VIT Vidyalankar Institute of Technology Accredited A+ by NAAC	*DEPARTMENT OF INFORMATION TECHNOLOGY*

Semester	T.E. Semester V – Information Technology
Subject	Advance DevOps Lab
Subject Professor In-charge	Prof. Indu Anoop
Laboratory	(Leave blank for now)

Student Name	Khot Mohammed Siddique	
Roll Number	20101A0048	
Grade and Subject Teacher's Signature		

Experiment	11-12	
Problem Statement	AWS lambda	
Resources / Apparatus Required	Hardware: Computer System	Software: Ubuntu/AWS

Details

What is AWS Lambda?

AWS lambda is a serverless compute service that lets you run your code without worrying about provisioning or managing any server. You can run your application or backend service using AWS scale the infrastructure with high availability.

The code which you run on AWS Lambda is called a lambda function. Currently, it supports the following programming languages.

- Java
- Python
- C#
- Node.js
- Go
- PowerShell
- Ruby

What is serverless?

Serverless most often refers to serverless applications. Serverless applications are ones that don't require you to provision or manage any servers. You can focus on your core product and business logic instead of responsibilities like operating system (OS) access control OS patching, provisioning, right-sizing, scaling and

	availability. By building your application on a serverless platform, the platform manages these responsibilities for you. For service or platform to be considered serverless, it should provide the following capabilities. No server management – You don't have to provision or maintain any servers. There is no software or runtime to install, maintain or administer. Flexile scaling – You can scale your application automatically or y adjusting its capacity through toggling the units of consumption rather than units of individual servers. High availability – Serverless applications have built-in availability and fault tolerance. You don't need to architect for these capabilities because the services running the applications provide them by default. No idle capacity – You don't have to pay for idle capacity. There is no need to pre-provision or over-provision capacity for things like compute and storage. There is no charge when your code isn't running. **AWS Lambda features:** 1. AWS Lambda easily scales the infrastructure without any additional configuration. It reduces the operational work involved. 2. It offers multiple options like AWS $3, CloudWatch, DynamoDB, API Gateway, Kinesis, Code Commit, and many more to trigger an event. 3. You don't need to invest upfront. You pay only for the memory used by the lambda function and minimal cost on the number of requests hence cost-efficient. 4. AWS Lambda is secure. It uses AWS IAM to define all the roles and security policies. 5. It offers fault tolerance for both services running the code and the function. You do not have to worry about the application down.
OUTPUT	**Step 1:** Create IAM Role: Create role with following attached permissions: Iam me jaa aur roles pe jaa aur new roles create kar. • AWSLambdaFullAccess • AmazonS3FullAccess • CloudWatchFullAccess **Step 2**: Amazon s3 pe jaa aur new bucket create kar Create a bucket in AWS S3 to upload image. Ensure region of bucket is same as that of lambda function. Unblock the bolcked acesss untick kar usko Warning aaegi ek usko acknowlege kar **Step 3:** Function scratch se create mat kar nodejs ka blueprint use kar Search kar udhar s3 aur blueprint attribute=s3 select kar

Usme phir nodejs ka code select kar 3 option me ek who select kar

Next step me select use existing role…jo iam se banaye the who wala

Neeche s3 ke option me bas step 2 wale bucket ko select kar aur baaki option default hi rakh..setup karde

Nhi ho rha to alag se trigger main page pe dikhega

Trigger ki configuration aur general configuration me timeout 3sec hai

Edit pe jaa aur time increase kar.20 sec tak

Configuration ke side me hi lambda function me hi

Monitor tab me vies logs in cloudwatch hai uspe jaa

Ab iske side me event hai ek new event create kar

Usse pehle code ko test kar bucket me ek image upload kar aur monitor kar cloudwatch pe

Ab event me jaake new event bana aur bucket ka arn code me daal copy karke…arn matlab naam hi hai

