Shravani Anil Patil **D15A-38**

Advanced DevOps Lab
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

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.

AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. It automatically scales your application by running code in response to events. One of the most common use cases for Lambda is integrating it with Amazon S3, a scalable object storage service.

When an object is added to or modified in an S3 bucket, you can configure an S3 event notification to trigger a Lambda function. This event-driven architecture simplifies processing data in real time and reduces the need for manual intervention.

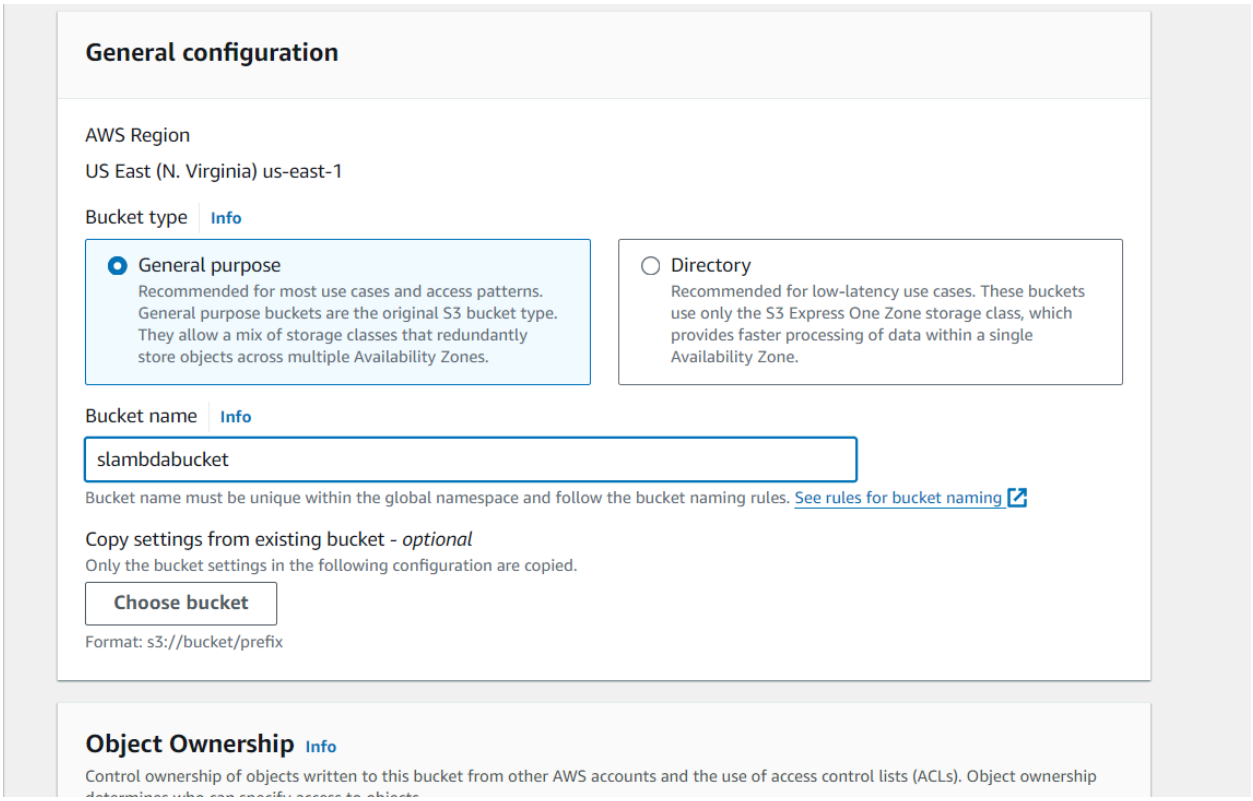
Key Benefits

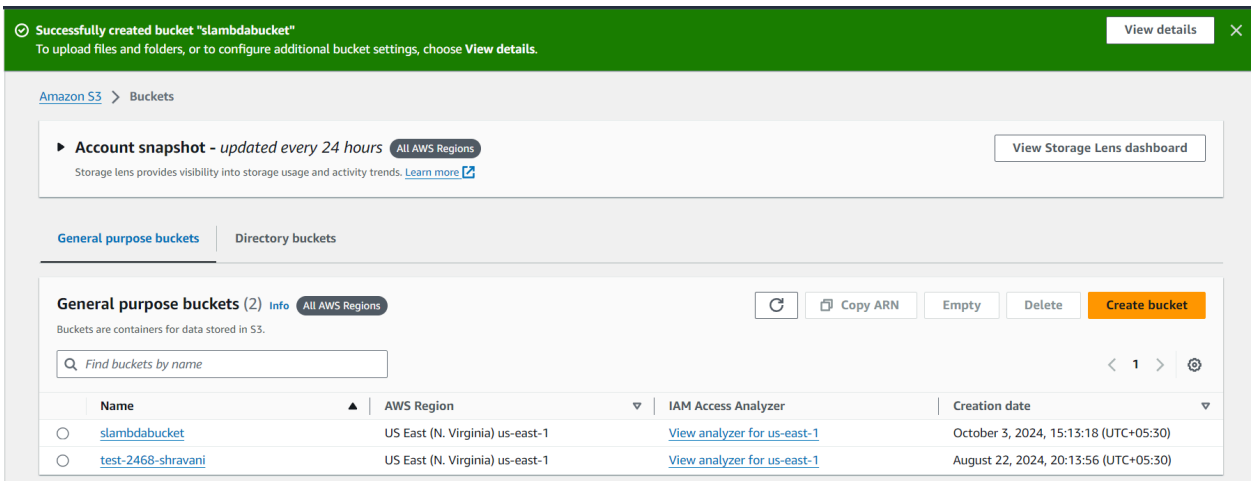
1. **Serverless Architecture**:
    - No need to manage servers or infrastructure. You focus solely on writing code.
