with S3 and Lambda

To create role that works with S3 and Lambda, please follow the Steps given below –

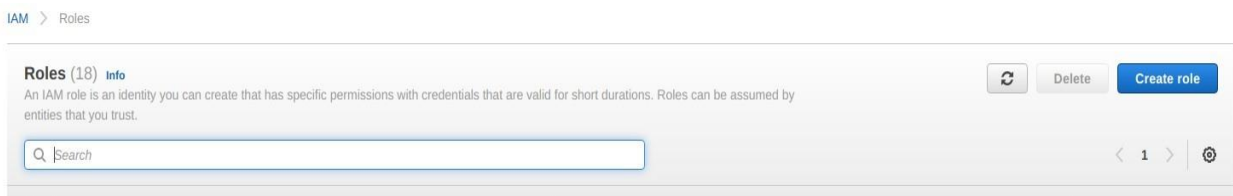
Step 1

Go to AWS services and select IAM as shown below –



Step 2

Now, click **IAM -> Roles** as shown below –








Step 3

Now, click **Create role** and choose the services that will use this role. Select Lambda and click **Permission** button.

Create role

1 2 3 4

Select type of trusted entity

 AWS service EC2, Lambda and others	 Another AWS account Belonging to you or 3rd party	 Web identity Cognito or any OpenID provider	 SAML 2.0 federation Your corporate directory
--	---	---	--

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeBuild	EMR Containers	IoT SiteWise	RDS
AWS Backup	CodeDeploy	ElasticCache	IoT Things Graph	Redshift
AWS Chatbot	CodeGuru	Elastic Beanstalk	KMS	Rekognition
AWS Marketplace	CodeStar Notifications	Elastic Container Registry	Kinesis	RoboMaker
AWS Support	Comprehend	Elastic Container Service	Lake Formation	S3
Amplify	Config	Elastic Transcoder	Lambda	SMS
AppStream 2.0	Connect	ElasticLoadBalancing	Lex	SNS
AppSync	DMS	EventBridge	License Manager	SWF
Application Auto Scaling	Data Lifecycle Manager	Forecast	MQ	SageMaker
Application Discovery	Data Pipeline	GameLift	Machine Learning	Security Hub

* Required

Cancel

Next: Permissions



Step 4

Add the permission from below and click Review.

AmazonS3FullAccess, AWSLambdaFullAccess and CloudWatchFullAccess.

Step 5

Observe that we have chosen the following permissions –

Create role

1

2

3

Review

Provide the required information below and review this role before you create it.

Role name*

Use alphanumeric and '+=, @-_' characters. Maximum 64 characters.

Role description

Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.

Trusted entities AWS service: lambda.amazonaws.com

Policies



AmazonS3FullAccess [↗](#)



AWSLambda_FullAccess [↗](#)



CloudWatchFullAccess [↗](#)

Permissions boundary Permissions boundary is not set

No tags were added.

Observe that the Policies that we have selected are **AmazonS3FullAccess**, **AWSLambdaFullAccess** and **CloudWatchFullAccess**.



Step 6

Now, enter the Role name, Role description and click Create Role button at the bottom.

<input type="checkbox"/>	lambdawiths3service	AWS Service: lambda
--------------------------	---------------------	---------------------

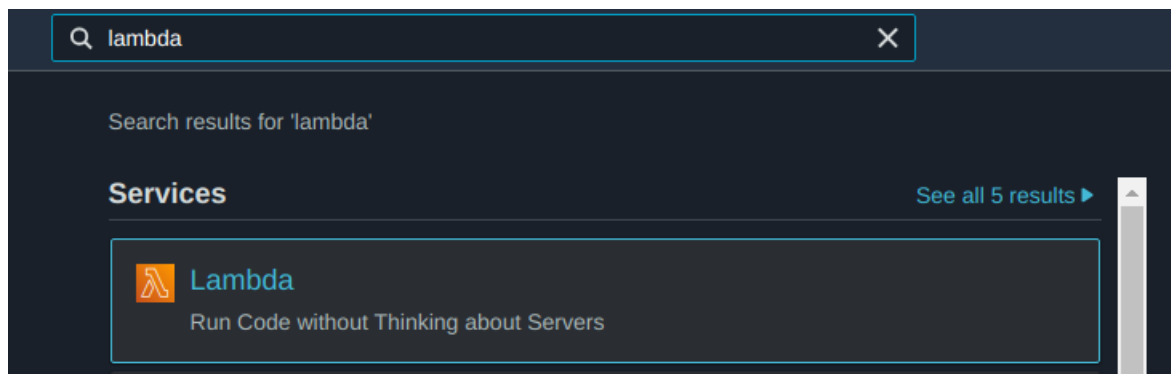
Thus, our role named lambdawiths3service is created.

