

CLOUD CONCEPTS

(Serverless Computing- AWS Lambda)

- **Serverless Computing:** It **does not mean NO SERVER**. It just means that you are not concerned about the infrastructure and other configurations (like you are deploying an application on EC2 instance, you are configuring ... RAM, HDD/SSD, Cores etc.).

Serverless computing is a type of cloud computing that reduces the burden of managing infrastructure to the cloud provider. A developer uploads their code into a Serverless platform (often based on the popular Docker or Kubernetes containers) where the code sits and waits to be triggered by an external event. Once the code has completed it is no longer left running. The developer or operations teams do not need to manage the environment. The Serverless platform provider is responsible to ensure that there are sufficient computing resources to handle spikes in usage including processing power and storage. The Serverless computing provider deals with patching and all aspects of infrastructure management so that the developer does not have to. Serverless lends itself well to small discrete portions of code known as *functions* that perform a short lived defined task (often lasting less than five minutes). Another significant benefit of adopting Serverless Computing is that you pay for what you consume often in the cents rather than paying for a predefined volume of server resources. Serverless computing has become popular by the adoption of Amazon's Lambda and the Microsoft Azure Functions environments.

- **SERVER IS THERE...! JUST WRITE YOUR SCRIPT.**

Event-driven architecture (EDA) is a software component, or event, which executes a response to receiving one or more event notifications (or triggers).

Serverless computing is a type of cloud computing where the customer does not have to provision servers for code to run on. The cloud provider starts and stops a container platform as a service as requests are triggered. Consumption is often billed on a per second usage model.

Function as a service (FaaS) refers to cloud services that enable serverless app development and management. FaaS users are able to program and deploy their code without having to manage their own server(s). Code is triggered by remote events (such as by a mobile app) with code execution occurring in the Serverless environment rather than on the end user device.

Docker container is an open source software development platform that provides developer/operations (DevOps) staff to package applications in **containers**. Containers provide application portability between different application hosting providers that support applications built for the Docker platform. Unlike with Serverless computing environments the use of containers such as Docker require that users still must ensure that their application remains up and running.

- AWS provide two serverless engines:
 - AWS Lambda (*run your code i.e. simple scripts... .py or .java*)
 - NOT for something running all the time (24 x 7)
 - Good for intermittent work load when its not sure how much resources are required. E.g.,
 - Add servers if load is increased
 - Generate a notification when file is uploaded in S3
 - AWS Fargate (*to run your containers or dockerized applications*)

- **AWS Lambda** helps you run your code without managing or provisioning the servers. You will not have access to the server but all the resources and infrastructure is in use.
- It is a regional service you cannot trigger lambda from a resource or service in another region.
 - **Lambda function** is a code that runs on AWS Lambda. When you upload your code on AWS Lambda it is stored on AWS S3 as an encrypted zip file which runs in its own isolated environment, that separates resources and infrastructure.
 - For troubleshooting or in case of any errors CloudWatch logs are used. Whenever lambda function is created an automatic log group is created in CloudWatch which stores the logs as lambda runs.
- Lambda also scales automatically without any delays. For Lambdas the code should be stateless i.e. no dependency on infrastructure or underlying file system. In order to store any stateful information other stateful resources like S3 or DynamoDB can be used.
- Some of the facilities that Lambda provides are:
 - ✓ completely automated. (all the information, deployment, maintenance, patching, built-in logging, monitoring is managed)
 - ✓ focuses on code and business logic. (You don't worry about scaling or pricing)
 - ✓ Makes it simple to apply compute to data. (You can extend AWS service with your custom logic i.e., Trigger your lambda code whenever you upload file on S3 or any entry changes in DynamoDB OR other service)
 - ✓ Build your custom backend services. (backend services that act as API endpoints using API gateways triggered on demand)
 - ✓ Maintain compute capacity across multiple AZs in same region. (Protecting your code from any datacenter failure providing high availability, low maintenance window and less downtime)
 - ✓ It can autoscale as soon as the request is received and launch multiple instances in no time to manage high request rate.
 - ✓ Lambda has integrated security. (Runs in a VPC by default)
- Languages supported by lambda are NodeJS, Python, Java, Ruby, Go, .NET(C#). AWS provides SDKs for all these languages to integrate with other AWS services.
- Memory of lambda can be from 128 MB to 3008 MB with 64 MB increment on basis of which AWS allocates the CPU capacity.
- Time limit of code execution is max 15 min (3 sec to 15 min) after that it times out.
- Every lambda comes with IAM role attached to it which determines the permissions of your function.
- For monitoring CloudWatch is attached with lambda.
- Define your environment variables as key-value pairs accessible from your function code can be used to save configuration settings.
- Event source can be a service/application that generates events to trigger lambda. This can be of three types.
 - **Lambda reads the events** from the service or the source
 - **Synchronous Lambda** triggers lambda and waits for the response.
 - **Asynchronous Lambda** triggers the lambda and does not wait for the response. (other service does not care what lambda is doing)
 - Asynchronous invocation handles retries.

