

Advance Devops-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:

AWS Lambda and S3 Integration: AWS Lambda allows you to execute code in response to various events, including those triggered by Amazon S3. When an object is added to an S3 bucket, it can trigger a Lambda function to execute, allowing for event-driven processing without managing servers.

Workflow:

1. Create an S3 Bucket:

First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

2. Create the Lambda Function:

Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java.

Write code that logs a message like “An Image has been added” when triggered.

3. Set Up Permissions:

Ensure that the Lambda function has the necessary permissions to access S3.

You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.

4. Configure S3 Trigger:

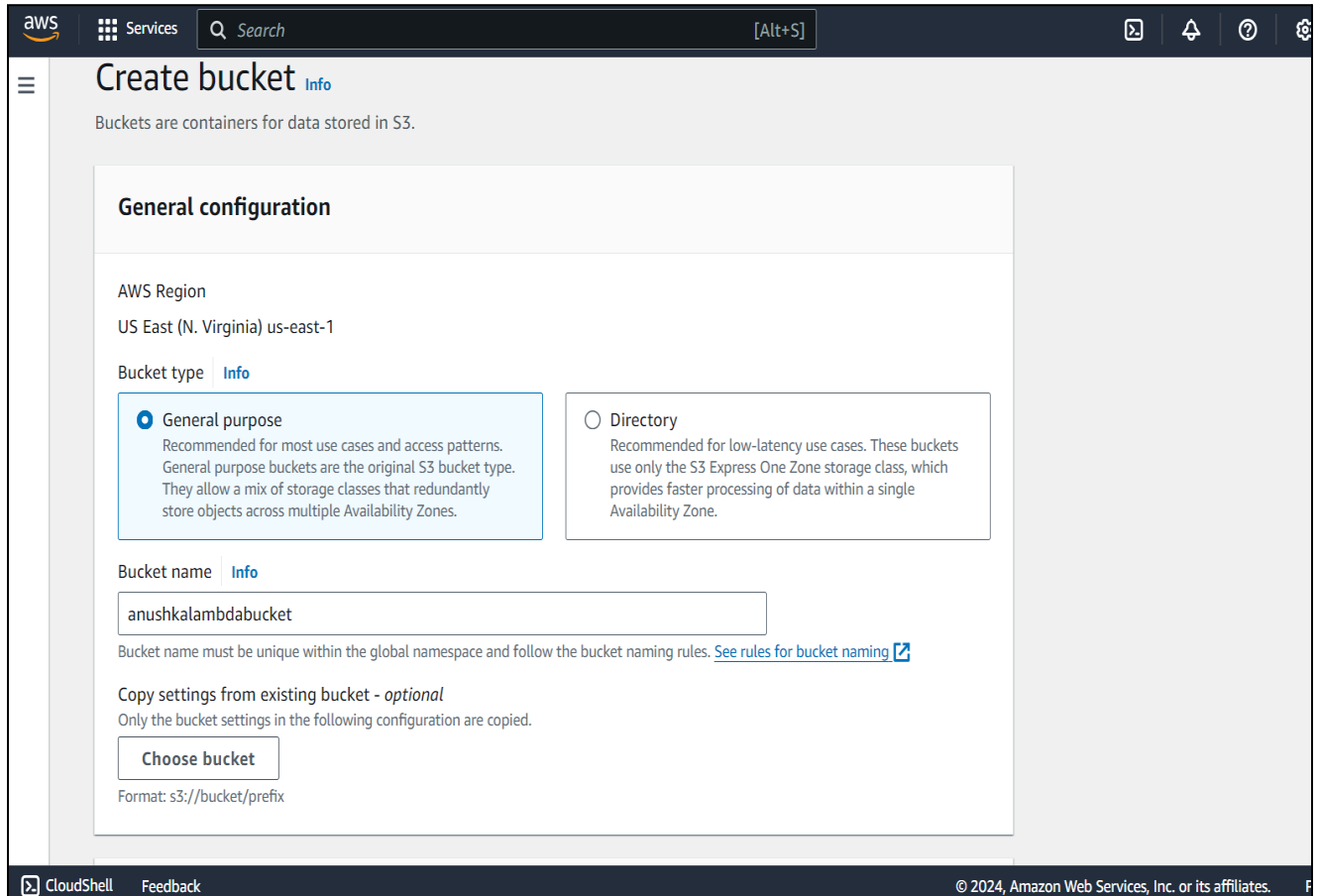
Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

5. Test the Setup:

Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.

