Cloud and Serverless Computing Project

Venkat Anand
2100030728

Cloud-Based Assignment Evaluator: Achieving Precision through Similarity Index Analysis

In this project we will be using the services like AWS Lambda, Amazon S3, AWS Textract, AWS IAM, AWS Cloud Watch.

AWS Lambda:

It is a serverless service provided by AWS.
There is no need to provision the resources and servers.
It is an event-driven architecture that is when the only when the event is triggered the function gets executed.
It uses pay-as-you-go pricing .
It has automatic scaling.

Amazon S3:

It is an object storage service.
It is a storage service where the data can be stored and retrieved anytime and anywhere.
It has got the best durability 99.999999999 .
It has unlimited storage and where each object should comprise the size of 0 bytes to 5 Tb.

AWS Textract:

It is a machine learning tool that can automate the printed text and handwriting.
Extract text, forms, and tables from documents with structured data, using the Amazon Textract Document Analysis API.

AWS IAM:

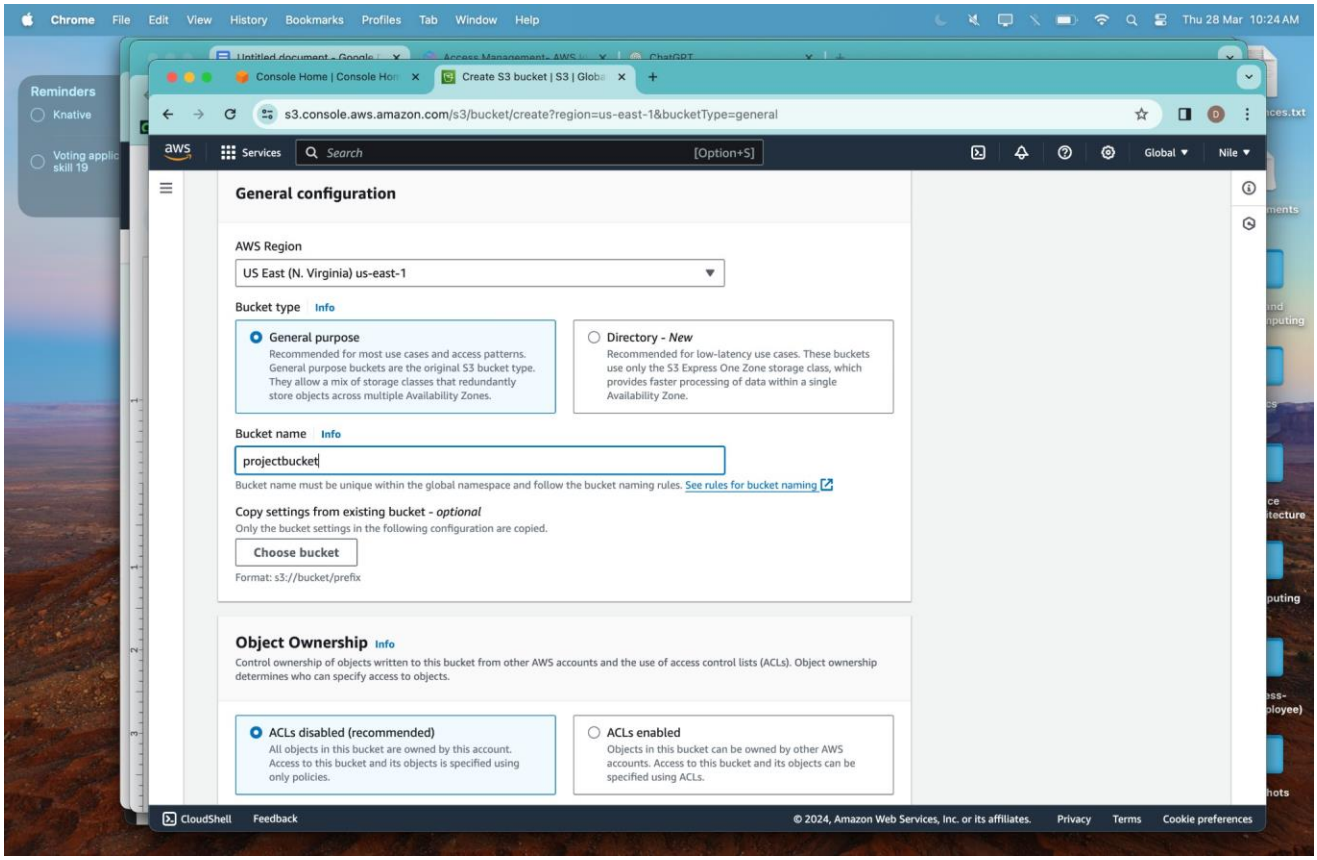
A web service that helps you securely control access to AWS resources.
In this project we will be using **Roles.**
Roles: They provide temporary permission for accessing the service in AWS.

AWS CloudWatch:

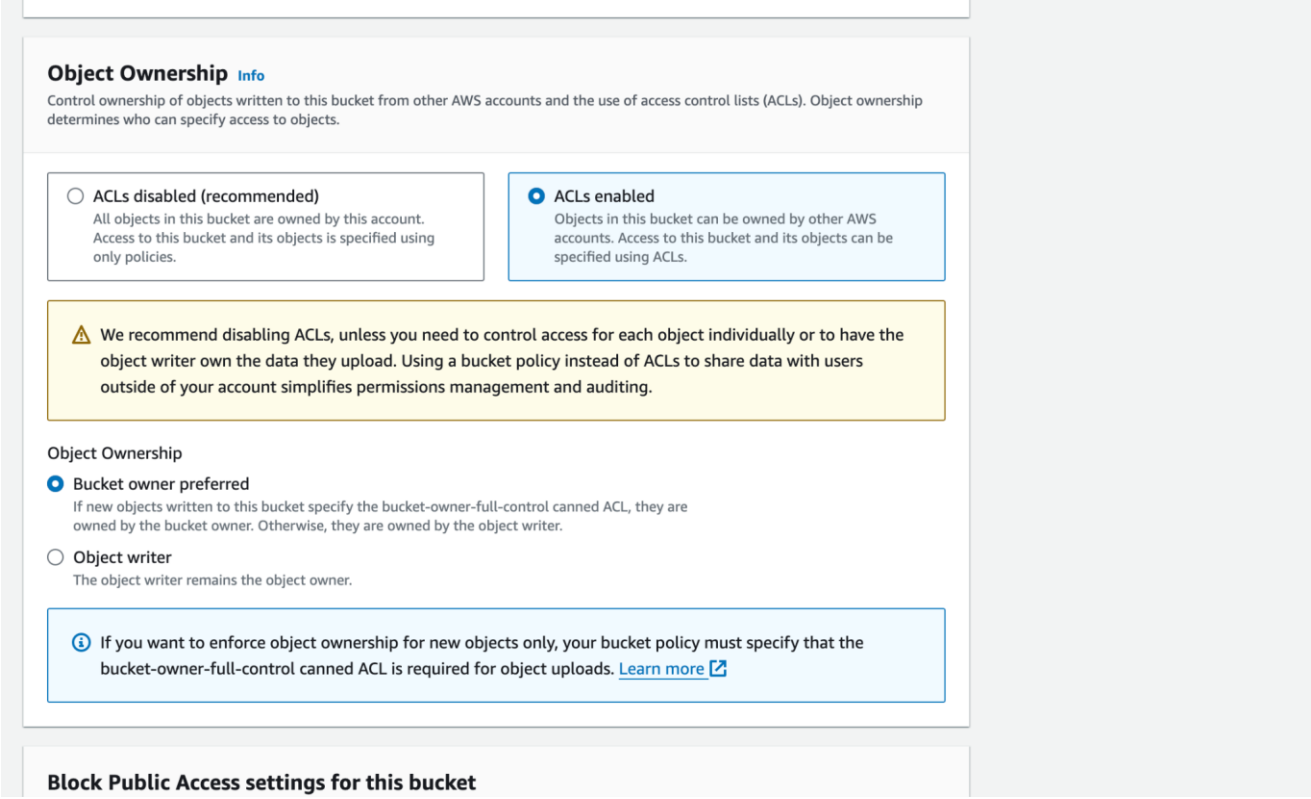
Observe and monitor resources and applications on AWS, on premises, and on other clouds.
It helps you to view the history of events that have been performed by the users.

Steps of Project :

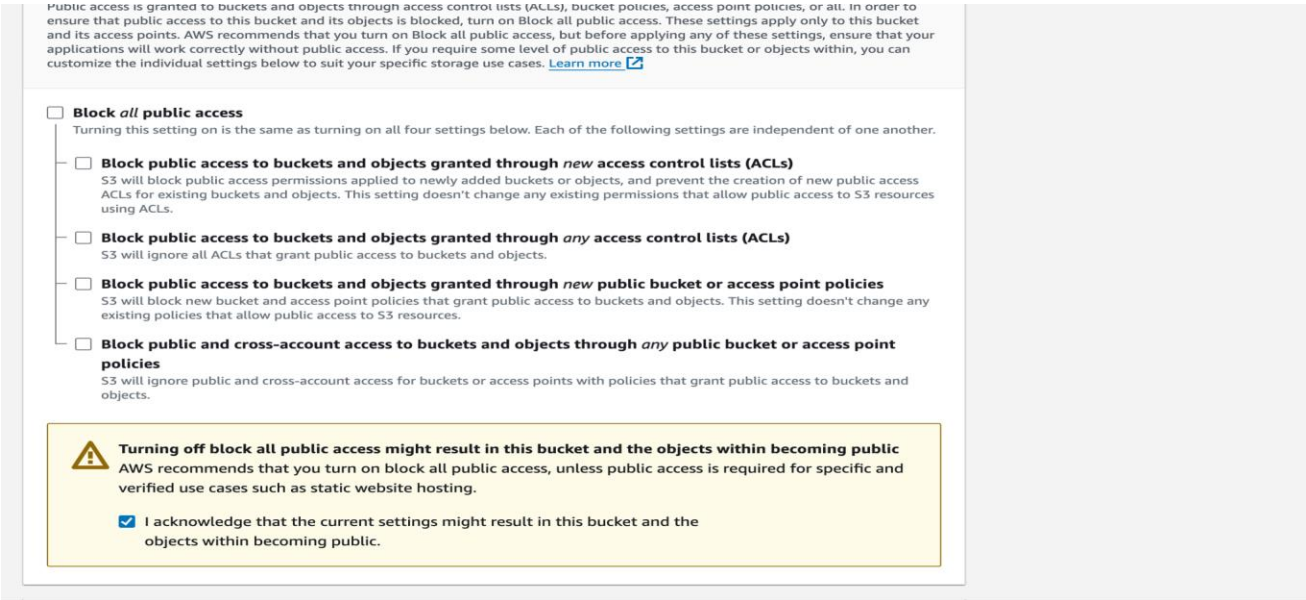
1. Create an S3 bucket.