| Lambda reads events from | Invoke Lambda functions synchronously | Invoke Lambda functions asynchronously |
|--------------------------|---------------------------------------|--|
| 1. DynamoDB | 1. ELB & ALB | 1. S3 |
| 2. Kinesis | 2. Cognito | 2. SNS |
| 3. MQ | 3. Lex | 3. SES |
| 4. Apache Kafka | 4. Alexa | 4. CloudFormation |
| 5. SQS | 5. API Gateway | 5. CloudWatch Logs |
| | 6. CloudFront | 6. CloudWatch Events |
| | 7. Kinesis | 7. CodeCommit |
| | 8. S3 Batch | 8. Config |
| | | 9. IoT & IoT Events |
| | | 10. CodePipeline |

- **Lambda Layers:** In order to access the common custom code then publish the code as layers and add them to lambda. At most 5 layers can be added. It will keep the code simple and place the code as libraries at separate place. You can also use the layers published by others.
- **Lambda step function:** Lambda has limited memory and timeout limits so in order to address complex long running tasks combine multiple lambda functions in workflow called step function. These complex long running tasks can be run in parallel or sequentially.
- **Provisioned concurrency** can help reducing response time too low. To make sure very low latency this keeps the function initialized and hyper ready to respond quickly. It does not require any code changes but only a configuration change. It comes with additional cost.
- **Lambda@edge** allows you to run your lambda globally (if your customers are distributed globally). Simple lambda function only runs in the region where it is deployed.
 - Trigger for lambda@edge is only CloudFront and all the information required to process should be present in the function and request itself.
- **Database proxies** help you handle many concurrent database connection requests. E.g., Lambda can connect with Amazon RDS which is a database service involving database connection for every invocation. Here high level of concurrency management is required which becomes difficult to handle. Database proxies can have pool of database connections and relay queries to the functions.