    - AWS handles scaling automatically based on the number of events.
2. **Cost Efficiency**:
    - Pay only for the compute time you consume. You are charged based on the number of requests and the duration of execution.
3. **Real-time Processing**:
    - Automatically process files as they are uploaded, allowing for instant reactions to events (e.g., generating thumbnails, analyzing data).
4. **Ease of Integration**:
    - Lambda integrates easily with other AWS services, enabling seamless workflows and data processing pipelines.
5. **Event-Driven Processing**:
    - Respond to changes in data, such as new uploads, deletions, or updates, without the need for polling or scheduled tasks.

Workflow:

1. Create an S3 Bucket:

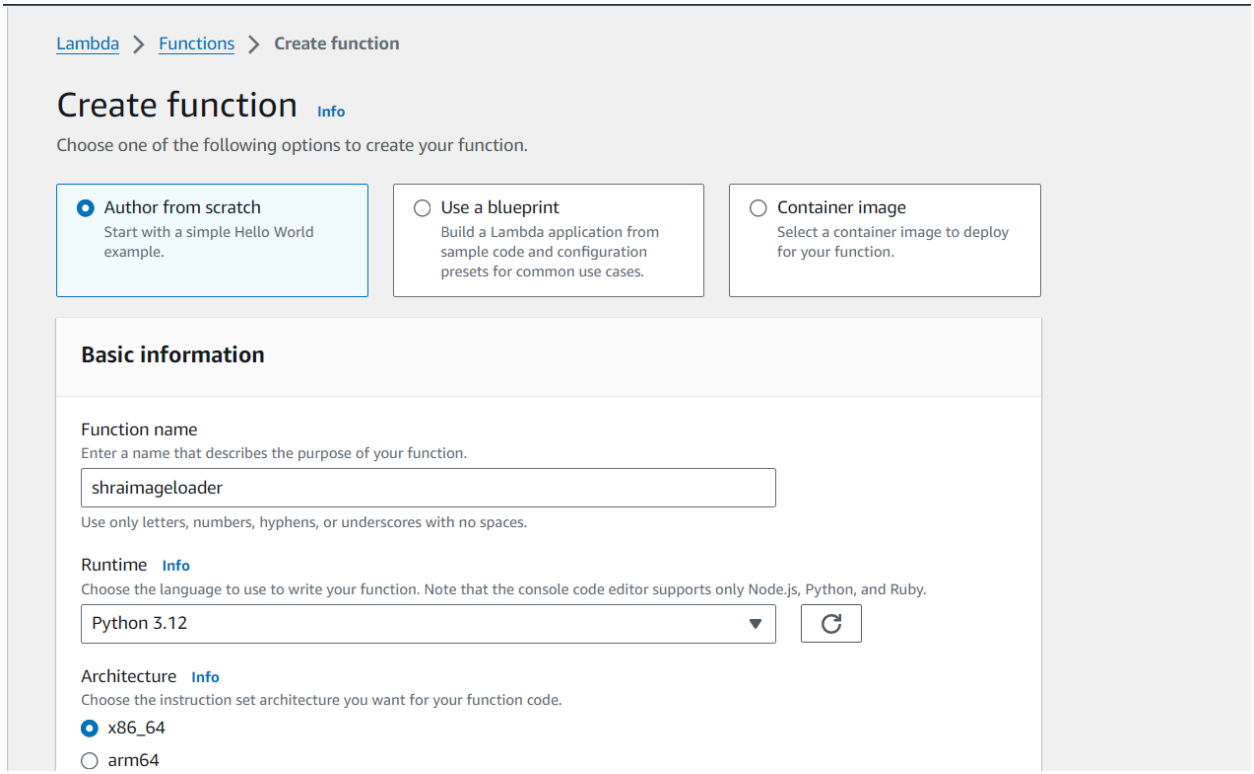
○ First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