Bucket ka naam bhi change kar

Object hai code me waha image.jpg jo bhi image hoga who daalna hai

Aur etag hota hai ek..woh bucket ke settings se lele

Ab save karke test karle

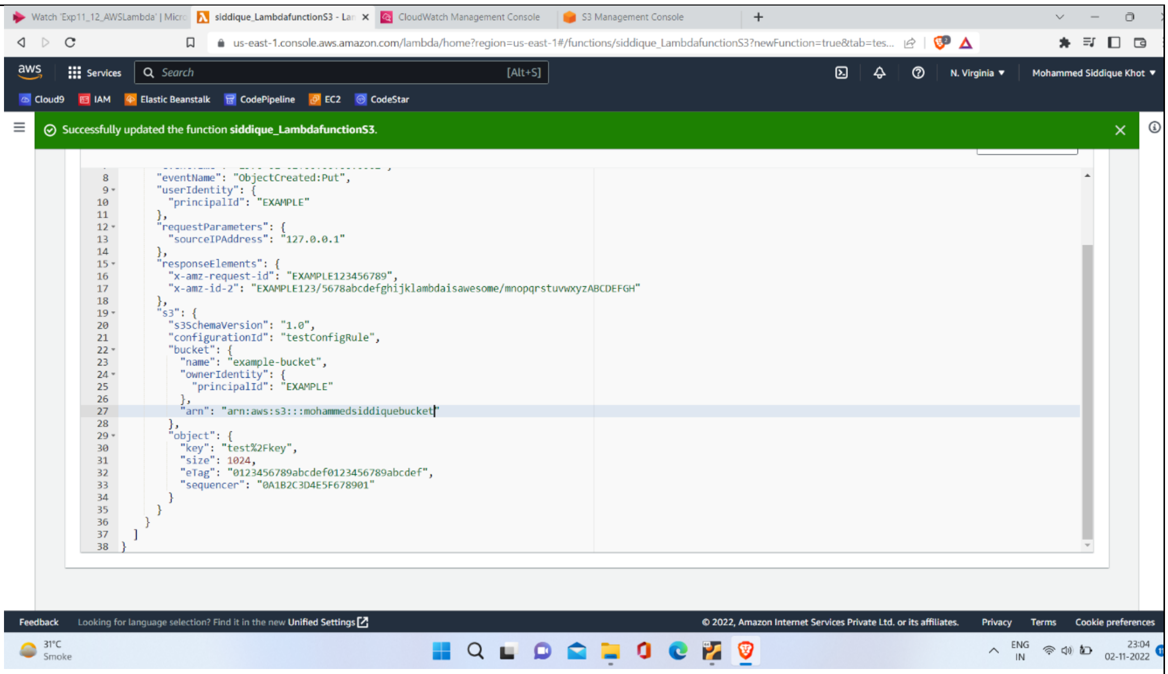
Code option hai test ke side me waha code me 4th image ke jaise changes kar

Deploy and then test

Create a Lambda Function using node.js Blueprint Template for uploading image to s3 Bucket.

Step 4: Create a trigger to invoke creation of logs in Cloud Watch when an image is uploaded in the specified bucket. Test the code and view the logs in Cloud Watch.

