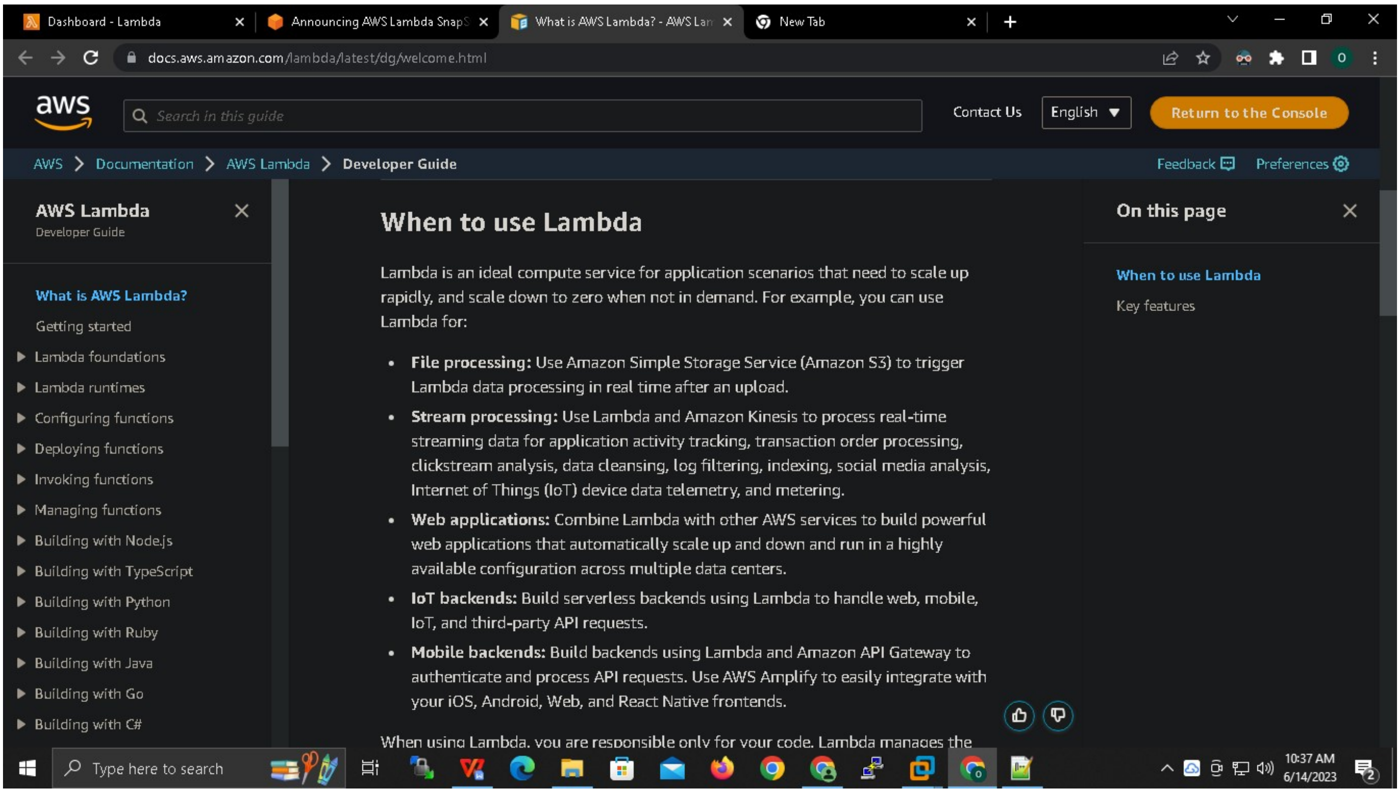
AWS Lambda:



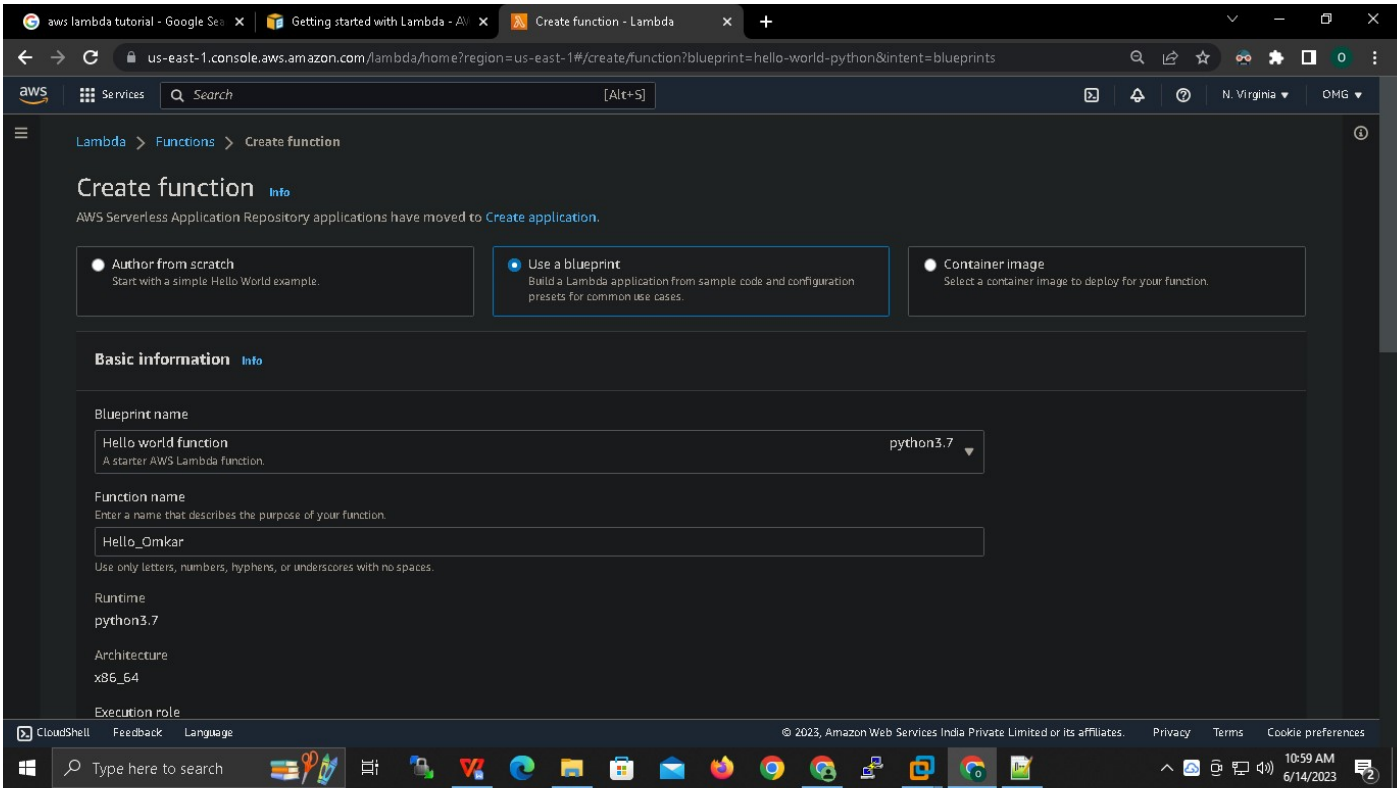
When to use Lambda

Lambda is an ideal compute service for application scenarios that need to scale up rapidly, and scale down to zero when not in demand. For example, you can use Lambda for:

- **File processing:** Use Amazon Simple Storage Service (Amazon S3) to trigger Lambda data processing in real time after an upload.
- **Stream processing:** Use Lambda and Amazon Kinesis to process real-time streaming data for application activity tracking, transaction order processing, clickstream analysis, data cleansing, log filtering, indexing, social media analysis, Internet of Things (IoT) device data telemetry, and metering.
- **Web applications:** Combine Lambda with other AWS services to build powerful web applications that automatically scale up and down and run in a highly available configuration across multiple data centers.
- **IoT backends:** Build serverless backends using Lambda to handle web, mobile, IoT, and third-party API requests.
- **Mobile backends:** Build backends using Lambda and Amazon API Gateway to authenticate and process API requests. Use AWS Amplify to easily integrate with your iOS, Android, Web, and React Native frontends.

When using Lambda, you are responsible only for your code. Lambda manages the

Lambda function ->use a blueprint ->



Create function

AWS Serverless Application Repository applications have moved to Create application.

○ 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.

○ Container image
Select a container image to deploy for your function.