1. Create an S3 Bucket:

First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.



The screenshot shows the AWS Management Console interface for creating a new S3 bucket. The top navigation bar includes the AWS logo, 'Services', a search bar, and utility icons. The main heading is 'Create bucket' with an 'Info' link. Below this, a sub-header states 'Buckets are containers for data stored in S3.' The configuration section is titled 'General configuration'. It shows the 'AWS Region' as 'US East (N. Virginia) us-east-1'. Under 'Bucket type', there are two options: 'General purpose' (selected with a radio button) and 'Directory'. The 'General purpose' option is highlighted with a blue border and contains descriptive text. The 'Directory' option is also described. Below these, the 'Bucket name' field is populated with 'anushkalamdbabucket'. A note indicates the name must be unique and follows naming rules, with a link to 'See rules for bucket naming'. There is a section for 'Copy settings from existing bucket - optional' with a 'Choose bucket' button. At the bottom, the format 's3://bucket/prefix' is shown. The footer includes 'CloudShell', 'Feedback', and a copyright notice for Amazon Web Services, Inc. or its affiliates.

aws Services Search [Alt+S]

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

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

Bucket type [Info](#)

☒ **General purpose**
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

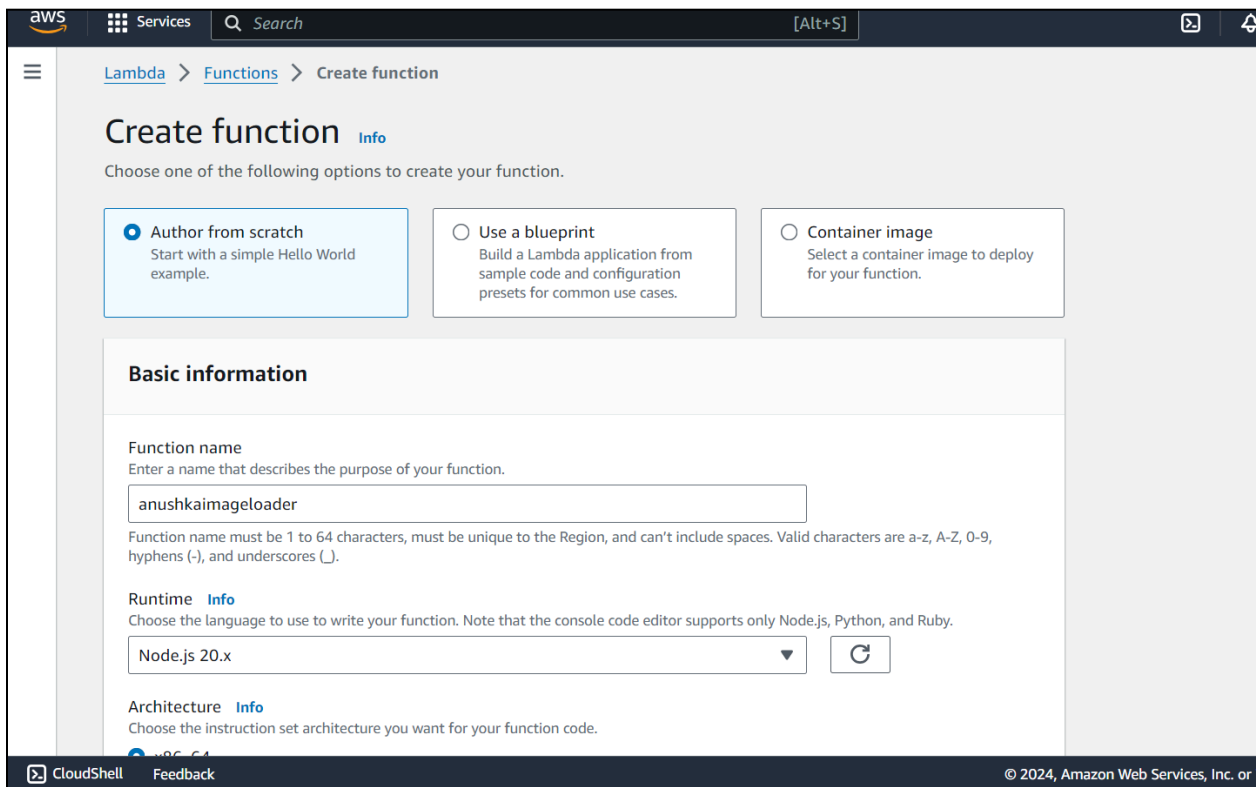
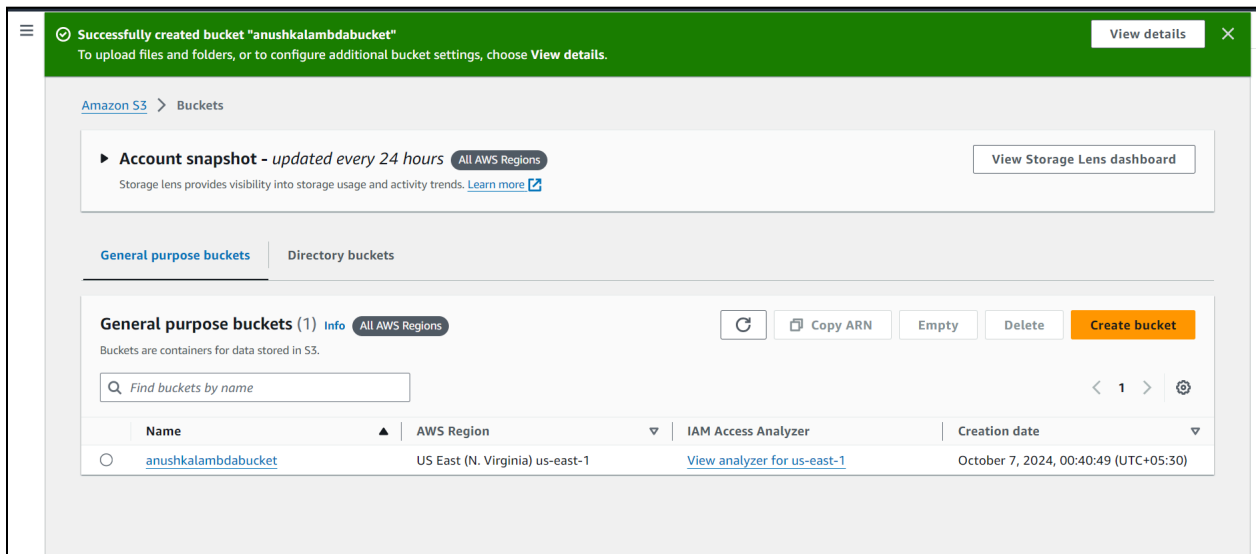
Format: s3://bucket/prefix

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

2. Create the Lambda Function:

Set up a new Lambda function using AWS Lambda's console. You can choose a runtime environment like Python, Node.js, or Java.