2. Enable the ACL (Access Control Lists):



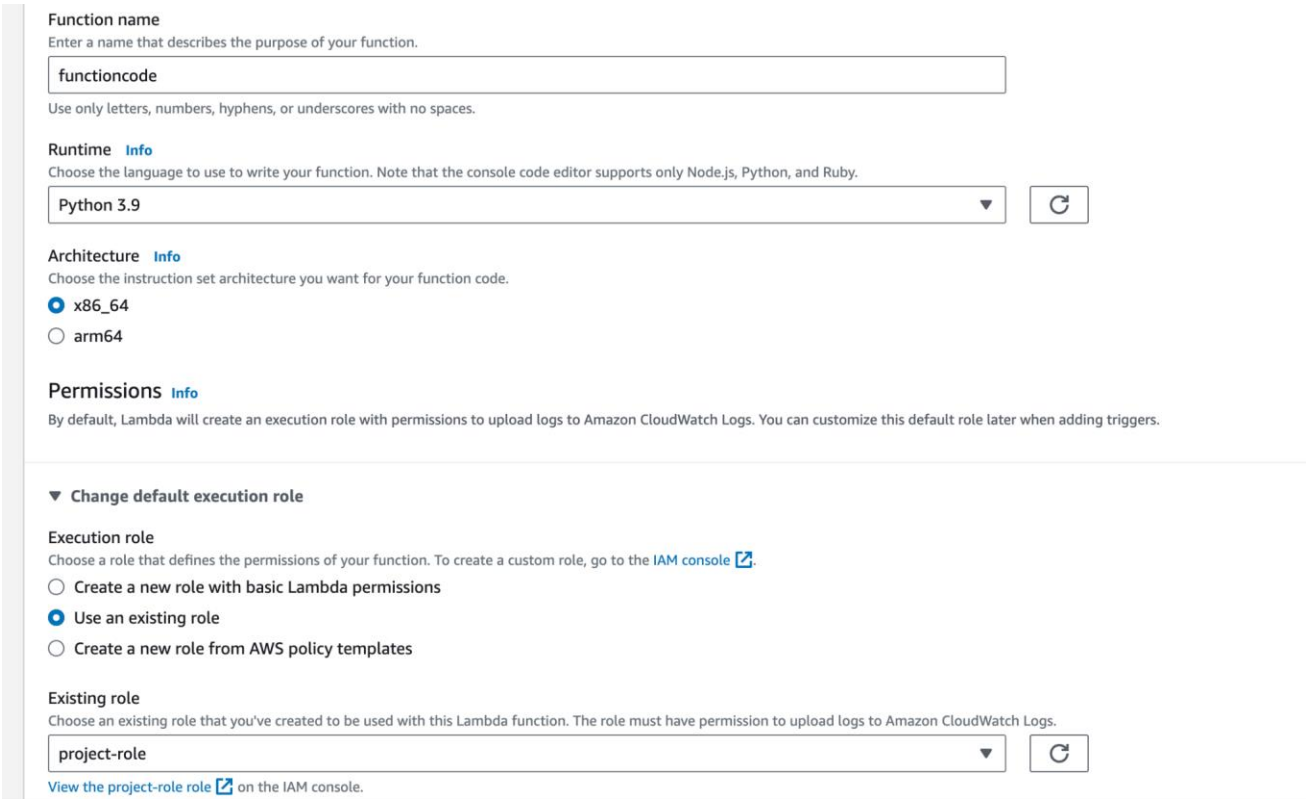
3. Enable public access.