Basic information

Blueprint name: Hello world function — A starter AWS Lambda function. python3.7

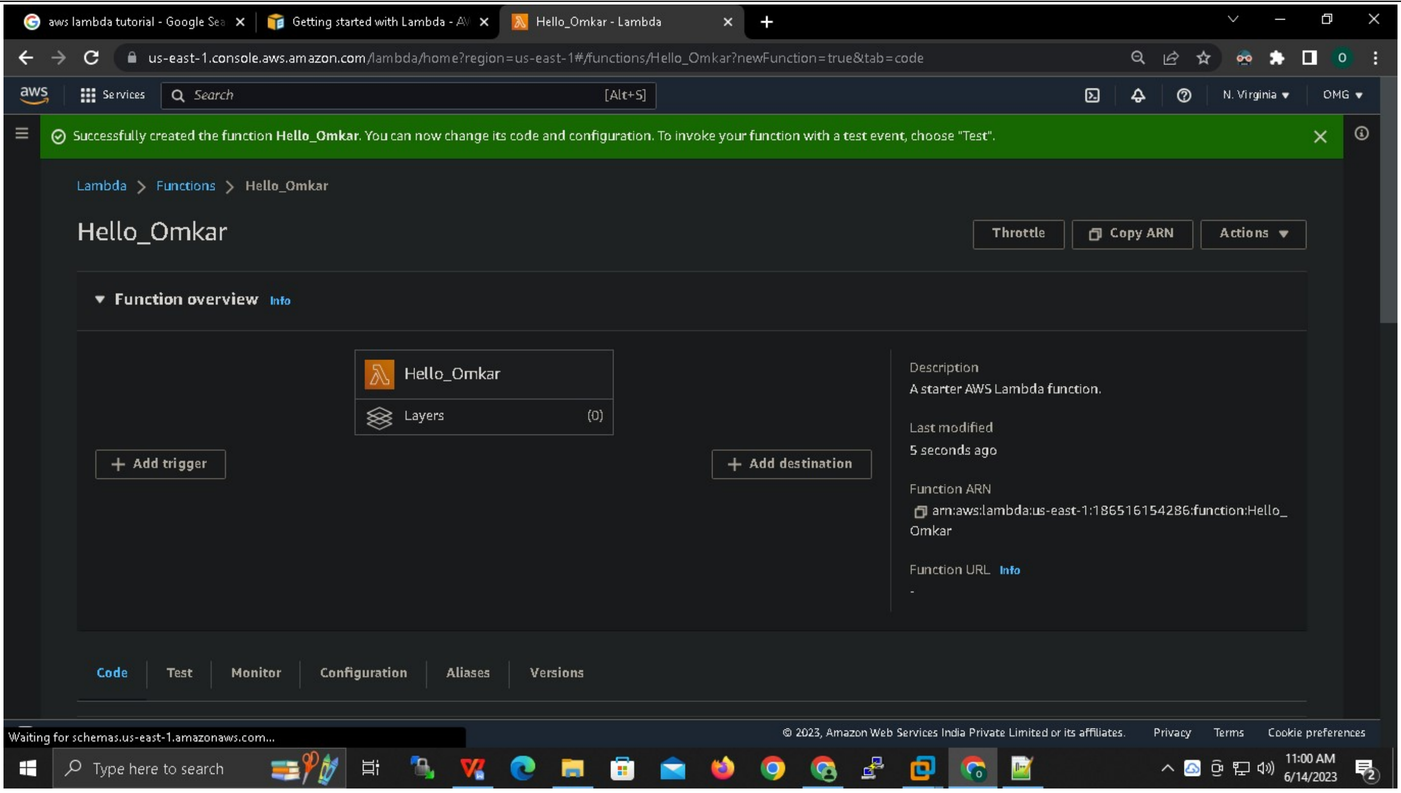
Function name: Hello_Omkar
Enter a name that describes the purpose of your function.
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime: python3.7

Architecture: x86_64

Execution role

us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function?blueprint=hello-world-python&intent=blueprints
Lambda > Functions > Create function
N. Virginia
OMG
10:59 AM
6/14/2023



Successfully created the function **Hello_Omkar**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > Hello_Omkar