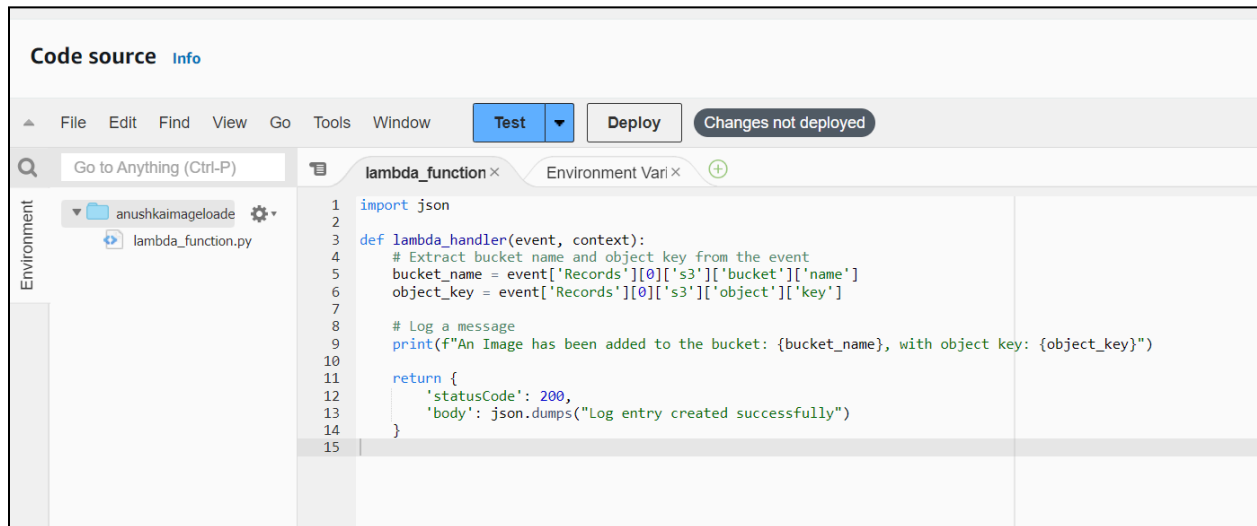
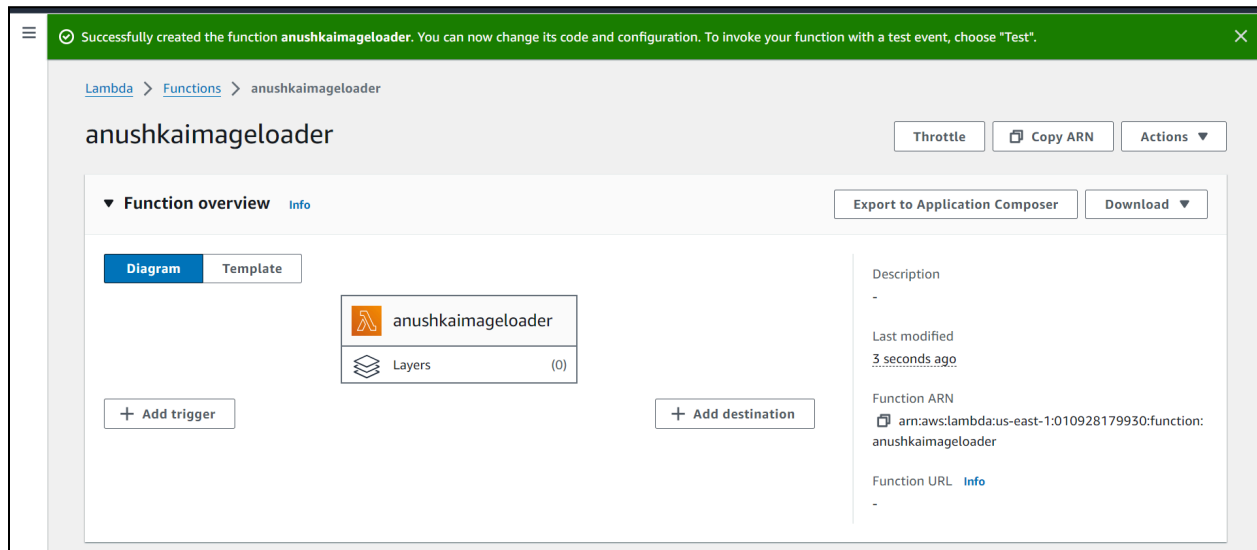
Write code that logs a message like "An Image has been added" when triggered.