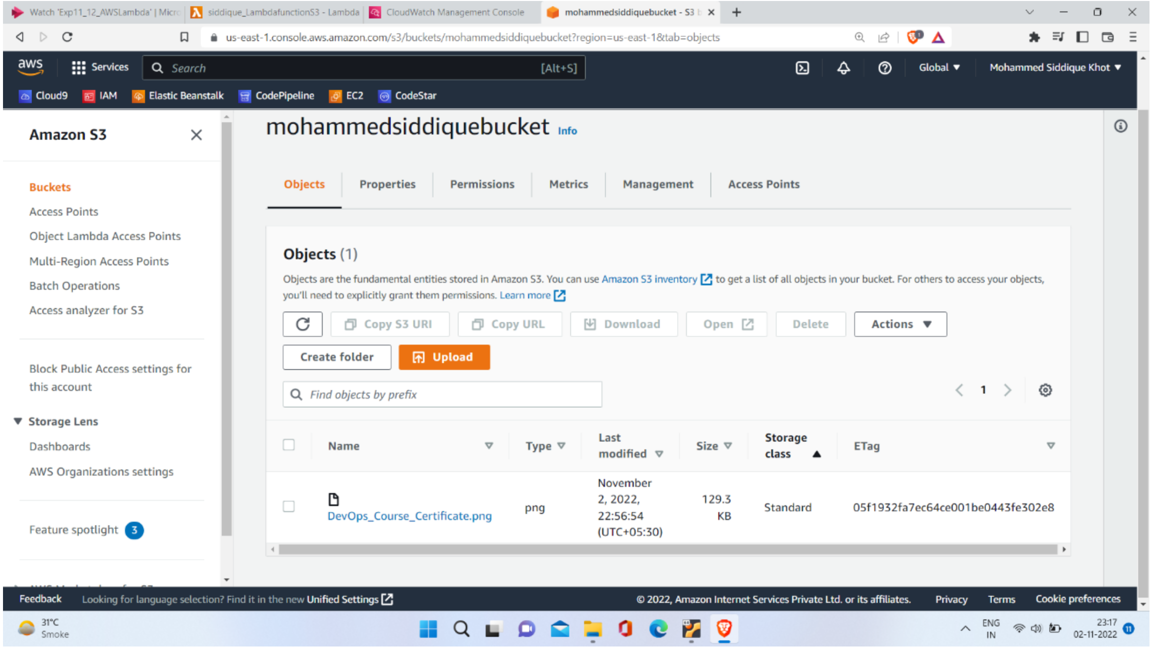
Configure Event name in Test configure button to mention your s3 bucket name and image name to be uploaded


Watch 'Exp11_12_AWSLambda' | Micro
siddique_LambdafunctionS3 - La
CloudWatch Management Console
S3 Management Console
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/siddique_LambdafunctionS3?newFunction=true&tab=tes...
Services
Search
[Alt+S]
N. Virginia
Mohammed Siddique Khot
Cloud9
IAM
Elastic Beanstalk
CodePipeline
EC2
CodeStar
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Smoke
ENG
IN
23:04
02-11-2022

Successfully updated the function **siddique_LambdafunctionS3**.

```
      "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:::mohammedsiddiquebucket"
        },
        "object": {
          "key": "test%2Fkey",
          "size": 1024,
          "eTag": "0123456789abcdef0123456789abcdef",
          "sequencer": "0A1B2C3D4E5F678901"
        }
      }
    }
  ]
}
```

1) **S3 Bucket:** Creation of bucket and later on Image upload to trigger lambda function.


Watch 'Exp11_12_AWSLambda' | Micro
siddique_LambdafunctionS3 - Lambda
CloudWatch Management Console
mohammedsiddiquebucket - S3
us-east-1.console.aws.amazon.com/s3/buckets/mohammedsiddiquebucket?region=us-east-1&tab=objects
Services
Search
[Alt+S]
Global
Mohammed Siddique Khot
Cloud9
IAM
Elastic Beanstalk
CodePipeline
EC2
CodeStar
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
1
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
31°C
Smoke
ENG
IN
23:17
02-11-2022