Create Lambda function and Add S3 Trigger

In this section, let us see how to create a Lambda function and add a S3 trigger to it. For this purpose, you will have to follow the Steps given below –

Step 1

Go to AWS Services and select Lambda as shown below –





Step 2

Click **Lambda** and follow the process for adding **Name**. Choose the **Runtime**, **Role** etc. and create the function. The Lambda function that we have created is shown in the screenshot below –

Lambda > Functions > Create function

Create function [Info](#)

Choose one of the following options to create your function.

Author from scratch ☒
Start with a simple Hello World example.

Use a blueprint ☐
Build a Lambda application from sample code and configuration presets for common use cases.

Basic information

Function name
Enter a name that describes the purpose of your function.

lambdawiths3bucket

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Node.js 14.x

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions
☒ Use an existing role
☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

lambdawiths3service

[View the lambdawiths3service role](#) on the IAM console.

Step 3

Now let us add the S3 trigger.



Lambda > Functions > lambdawiths3bucket

lambdawiths3bucket Throttle Copy ARN Actions

Function overview [Info](#)

Code | Test | Monitor | **Configuration** | Aliases | Versions

General configuration

Triggers

Permissions

Destinations

Environment variables

Tags

VPC

Triggers (0) Refresh Enable Disable Fix errors Delete Add trigger

☐ Trigger

No triggers
No triggers are configured.


Add trigger

Step 4

Choose the trigger from above and add the details as shown below –

Add trigger

Trigger configuration

 **S3**
aws storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.


Refresh

Event type
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.

Prefix - optional
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

Suffix - optional
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

 **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.

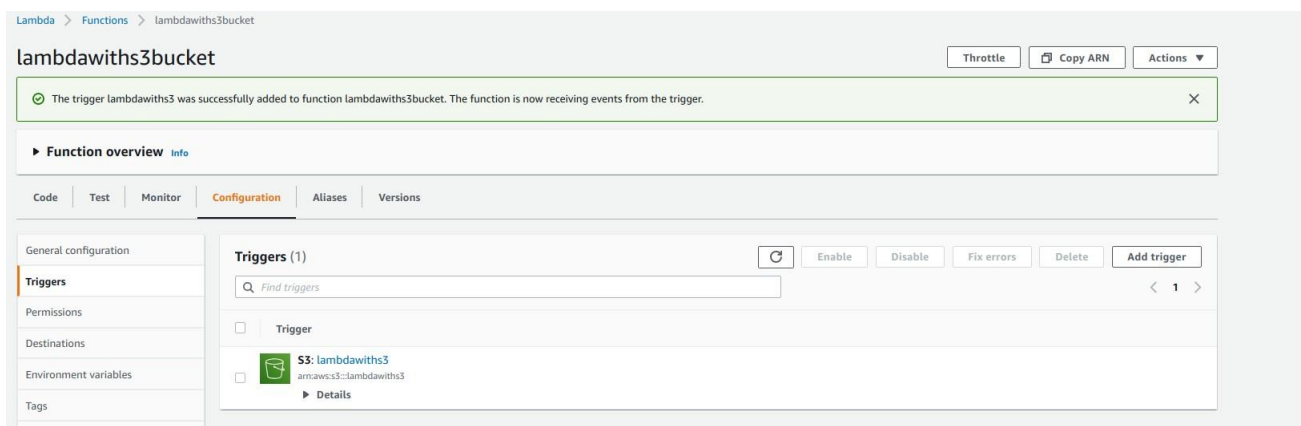
Cancel Add



You can add Prefix and File pattern which are used to filter the files added. For Example, to trigger lambda only for .jpg images. as we need to trigger Lambda for all jpg image files uploaded. Click Add button to add the trigger.