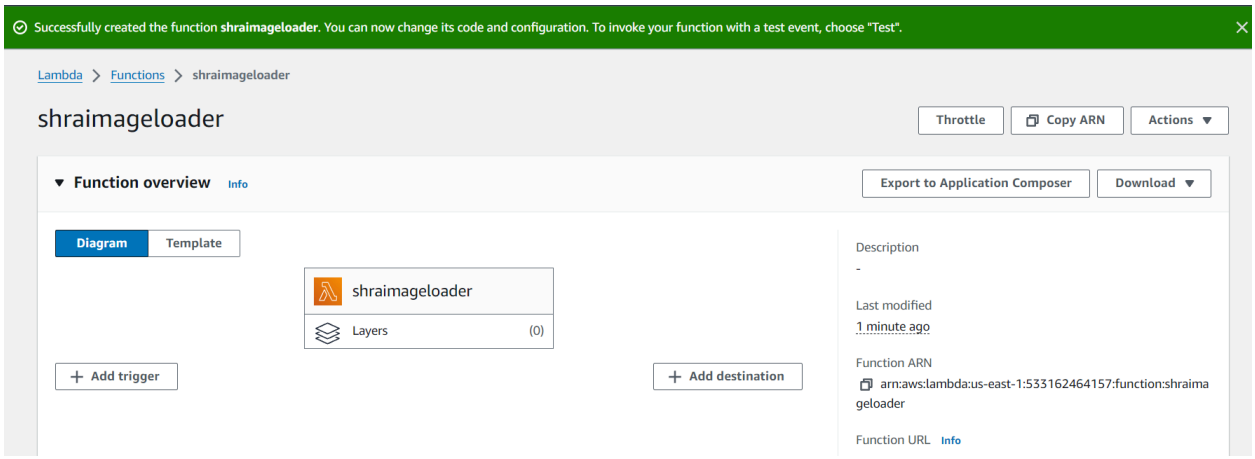




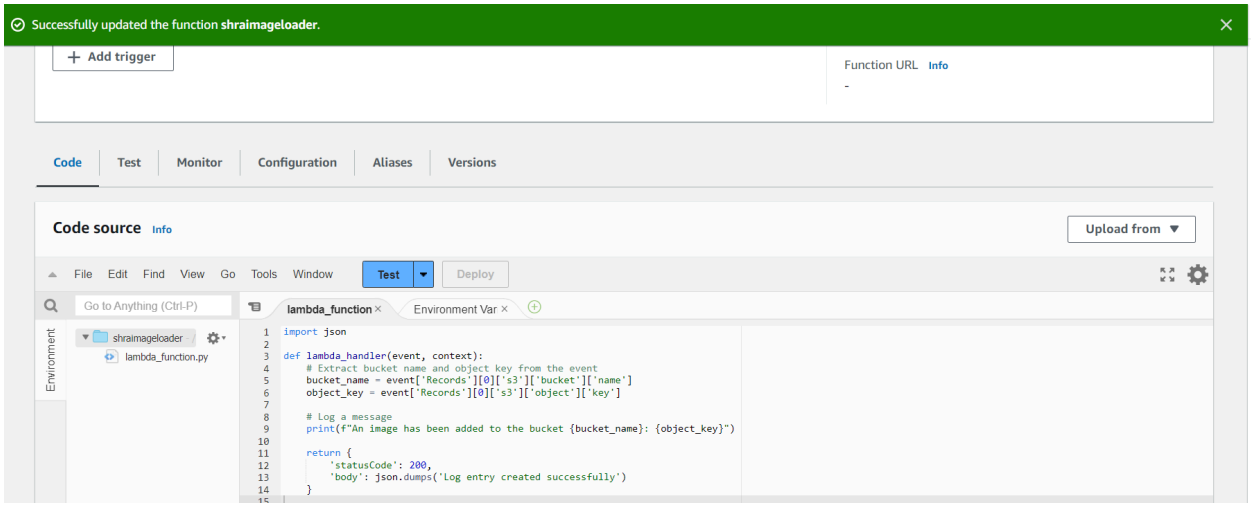
2. Create the Lambda Function:

○ 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.**

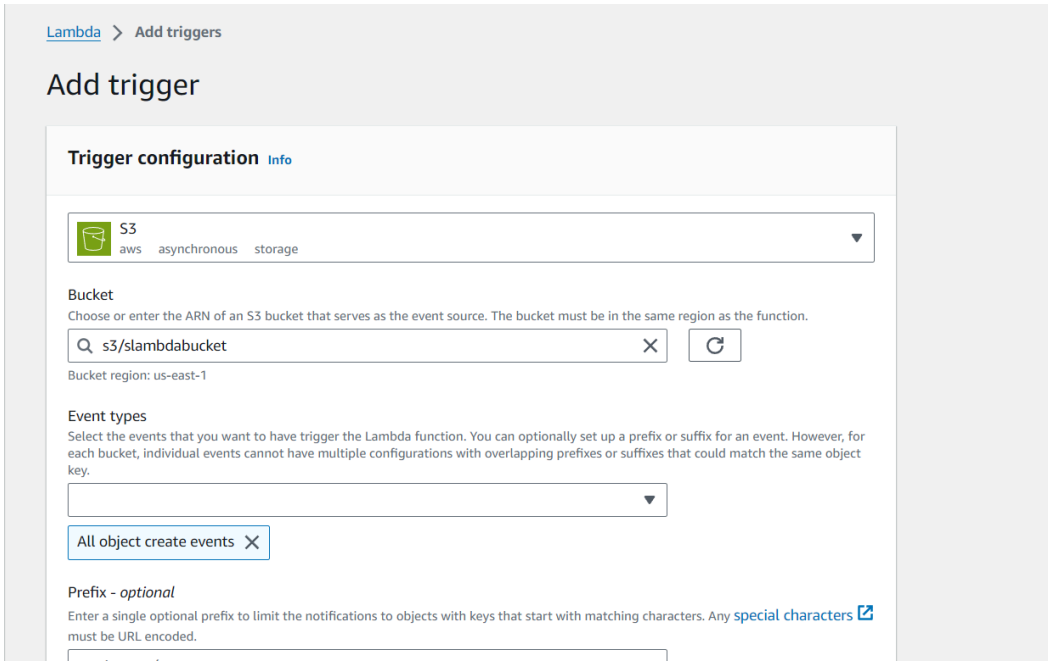


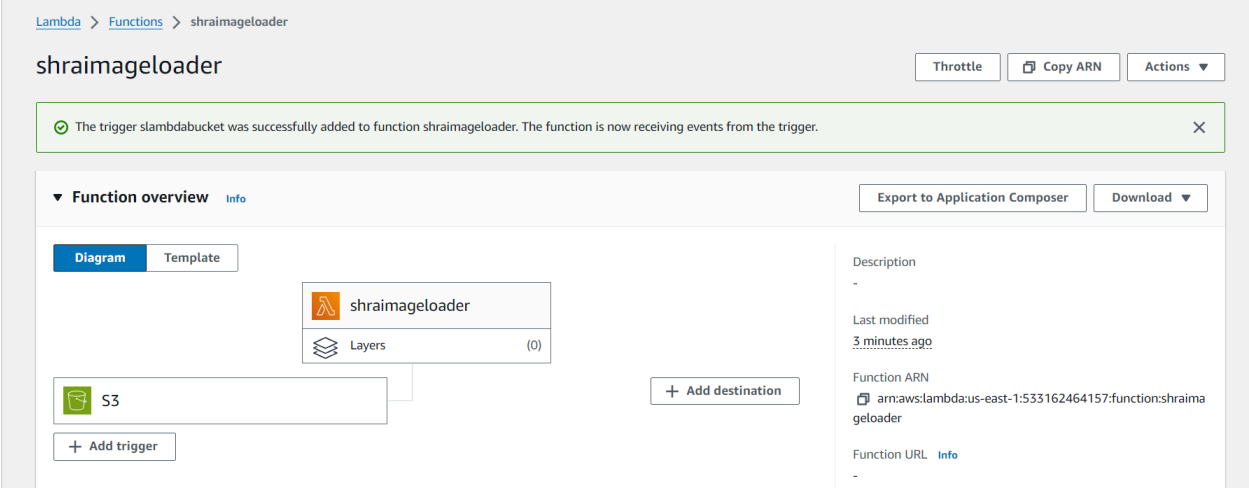
3. Set Up Permissions:

○ 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:

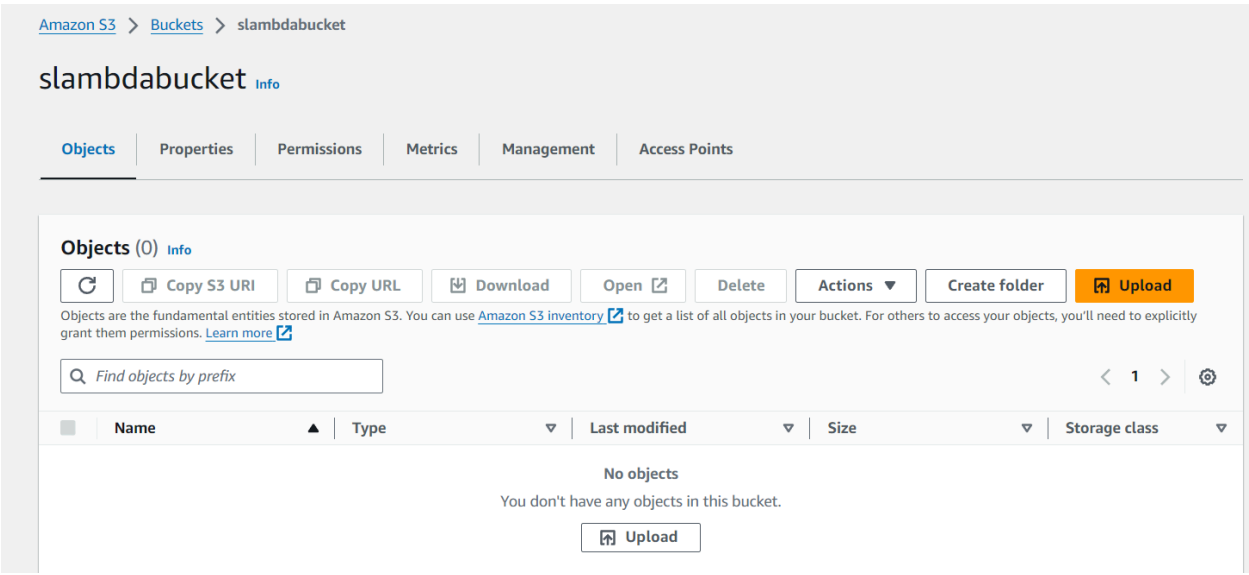
○ Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).