3. Set Up Permissions:

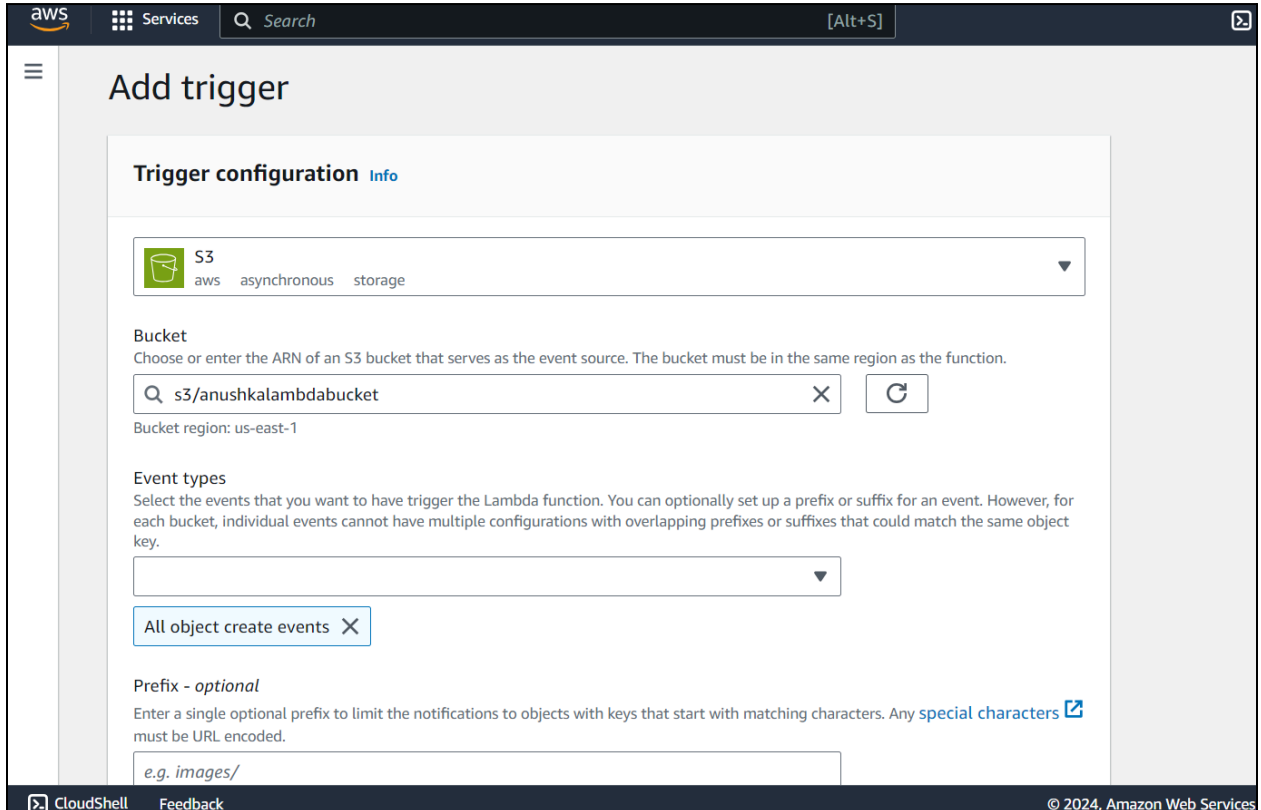
Ensure that the Lambda function has the necessary permissions to access S3.

You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.



4. Configure S3 Trigger:

Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).



The screenshot shows the AWS Lambda console's 'Add trigger' page. The top navigation bar includes the AWS logo, 'Services', a search bar, and a '[Alt+S]' shortcut. The main heading is 'Add trigger'. Below it, the 'Trigger configuration' section is active, showing 'S3' as the selected service with sub-labels 'aws', 'asynchronous', and 'storage'. The 'Bucket' section instructs the user to choose or enter the ARN of an S3 bucket in the same region as the function. A search bar contains 's3/anushkalambdabucket', with a 'Bucket region: us-east-1' label below it. The 'Event types' section explains that users can select events to trigger the function, with a dropdown menu currently showing 'All object create events'. The 'Prefix - optional' section allows for a single optional prefix to limit notifications to objects with keys starting with matching characters, with an example 'e.g. images/' and a link to 'special characters'.

aws Services Search [Alt+S]

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.