Step 5

You can find the the trigger display for the Lambda function as shown below –



Step 6

Let's add the details for the aws lambda function. Here, we will use the online editor to add our code and use nodejs as the runtime environment.

To trigger S3 with AWS Lambda, we will have to use S3 event in the code as shown below –



Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE
(All Programs Accredited by NBA)
Department of Information Technology



Lambda > Functions > lambdawiths3bucket

lambdawiths3bucket

Throttle Copy ARN Actions

The trigger lambdawiths3 was successfully added to function lambdawiths3bucket. The function is now receiving events from the trigger.

Function overview

Code Test Monitor Configuration Aliases Versions

Code source

Upload from

```
1 exports.handler = function(event, context, callback) {
2   console.log("Incoming Event: ", event);
3   const bucket = event.Records[0].s3.bucket.name;
4   const filename = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
5   const message = `An Image has been added - ${bucket} -> ${filename}`;
6   console.log(message);
7   callback(null, message);
8 };
```

8:3 JavaScript Spaces: 4

Code properties

Package size 304.0 byte	SHA256 hash uTJfXT0sQY0Bf6CtzoZbCLT6Hd0A48LniMm4gpXgDw=	Last modified August 3, 2021, 11:36 AM GMT+5:30
----------------------------	--	--

Runtime settings

Runtime Node.js 14.x	Handler index.handler
-------------------------	--------------------------

Step 7:

let us save the changes and test the lambda function with S3upload.



Step 8:

Now, save the Lambda function. Open S3 from Amazon services and open the bucket we created earlier namely lambdawiths3.

Upload the image in it as shown below –

Click **Add files** to add files. You can also drag and drop the files. Now, click **Upload** button.

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 folders**.

Files and folders (1 Total, 44.0 KB)
All files and folders in this table will be uploaded.

Remove

Add files

Add folder

< 1 >

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	apsit_logo.jpg	-	image/jpeg	44.0 KB

Thus, we have uploaded one image in our S3 bucket.

Step 9

To see the trigger details, go to AWS service and select CloudWatch. Open the logs for the Lambda AWS Lambda function gets triggered when file is uploaded in S3 bucket and the details are logged in Cloudwatch as shown below –



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



CloudWatch > Log groups > /aws/lambda/lambda3bucket > 2021/08/03/[\$LATEST]0f36a60d46ca40078172fc11de9d735f

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](#)

1m

Timestamp	Message
No older events at this moment. Retry	
2021-08-03T12:01:00.069+05:30	START RequestId: ae43508a-8eb7-4b08-8fa1-841814d597c1 Version: \$LATEST
2021-08-03T12:01:00.098+05:30	2021-08-03T06:31:00.097Z ae43508a-8eb7-4b08-8fa1-841814d597c1 INFO Incoming Event: { Records: [{ eventVersion: '2.1', eventSource: 'aws:s3', awsRegion: 'ap-south-1',
2021-08-03T12:01:00.098+05:30	2021-08-03T06:31:00.098Z ae43508a-8eb7-4b08-8fa1-841814d597c1 INFO An Image has been added - lambda3bucket -> apsit_logo.jpg
2021-08-03T06:31:00.098Z	ae43508a-8eb7-4b08-8fa1-841814d597c1 INFO An Image has been added - lambda3bucket -> apsit_logo.jpg
2021-08-03T12:01:00.119+05:30	END RequestId: ae43508a-8eb7-4b08-8fa1-841814d597c1
2021-08-03T12:01:00.119+05:30	REPORT RequestId: ae43508a-8eb7-4b08-8fa1-841814d597c1 Duration: 49.40 ms Billed Duration: 50 ms Memory Size: 128 MB Max Memory Used: 65 MB Init Duration: 155.37 ms
No newer events at this moment. Auto retry paused. Resume	

An image has been Added -> apsit_logo.jpg you can see in cloudwatch logs.

Conclusion: Write your own findings.