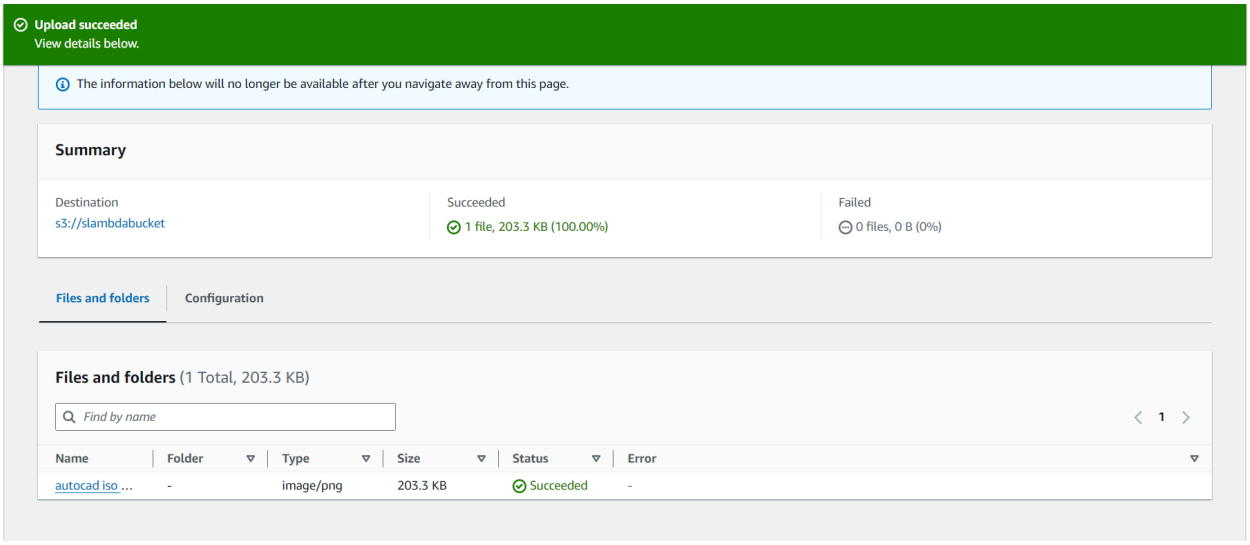


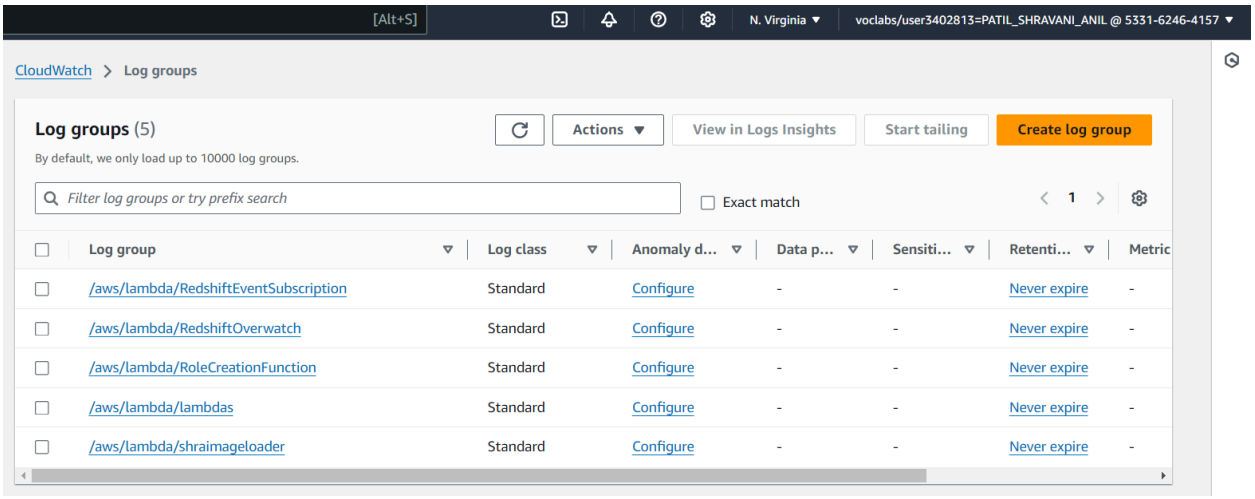
5. Test the Setup:

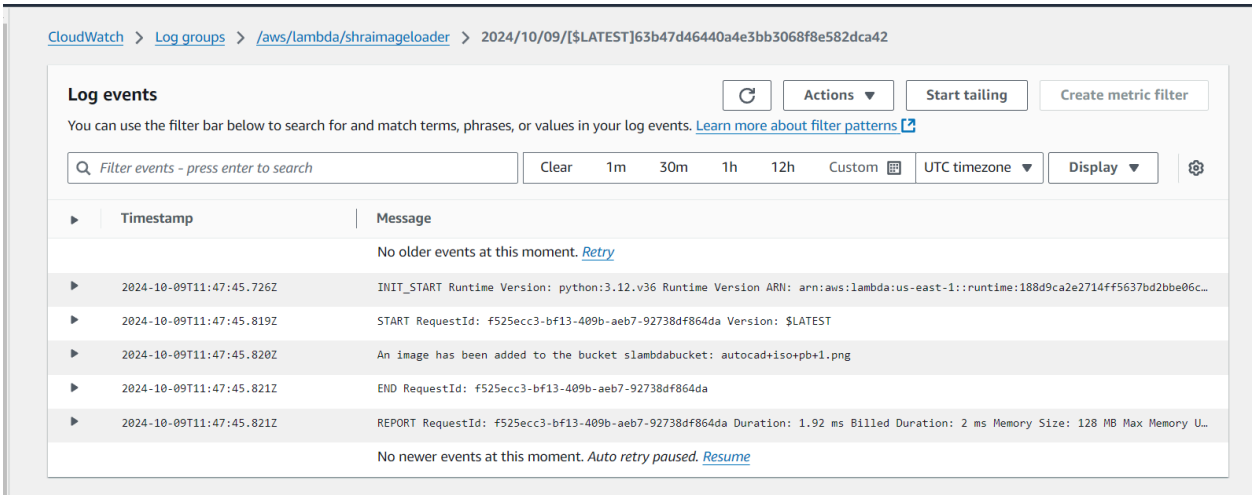
○ Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message "An Image has been added" in AWS CloudWatch Logs.



Outcomes:


Upload succeeded
View details below.
The information below will no longer be available after you navigate away from this page.
Summary
Destination
s3://slambdabucket
Succeeded
1 file, 203.3 KB (100.00%)
Failed
0 files, 0 B (0%)
Files and folders
Configuration
Files and folders (1 Total, 203.3 KB)
Find by name
Name
Folder
Type
Size
Status
Error
autocad iso ...
-
image/png
203.3 KB
Succeeded
-


[Alt+S]
N. Virginia
voclabs/user3402813=PATIL_SHRAVANI_ANIL @ 5331-6246-4157
CloudWatch
Log groups
Log groups (5)
By default, we only load up to 10000 log groups.
Actions
View in Logs Insights
Start tailing
Create log group
Filter log groups or try prefix search
Exact match
Log group
Log class
Anomaly d...
Data p...
Sensiti...
Retenti...
Metric
/aws/lambda/RedshiftEventSubscription
Standard
Configure
Never expire
/aws/lambda/RedshiftOverwatch
Standard
Configure
Never expire
/aws/lambda/RoleCreationFunction
Standard
Configure
Never expire
/aws/lambda/lambdas
Standard
Configure
Never expire
/aws/lambda/shraimageloader
Standard
Configure
Never expire


CloudWatch
Log groups
/aws/lambda/shraimageloader
2024/10/09/[$LATEST]63b47d46440a4e3bb3068f8e582dca42
Log events
Actions
Start tailing
Create metric filter
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
Filter events - press enter to search
Clear
1m
30m
1h
12h
Custom
UTC timezone
Display
Timestamp
Message
No older events at this moment. Retry
2024-10-09T11:47:45.726Z
INIT_START Runtime Version: python:3.12.v36 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:188d9ca2e2714ff5637bd2bbe06c...
2024-10-09T11:47:45.819Z
START RequestId: f525ecc3-bf13-409b-aeb7-92738df864da Version: $LATEST
2024-10-09T11:47:45.820Z
An image has been added to the bucket slambdabucket: autocad+iso+pb+1.png
2024-10-09T11:47:45.821Z
END RequestId: f525ecc3-bf13-409b-aeb7-92738df864da
2024-10-09T11:47:45.821Z
REPORT RequestId: f525ecc3-bf13-409b-aeb7-92738df864da Duration: 1.92 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory U...
No newer events at this moment. Auto retry paused. Resume

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.