Hello_Omkar

Function overview

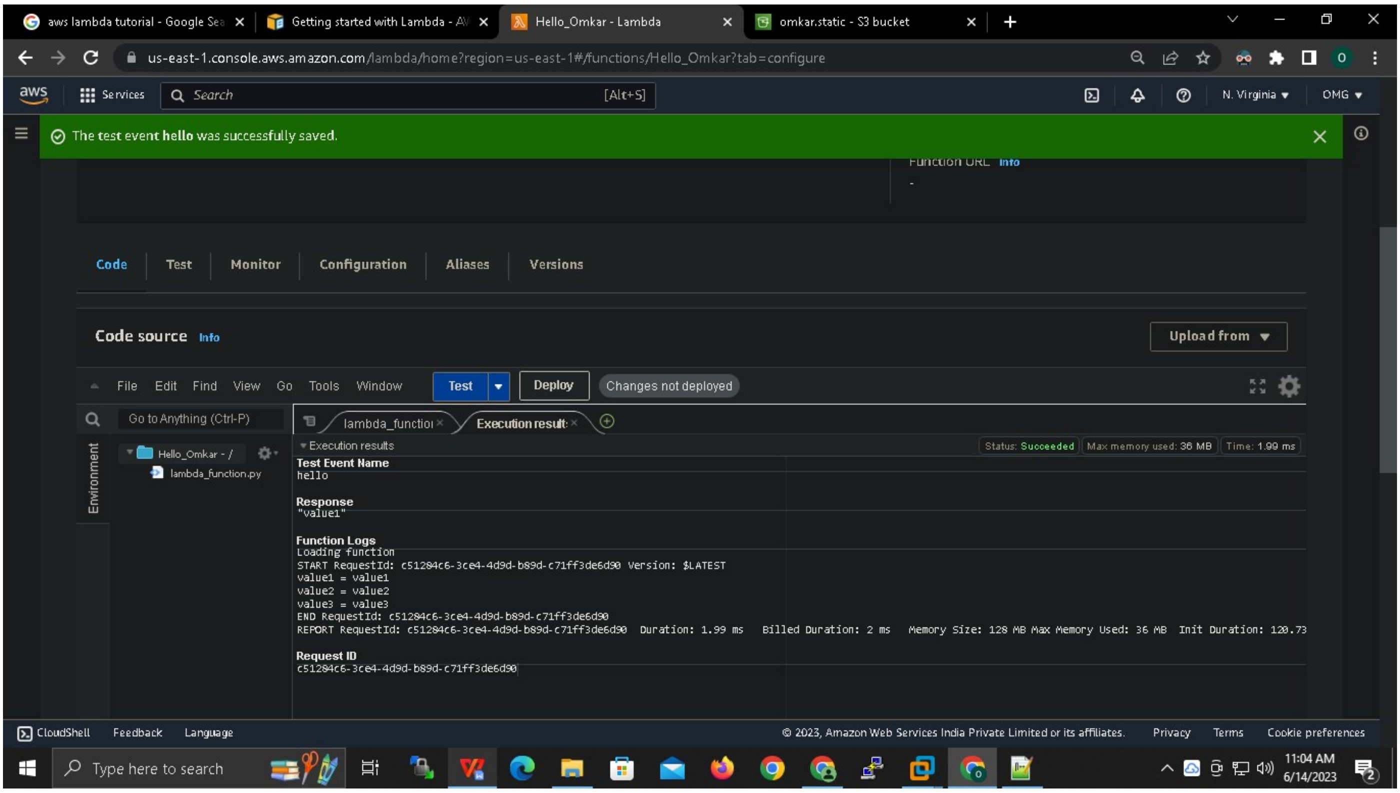
Description
A starter AWS Lambda function.

Last modified
5 seconds ago

Function ARN
arn:aws:lambda:us-east-1:186516154286:function:Hello_Omkar

Function URL
-

There are various option available below it like code , test do required changes and run the code:



The test event **hello** was successfully saved.

Code source

Execution results

Status: Succeeded | Max memory used: 36 MB | Time: 1.99 ms

Test Event Name

```
hello
```

Response

```
"value1"
```

Function Logs

```
Loading function
START RequestId: c51284c6-3ce4-4d9d-b89d-c71ff3de6d90 Version: $LATEST
value1 = value1
value2 = value2
value3 = value3
END RequestId: c51284c6-3ce4-4d9d-b89d-c71ff3de6d90
REPORT RequestId: c51284c6-3ce4-4d9d-b89d-c71ff3de6d90  Duration: 1.99 ms   Billed Duration: 2 ms   Memory Size: 128 MB Max Memory Used: 36 MB  Init Duration: 120.73
```