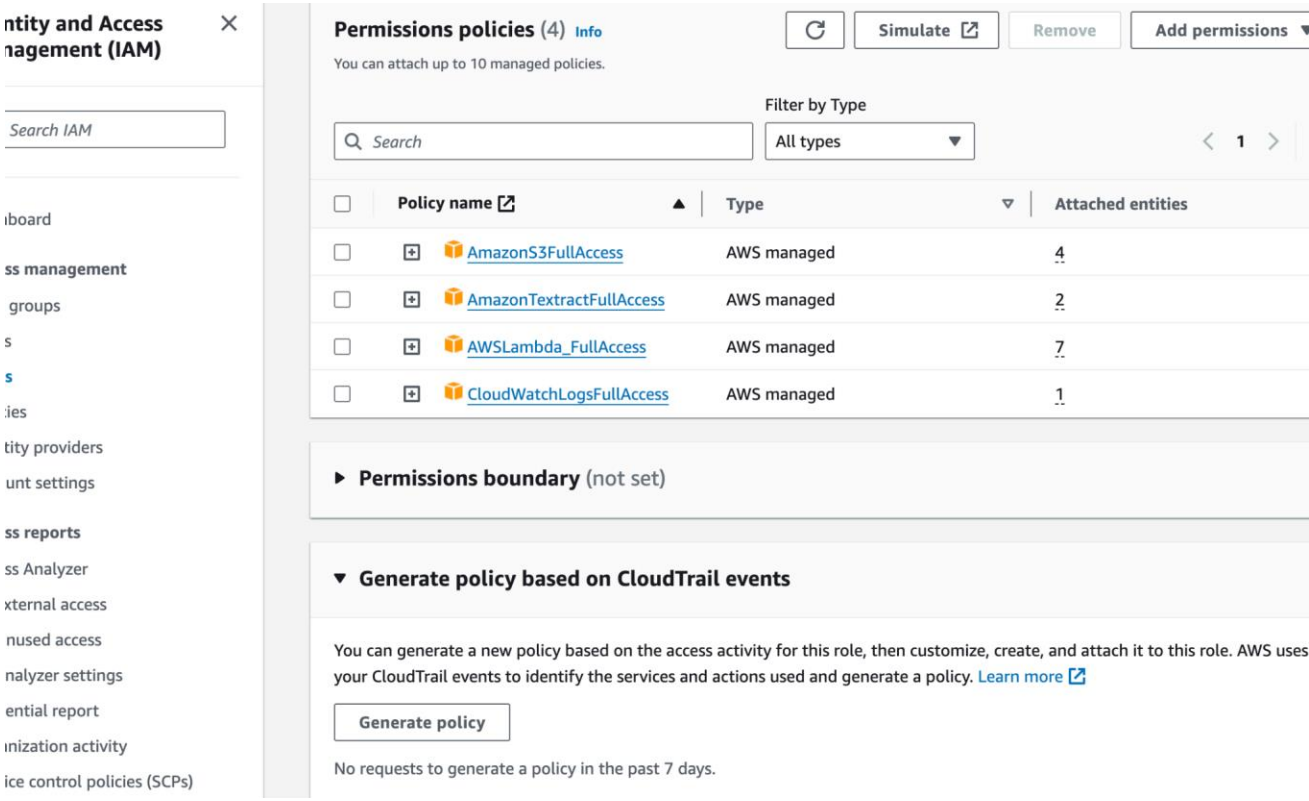


4. Create the bucket.

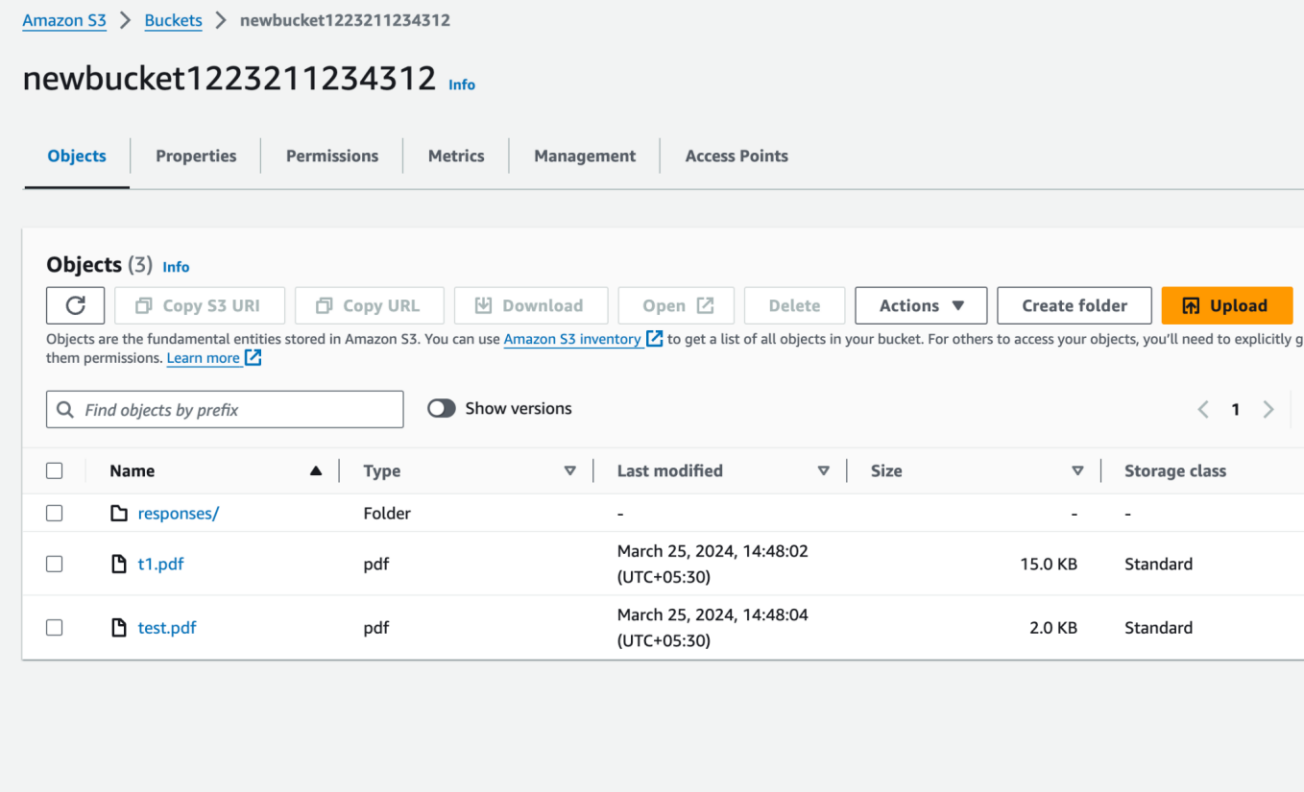
Create a Lambda Function:

1. Create a function: Provide a name for the function
2. Choose the RunTime as : Python 3.9
3. And create a role: which includes the permissions Amazon S3, AWS Textract, AWS CloudWatch, AWS Lambda and also give full access to all the services.