Q s3/anushkalambdabucket X ↺

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 X

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/

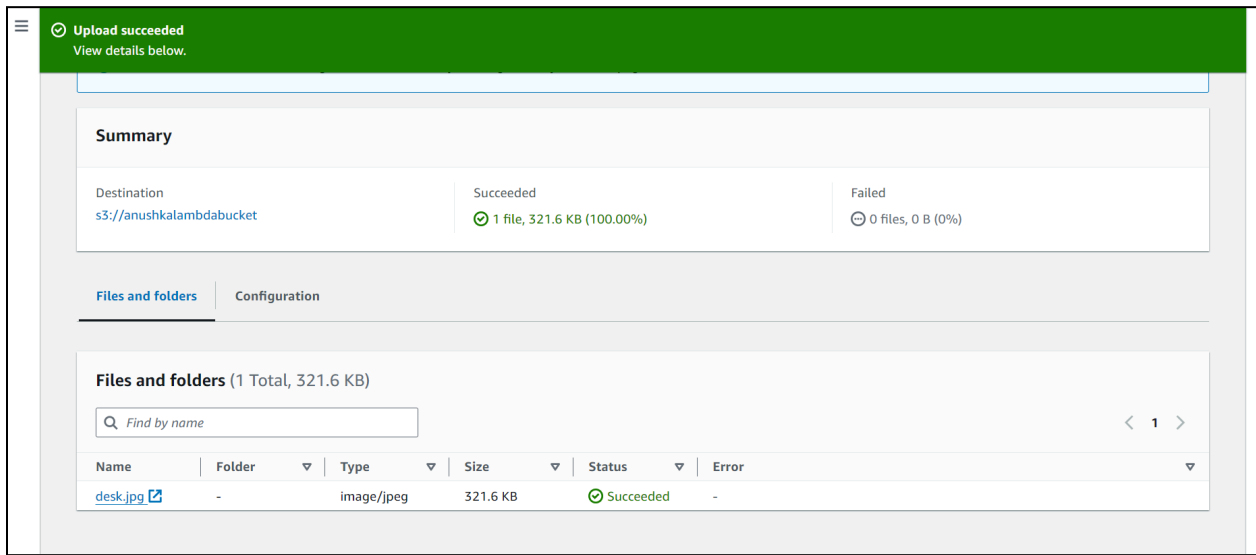
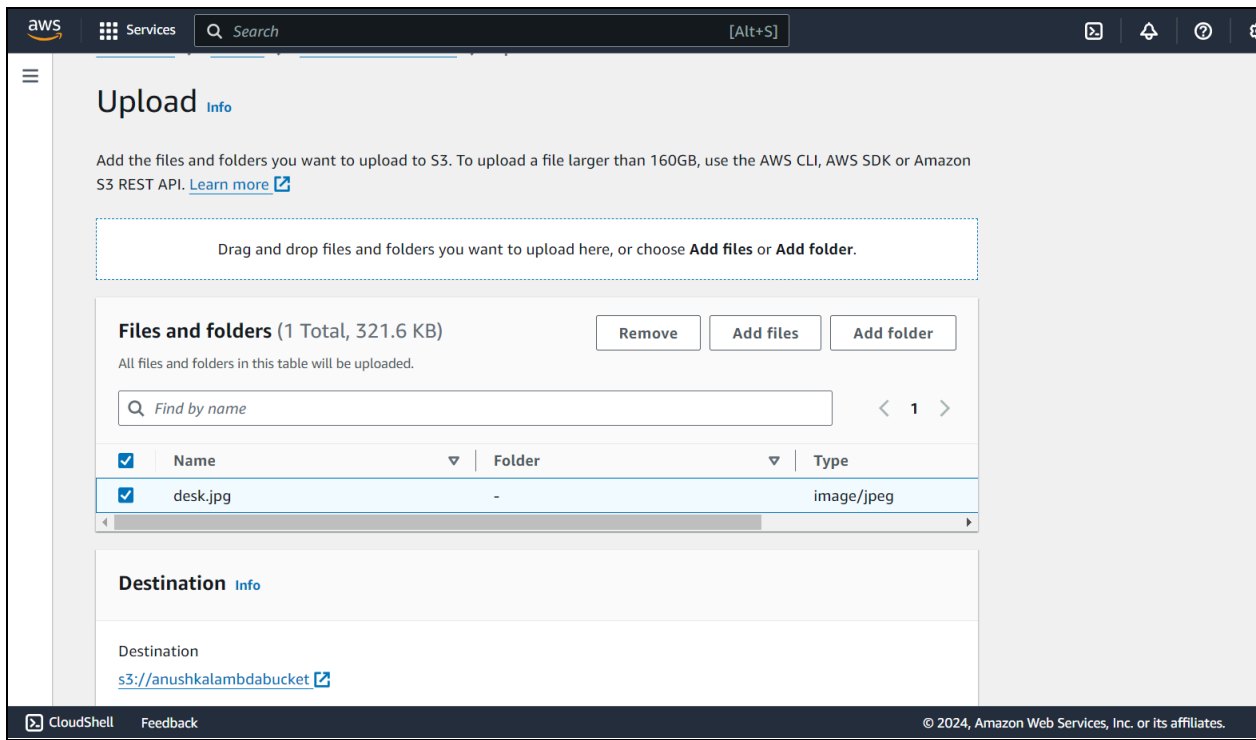
CloudShell Feedback © 2024, Amazon Web Services