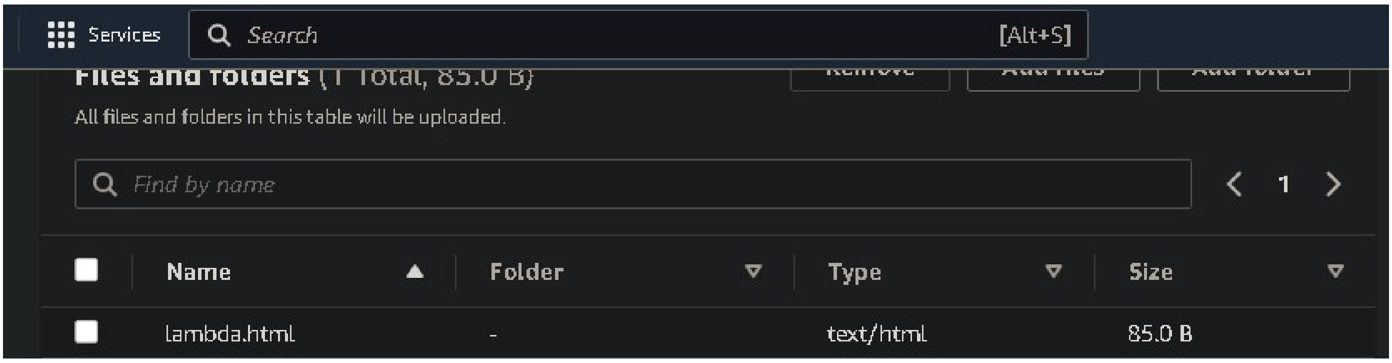
Request ID

```
c51284c6-3ce4-4d9d-b89d-c71ff3de6d90
```

Add a trigger : uploading of file in s3 should run a trigger in lambda ;

To see updation go to cloudwatch :

Go to s3 bucket upload a file :

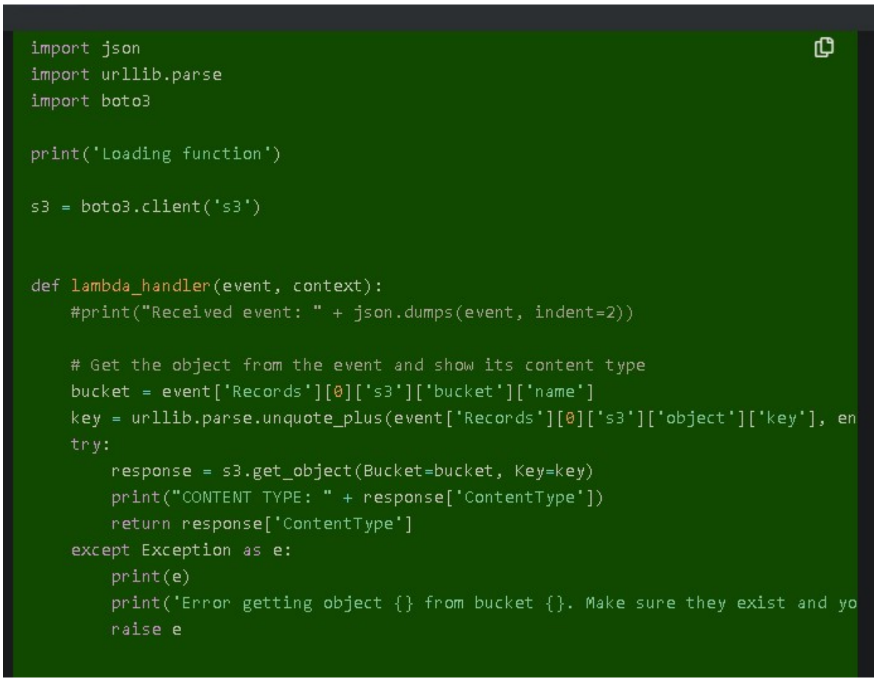


Files and folders (1 Total, 85.0 B)