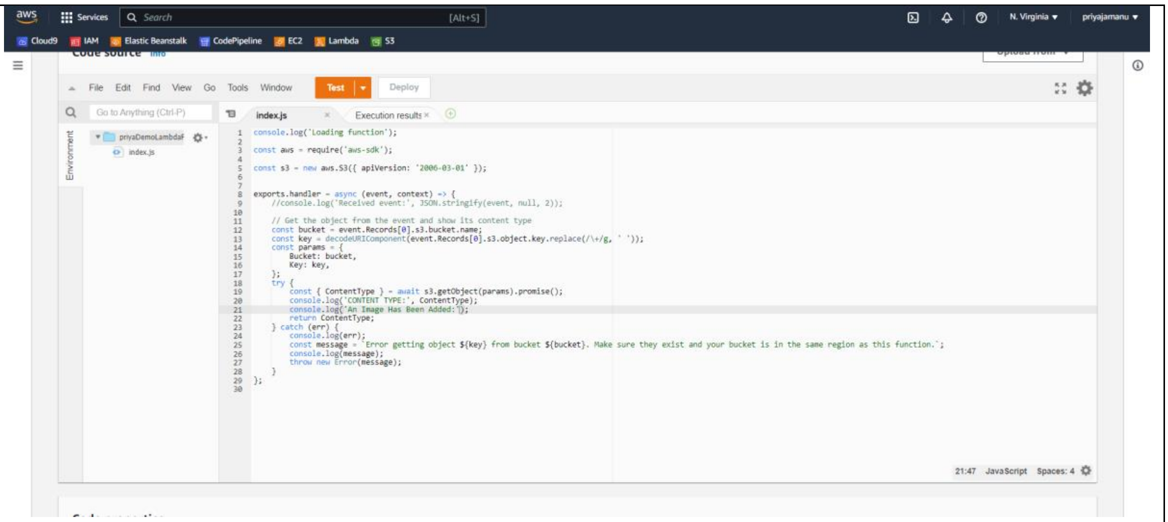
mohammedsiddiquebucket

Objects (1)

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	ETag
□	DevOps_Course_Certificate.png	png	November 2, 2022, 22:56:54 (UTC+05:30)	129.3 KB	Standard	05f1932fa7ec64ce001be0443fe302e8

2) **Lambda Function:**



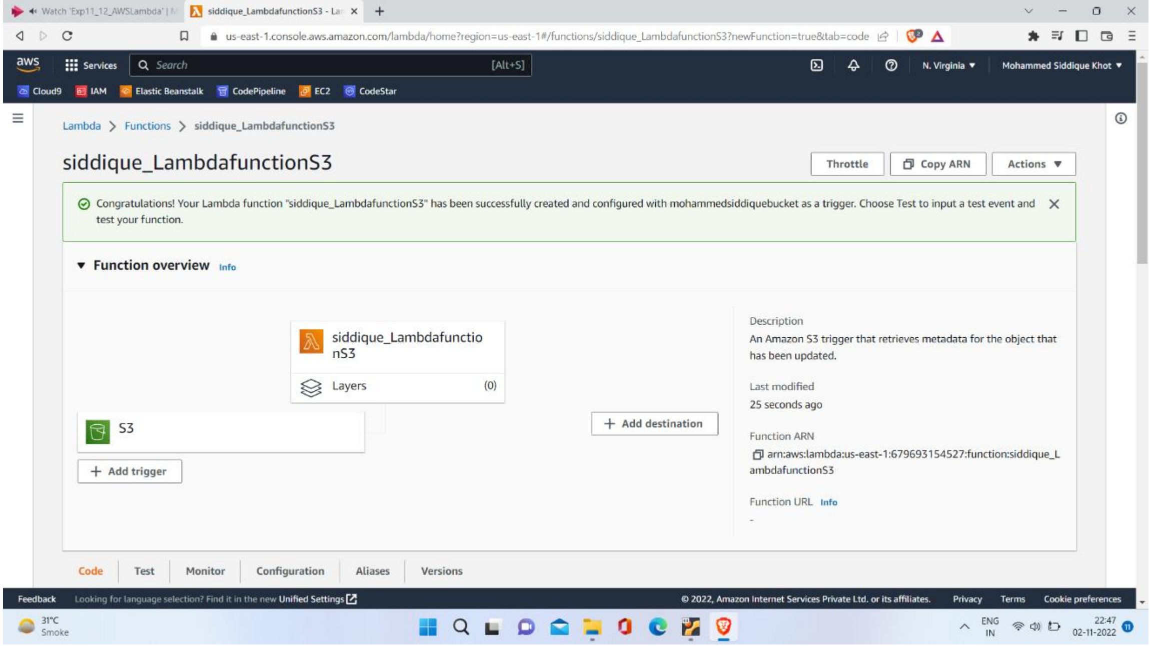
```
console.log('Loading function');

const aws = require('aws-sdk');

const s3 = new aws.S3({ apiVersion: '2006-03-01' });


exports.handler = async (event, context) => {
    //console.log('Received event:', JSON.stringify(event, null, 2));

    // Get the object from the event and show its content type
    const bucket = event.Records[0].s3.bucket.name;
    const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, ' '));
    const params = {
        Bucket: bucket,
        Key: key,
    };
    try {
        const { ContentType } = await s3.getObject(params).promise();
        console.log('CONTENT TYPE:', ContentType);
        console.log('An Image Has Been Added:');
        return ContentType;
    } catch (err) {
        console.log(err);
        const message = `Error getting object ${key} from bucket ${bucket}. Make sure they exist and your bucket is in the same region as this function.`;
        console.log(message);
        throw new Error(message);
    }
};
```