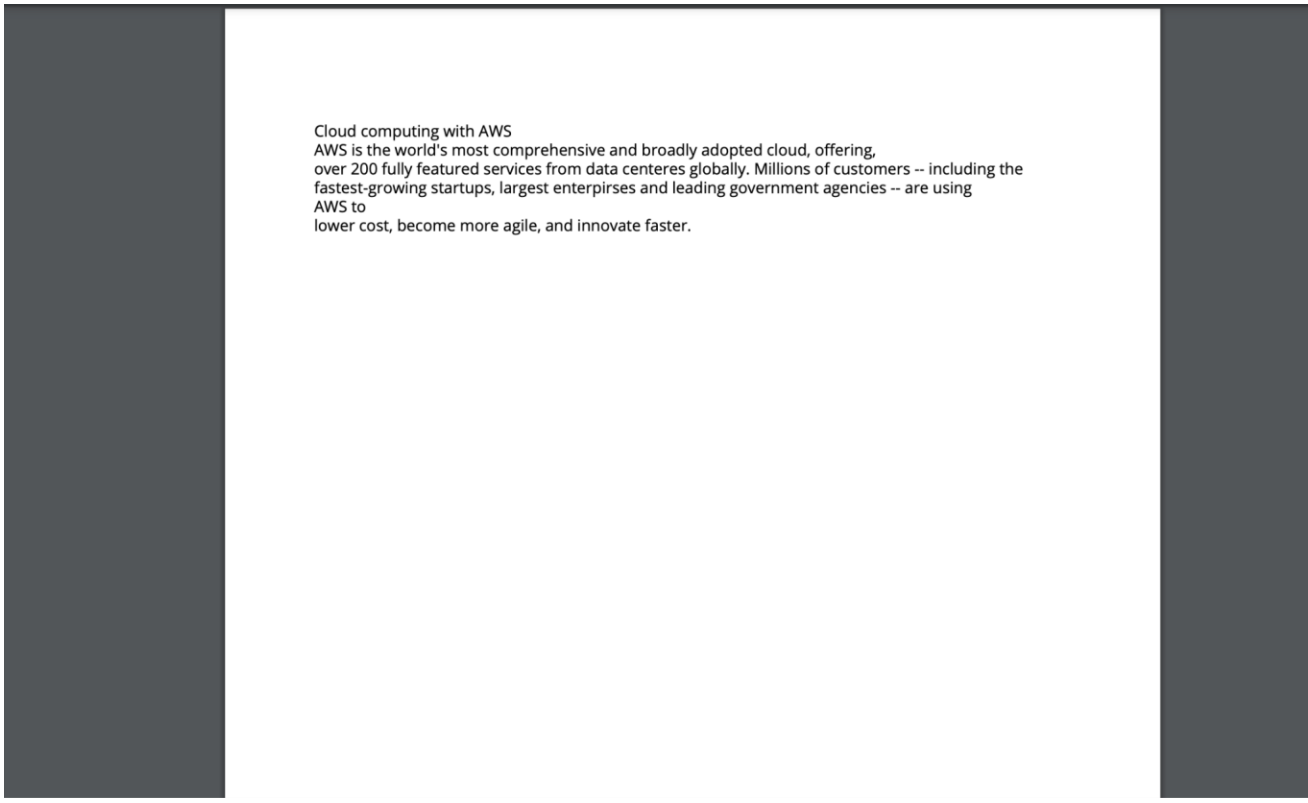




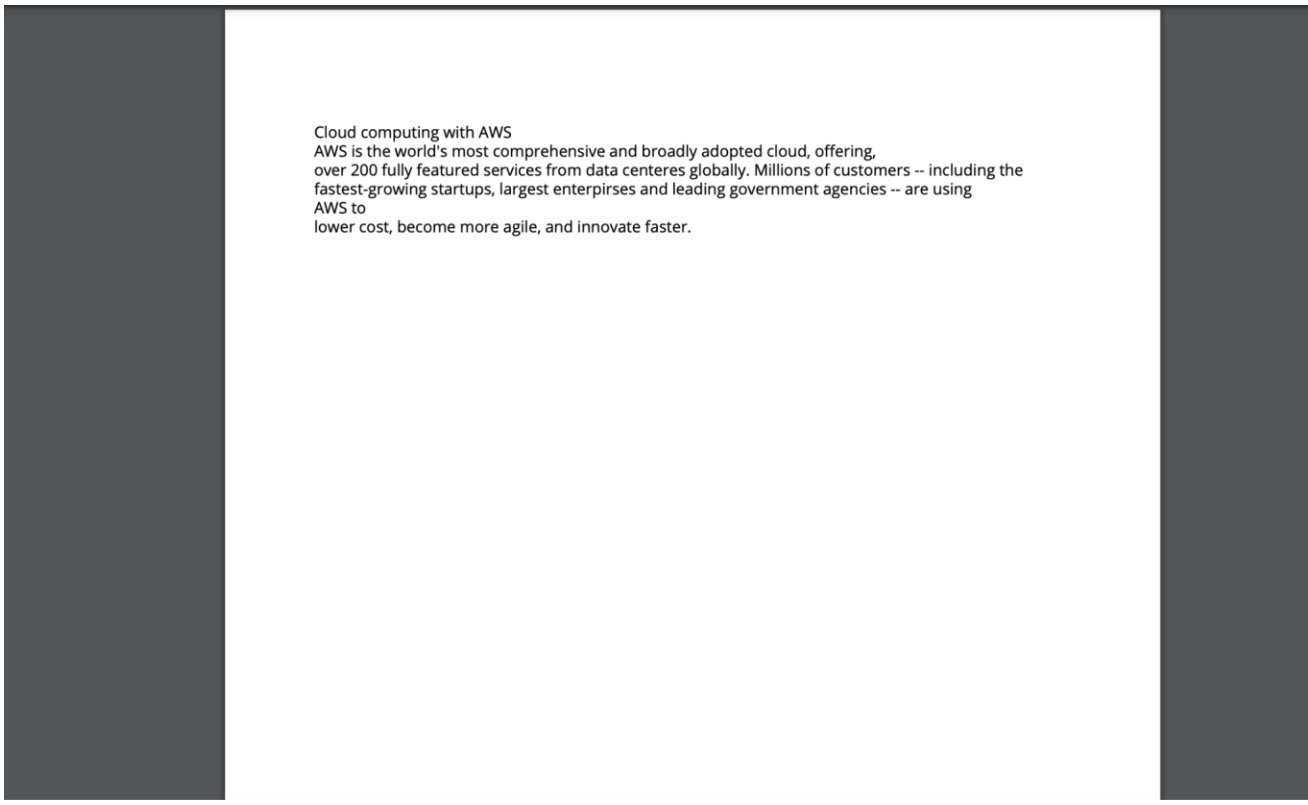
And now add two files one as the Main file and the other as sub file in the S3 to find the similarity index between the files.