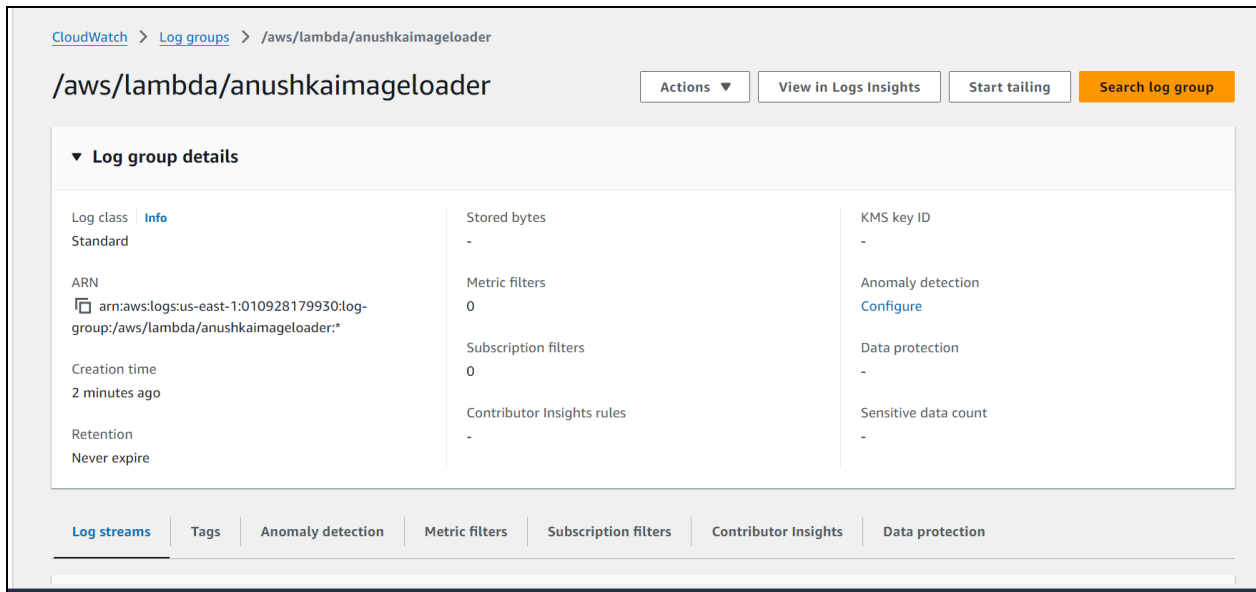
The screenshot shows the AWS IAM console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a user profile 'Anushka32'. Below the navigation bar, a list of managed policies is shown: 'Managed policy AWSLambdaBasicExecutionRole-9ecf207b-9f4b-4ac0-b11b-8f0a2b5eb705, statement 0' and 'Managed policy AWSLambdaBasicExecutionRole-9ecf207b-9f4b-4ac0-b11b-8f0a2b5eb705, statement 1'. The main content area is titled 'Resource-based policy statements (1) Info'. It includes a search bar with the text 'Find policy statements' and a table with the following columns: 'Statement ID', 'Principal', 'PrincipalOrgID', 'Conditions', and 'Action'. The table contains one entry with 'Statement ID' 'lambda-de656519-7...', 'Principal' 's3.amazonaws.com', 'PrincipalOrgID' '-', 'Conditions' 'StringEquals, ArnLike', and 'Action' 'lambda:InvokeFunction'. Below the table, there's a section titled 'Auditing and compliance' with a paragraph stating: 'AWS CloudTrail can log this function's invocations for operational and risk auditing, governance, and compliance. [Get started](#) on the CloudTrail console.'