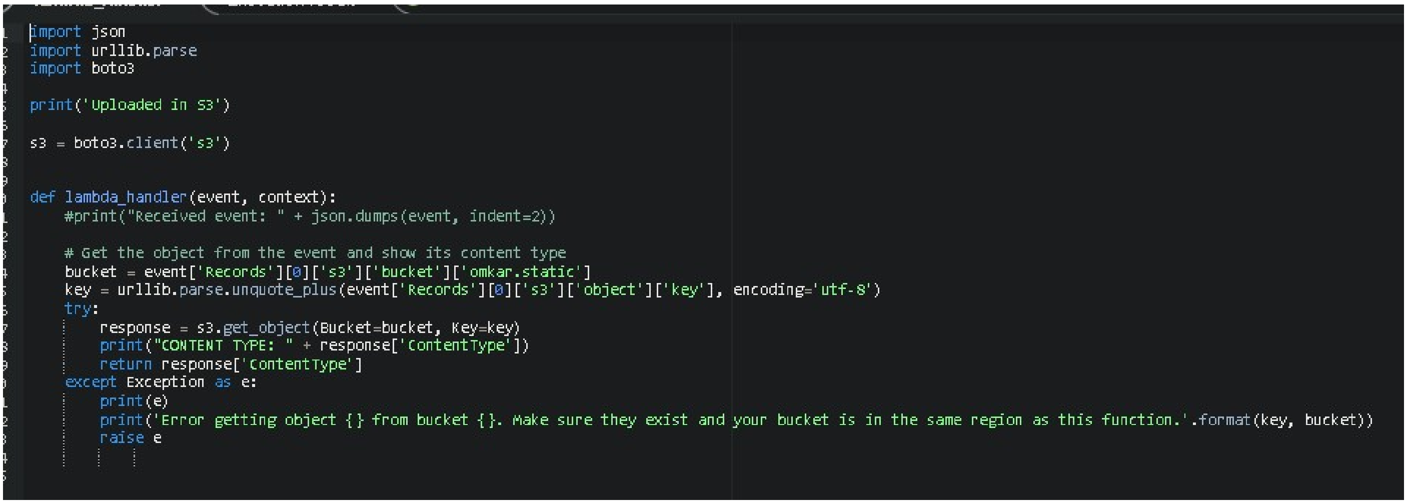
All files and folders in this table will be uploaded.

☐	Name	Folder	Type	Size
☐	lambda.html	-	text/html	85.0 B

Add this code : into lambda code tab :



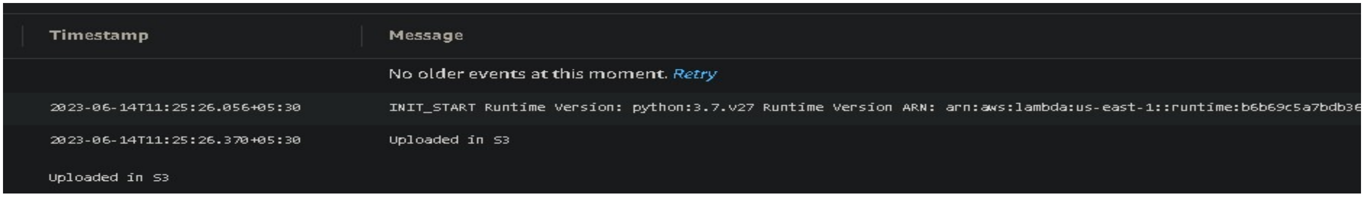
```
import json
import urllib.parse
import boto3

print('Loading function')

s3 = boto3.client('s3')


def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))

    # Get the object from the event and show its content type
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], en
    try:
        response = s3.get_object(Bucket=bucket, Key=key)
        print("CONTENT TYPE: " + response['ContentType'])
        return response['ContentType']
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and yo
        raise e
```



```
import json
import urllib.parse
import boto3

print('Uploaded in S3')

s3 = boto3.client('s3')


def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))

    # Get the object from the event and show its content type
    bucket = event['Records'][0]['s3']['bucket']['omkar.static']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
    try:
        response = s3.get_object(Bucket=bucket, Key=key)
        print("CONTENT TYPE: " + response['ContentType'])
        return response['ContentType']
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(key, bucket))
        raise e
```

Change bucket name in this code :

After that upload the file in s3 bucket and check logs on cloudwatch :



```
Timestamp                          Message
                                   No older events at this moment. Retry
2023-06-14T11:25:26.056+05:30      INIT_START Runtime Version: python:3.7.v27 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:b6b69c5a7bdb3e
2023-06-14T11:25:26.370+05:30      Uploaded in S3
Uploaded in S3
```