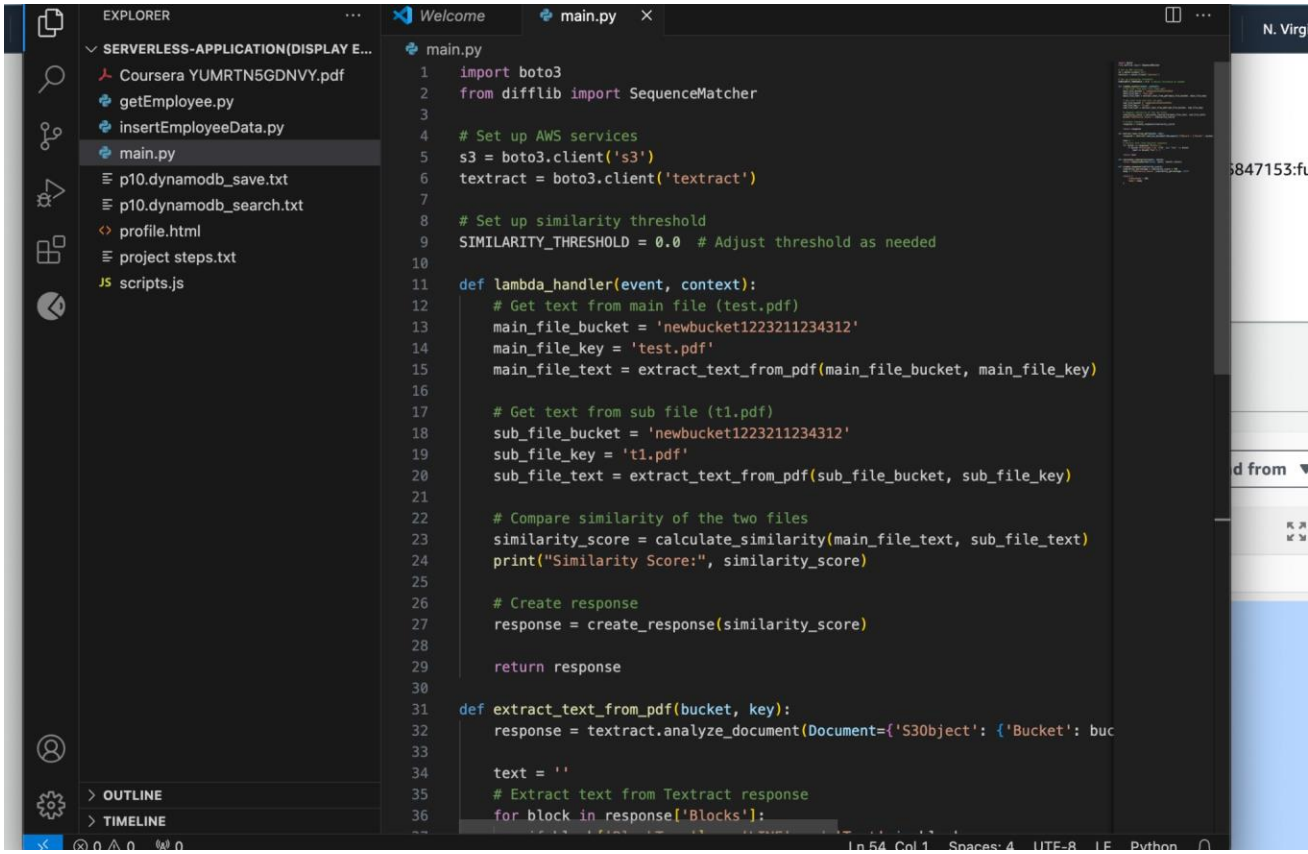
Test.pdf: This is the main file



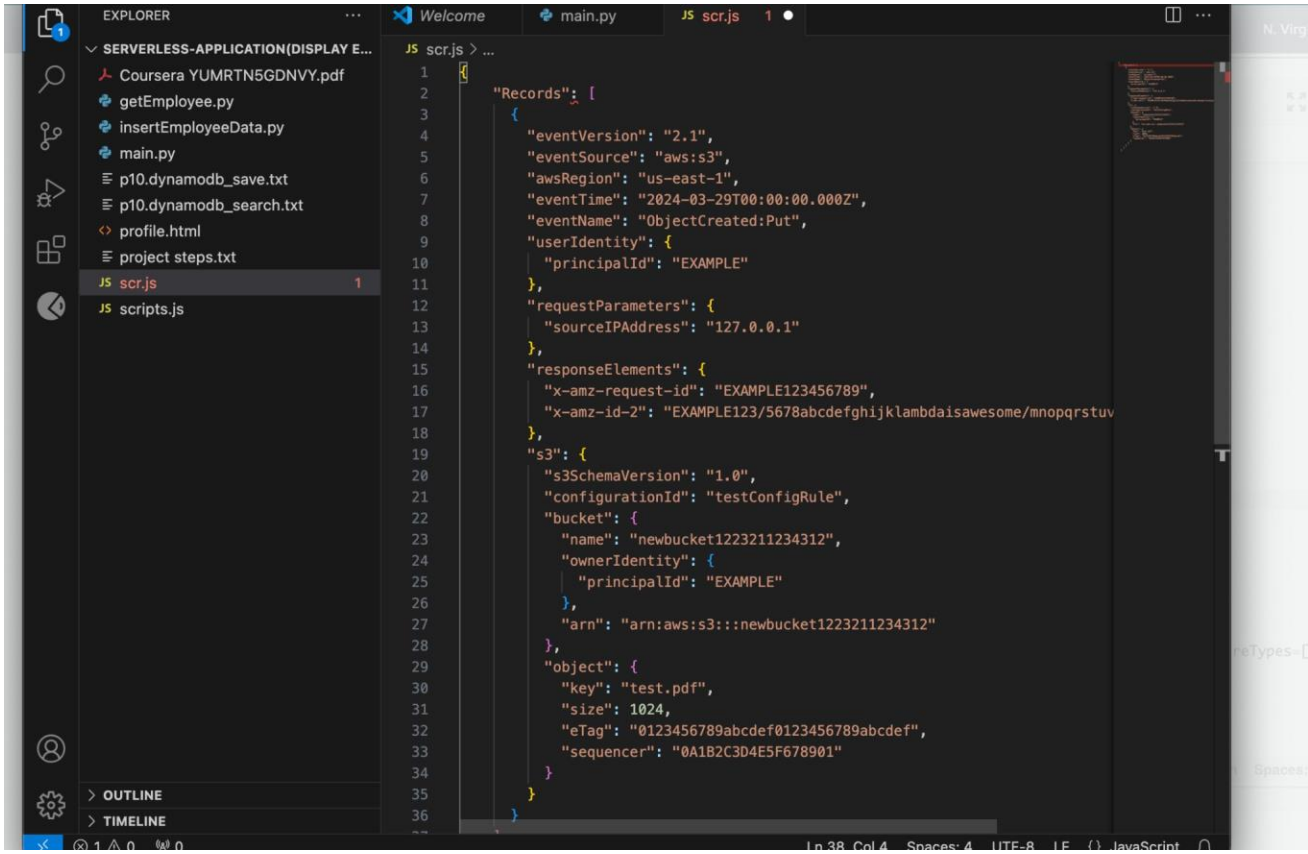
T1.pdf: This is the sub file.