Lambda > Functions > siddique_LambdafunctionS3

siddique_LambdafunctionS3

Congratulations! Your Lambda function "siddique_LambdafunctionS3" has been successfully created and configured with mohammedsiddiquebucket as a trigger. Choose Test to input a test event and test your function.

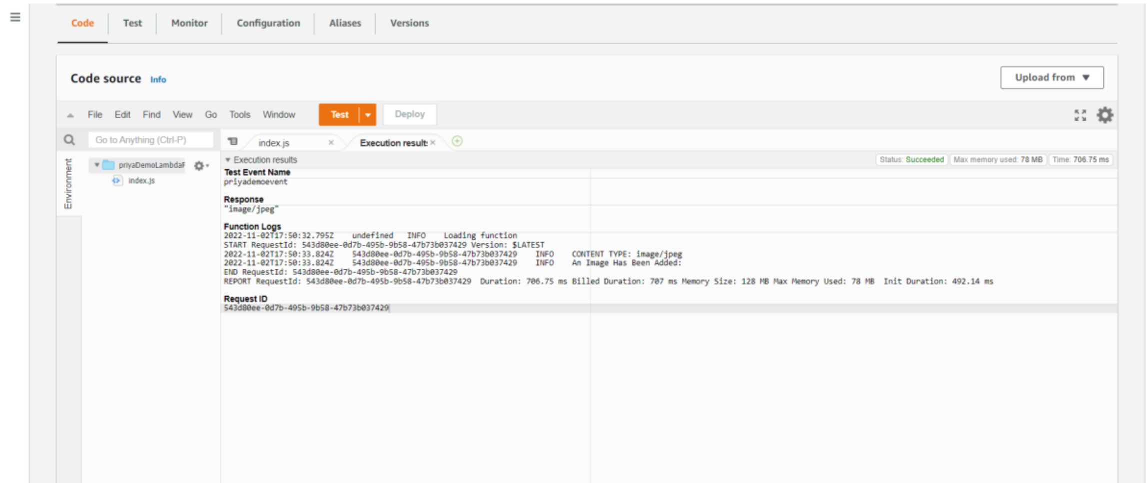
Function overview

Description
An Amazon S3 trigger that retrieves metadata for the object that has been updated.

Last modified
25 seconds ago

Function ARN
arn:aws:lambda:us-east-1:679693154527:function:siddique_LambdafunctionS3