Statement ID	Principal	PrincipalOrgID	Conditions	Action
lambda-de656519-7...	s3.amazonaws.com	-	StringEquals, ArnLike	lambda:InvokeFunction

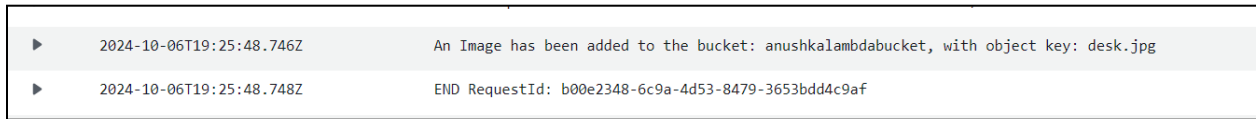
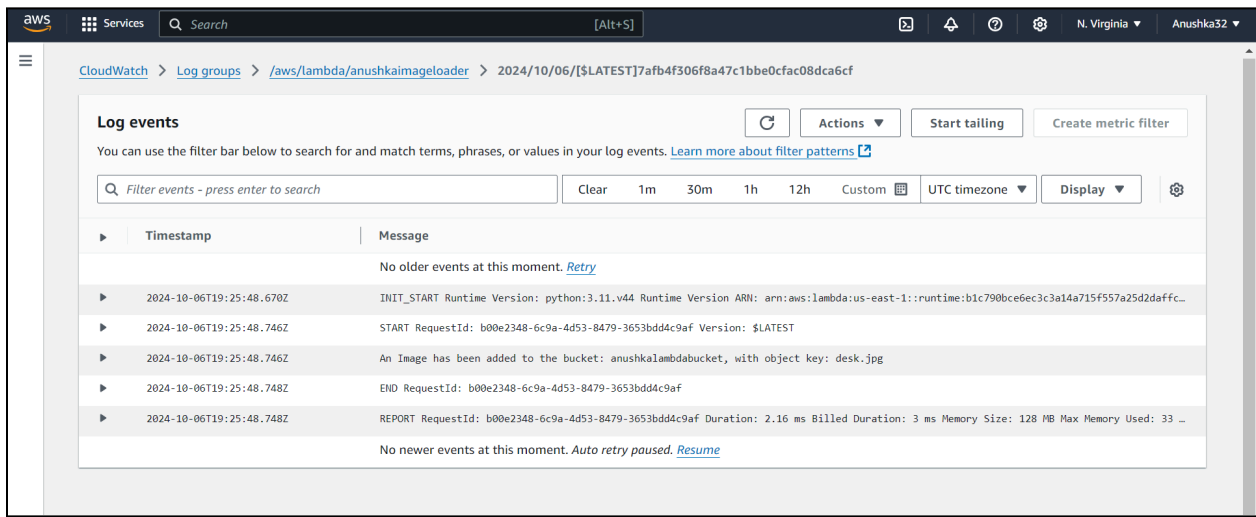
The screenshot shows the AWS S3 console interface. The left sidebar has a search bar and a list of services, including 'Amazon S3'. The main content area is titled 'anushkalambdabucket Info'. It includes a search bar with the text 'Find objects by prefix' and a table with the following columns: 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. The table is empty, and a message states: 'No objects. You don't have any objects in this bucket.' Below the message is an 'Upload' button. The top navigation bar shows the user profile 'Anushka32' and the region 'N. Virginia'.

Name	Type	Last modified	Size	Storage class
------	------	---------------	------	---------------





5. Test the Setup:
Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.



Conclusion:

Integrating AWS Lambda with S3 allows for real-time, automated processing of events such as file uploads. In this example, a Lambda function is configured to log a message whenever an image is added to a specific S3 bucket. This setup demonstrates the power and flexibility of serverless computing by automating tasks without requiring manual intervention or server management. By leveraging AWS Lambda, developers can efficiently handle event-driven workflows, reduce operational overhead, and quickly deploy scalable solutions that respond to specific actions within cloud environments.