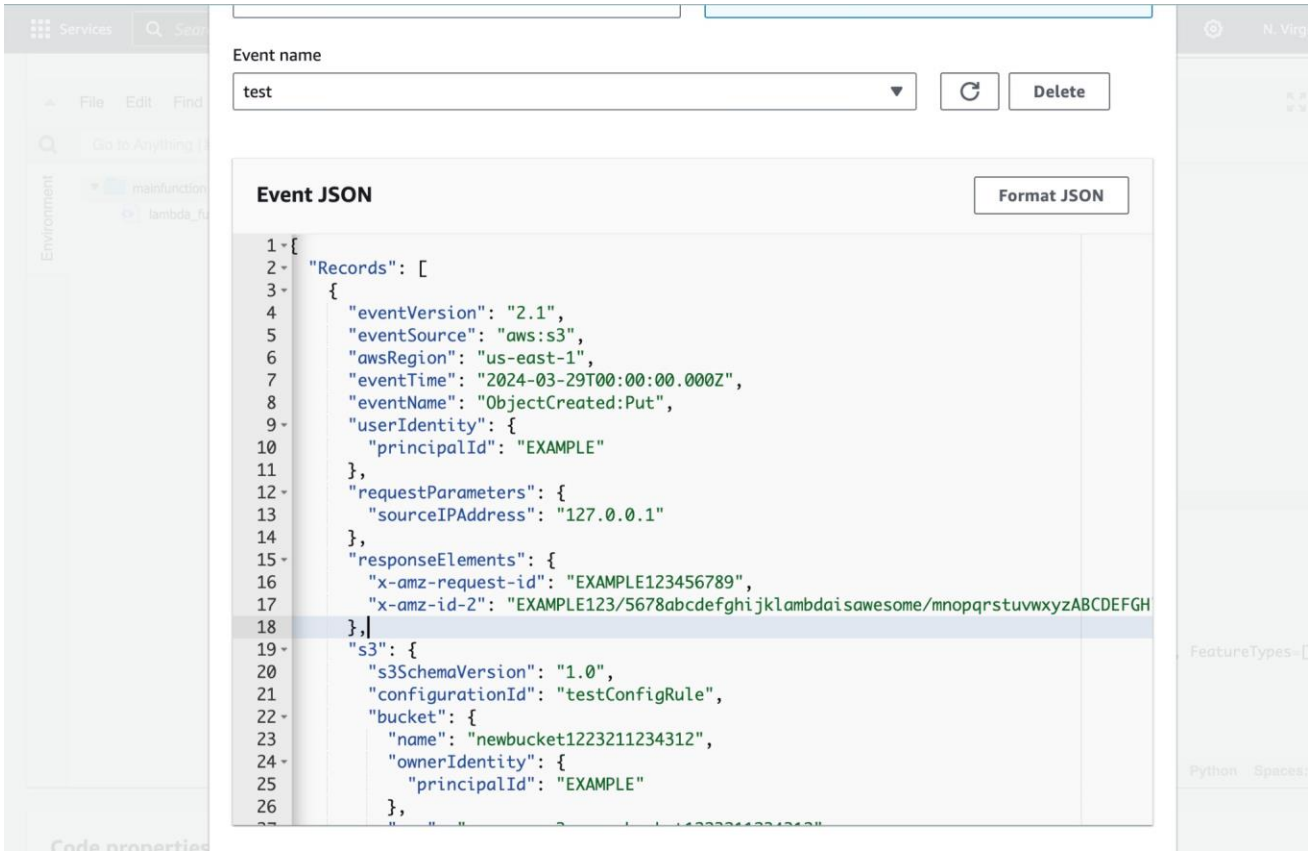


And now add the code in the lambda function to check the similarity index of both the files:

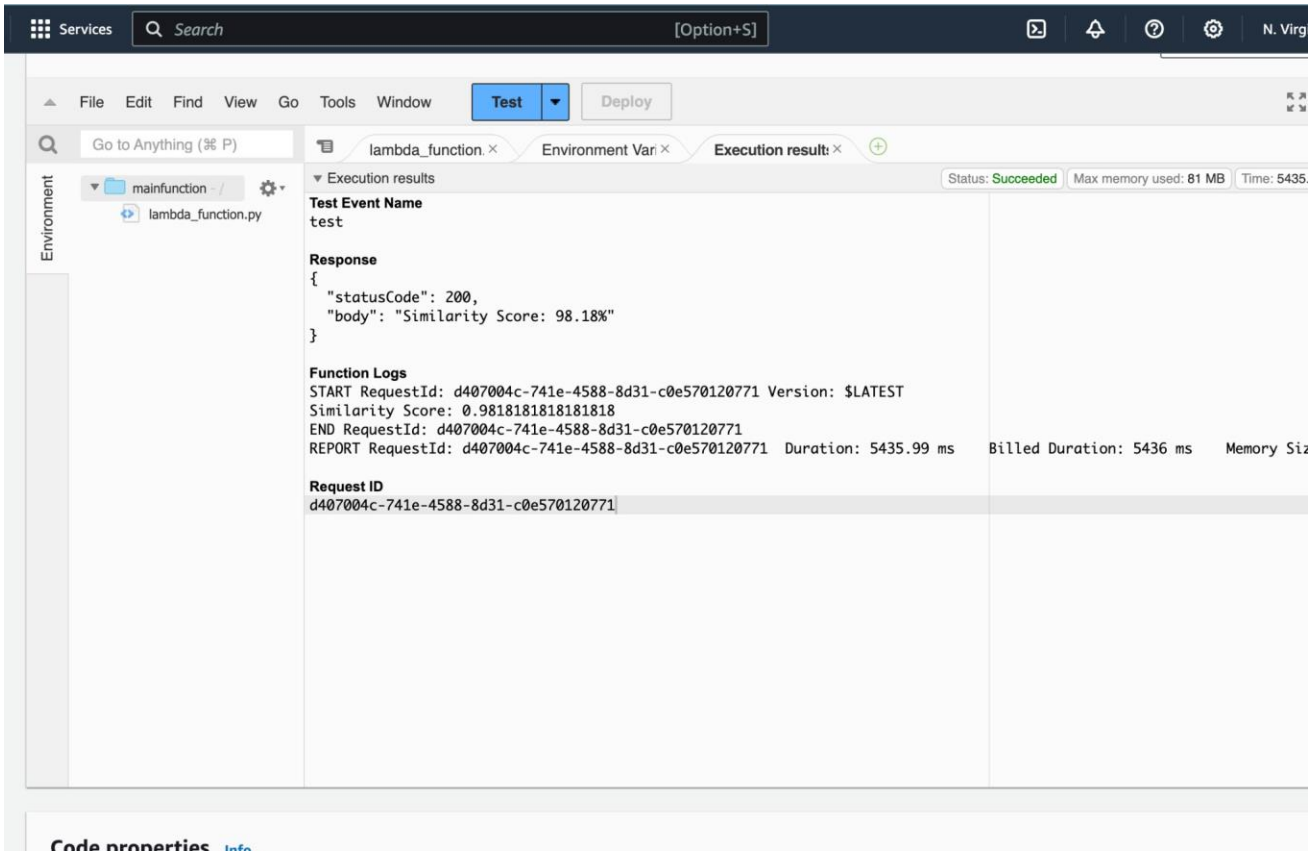


Create an Event to trigger the function:

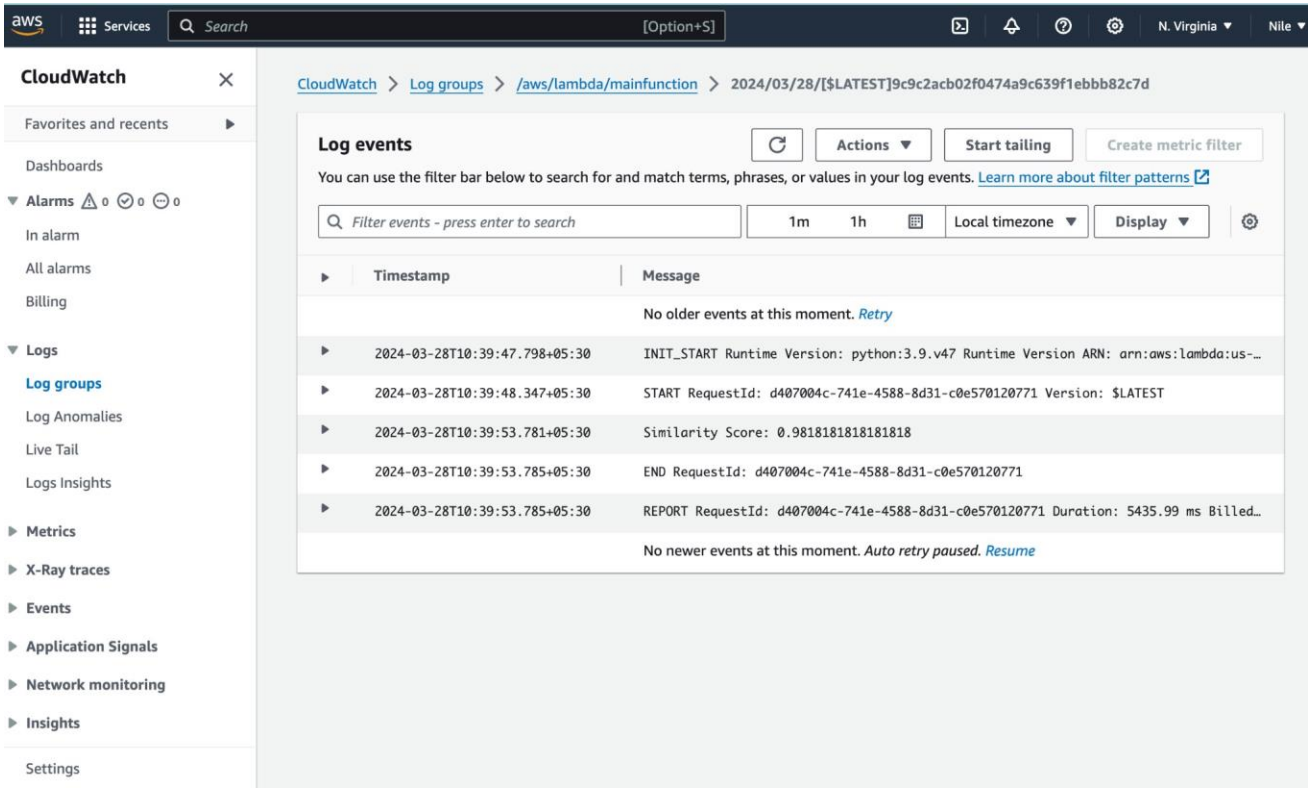

EXPLORER
SERVERLESS-APPLICATION(DISPLAY E...
Coursera YUMRTN5GDNVY.pdf
getEmployee.py
insertEmployeeData.py
main.py
p10.dynamodb_save.txt
p10.dynamodb_search.txt
profile.html
project steps.txt
scr.js
scripts.js
OUTLINE
TIMELINE
Welcome
main.py
scr.js
scr.js > ...
{
"Records": [
{
"eventVersion": "2.1",
"eventSource": "aws:s3",
"awsRegion": "us-east-1",
"eventTime": "2024-03-29T00:00:00.000Z",
"eventName": "ObjectCreated:Put",
"userIdentity": {
"principalId": "EXAMPLE"
},
"requestParameters": {
"sourceIPAddress": "127.0.0.1"
},
"responseElements": {
"x-amz-request-id": "EXAMPLE123456789",
"x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuv
},
"s3": {
"s3SchemaVersion": "1.0",
"configurationId": "testConfigRule",
"bucket": {
"name": "newbucket1223211234312",
"ownerIdentity": {
"principalId": "EXAMPLE"
},
"arn": "arn:aws:s3:::newbucket1223211234312"
},
"object": {
"key": "test.pdf",
"size": 1024,
"eTag": "0123456789abcdef0123456789abcdef",
"sequencer": "0A1B2C3D4E5F678901"
}
}
}
Ln 38, Col 4
Spaces: 4
UTF-8
LF
JavaScript



And now test the function:



The status code is 200 and the function worked !!
The similarity index between both the files is 98.18%.



And check the results in the Cloud Watch log: 98.18%.

GitHub Profile : https://github.com/Nilesh-27/Home-Assignment-Similarity-Index-AWS