Function URL
-



Execution results

Status: Succeeded | Max memory used: 78 MB | Time: 706.75 ms

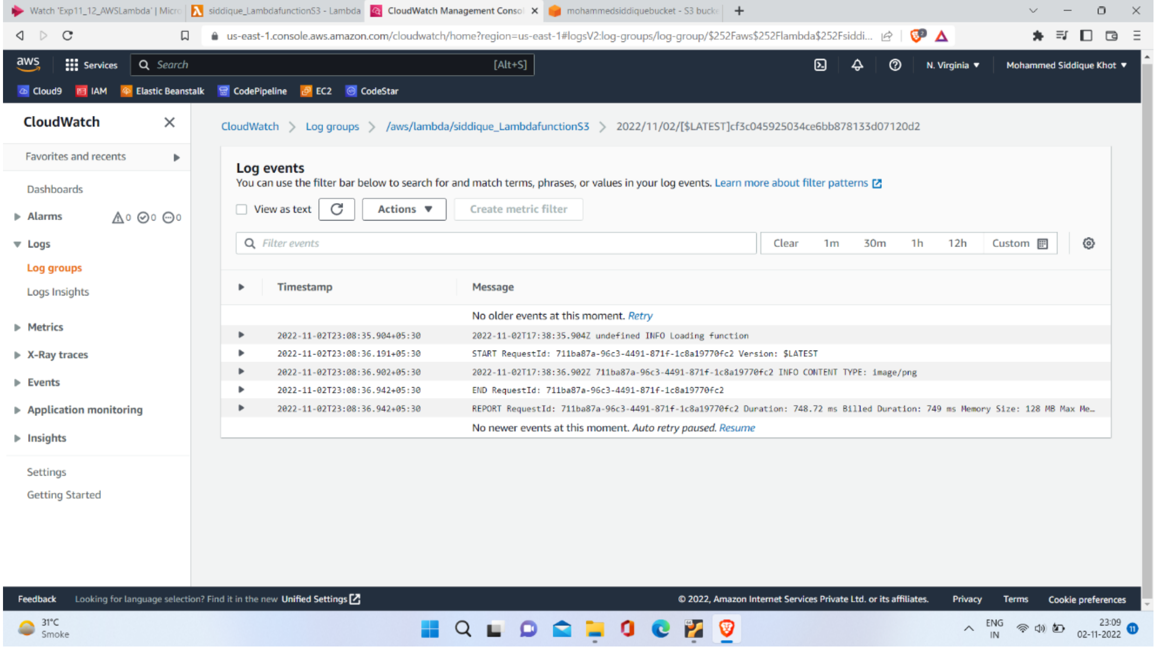
Test Event Name
```
priyademoevent
```

Response
```
"image/jpeg"
```

Function Logs
```
2022-11-02T17:50:32.795Z    undefined    INFO    Loading function
START RequestId: 543d80ee-0d7b-495b-9b58-47b73b037429 Version: $LATEST
2022-11-02T17:50:33.824Z    543d80ee-0d7b-495b-9b58-47b73b037429    INFO    CONTENT TYPE: image/jpeg
2022-11-02T17:50:33.824Z    543d80ee-0d7b-495b-9b58-47b73b037429    INFO    An Image Has Been Added:
END RequestId: 543d80ee-0d7b-495b-9b58-47b73b037429
REPORT RequestId: 543d80ee-0d7b-495b-9b58-47b73b037429  Duration: 706.75 ms Billed Duration: 707 ms Memory Size: 128 MB Max Memory Used: 78 MB  Init Duration: 492.14 ms
```

Request ID
```
543d80ee-0d7b-495b-9b58-47b73b037429
```

3) Cloud Watch Logs:

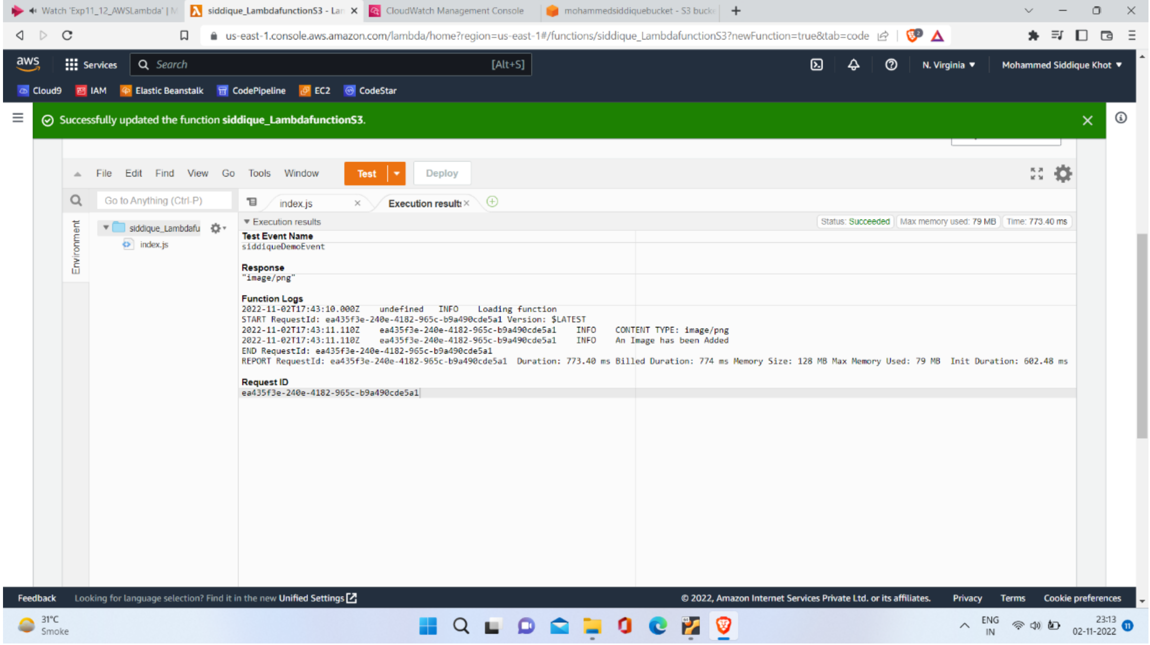


CloudWatch > Log groups > /aws/lambda/siddique_LambdafunctionS3 > 2022/11/02/[$LATEST]cf3c045925034ce6bb878133d07120d2

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

```
Timestamp                      Message
                               No older events at this moment. Retry
2022-11-02T23:08:35.904+05:30  2022-11-02T17:38:35.904Z undefined INFO Loading function
2022-11-02T23:08:36.191+05:30  START RequestId: 711ba87a-96c3-4491-871f-1c8a19770fc2 Version: $LATEST
2022-11-02T23:08:36.902+05:30  2022-11-02T17:38:36.902Z 711ba87a-96c3-4491-871f-1c8a19770fc2 INFO CONTENT TYPE: image/png
2022-11-02T23:08:36.942+05:30  END RequestId: 711ba87a-96c3-4491-871f-1c8a19770fc2
2022-11-02T23:08:36.942+05:30  REPORT RequestId: 711ba87a-96c3-4491-871f-1c8a19770fc2 Duration: 748.72 ms Billed Duration: 749 ms Memory Size: 128 MB Max Me…
                               No newer events at this moment. Auto retry paused. Resume
```



Successfully updated the function siddique_LambdafunctionS3.

Execution results — Status: Succeeded | Max memory used: 79 MB | Time: 773.40 ms

```
Test Event Name
siddiqueDemoEvent

Response
"image/png"

Function Logs
2022-11-02T17:43:10.000Z	undefined	INFO	Loading function
START RequestId: ea435f3e-240e-4182-965c-b9a490cde5a1 Version: $LATEST
2022-11-02T17:43:11.110Z	ea435f3e-240e-4182-965c-b9a490cde5a1	INFO	CONTENT TYPE: image/png
2022-11-02T17:43:11.110Z	ea435f3e-240e-4182-965c-b9a490cde5a1	INFO	An Image has been Added
END RequestId: ea435f3e-240e-4182-965c-b9a490cde5a1
REPORT RequestId: ea435f3e-240e-4182-965c-b9a490cde5a1	Duration: 773.40 ms	Billed Duration: 774 ms	Memory Size: 128 MB	Max Memory Used: 79 MB	Init Duration: 602.48 ms

Request ID
ea435f3e-240e-4182-965c-b9a490cde5a1
```

Conclusion	In this exepriment we learned its workflow,various functions and created first lambda function using Python/java/**node.js**