Pricing

You pay only for what you use

Number of Requests

\$0.20 per 1M requests

Duration

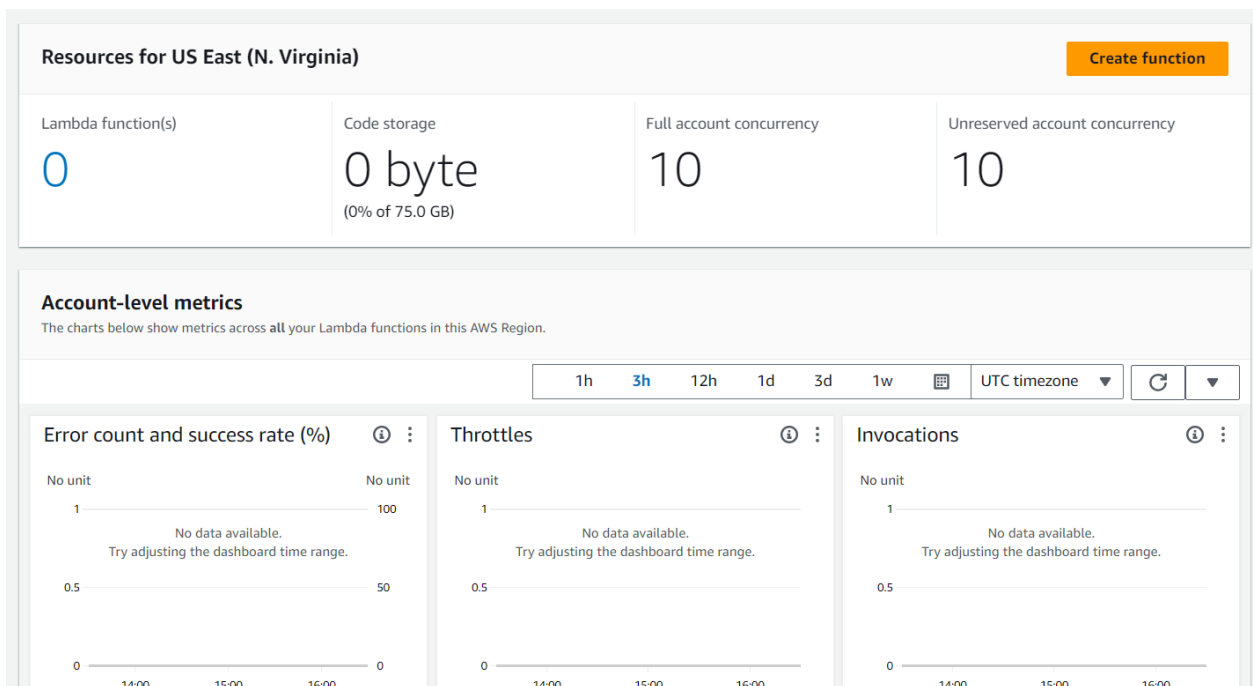
\$0.0000166667 for every GB-second

Provisioned Concurrency Pricing

additional costs for that duration

[Lab \(AWS Lambda\)](#)

1. Search for Lambda service and create the lambda function. Write the name and select the language of your choice.



Lambda > Functions > Create function

Create function [Info](#)

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

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.11

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64
☐ arm64

- Lambda will create the IAM (execution) role with some lambda permissions. You can use any existing role or create a new role.

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☒ Create a new role with basic Lambda permissions
☐ Use an existing role
☐ Create a new role from AWS policy templates

i Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named <myFunctionName>-role-c41xpvk3, with permission to upload logs to Amazon CloudWatch Logs.

- In advanced settings it has some options one is code signing which make sure that code is signed by an authority and will not be edited by anyone. And you can also specify the VPC in which you want your labda to be launched (leave them blank). Click create function.

▼ **Advanced settings**

☐ **Enable Code signing** [Info](#)
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

☐ **Enable function URL** [Info](#)
Use function URLs to assign HTTP(S) endpoints to your Lambda function.

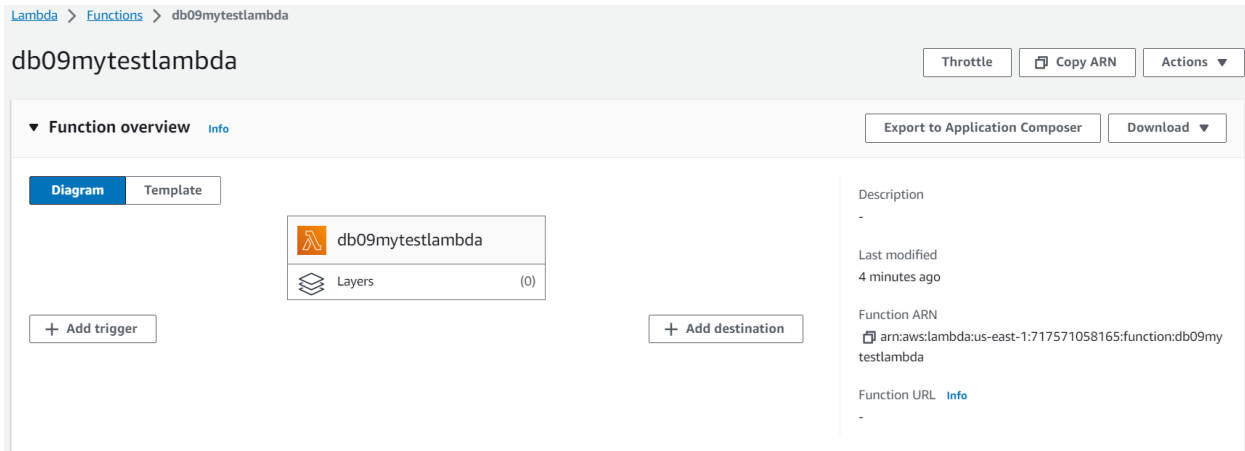
☐ **Enable tags** [Info](#)
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources, track your AWS costs, and enforce attribute-based access control.

☐ **Enable VPC** [Info](#)
Connect your function to a VPC to access private resources during invocation.

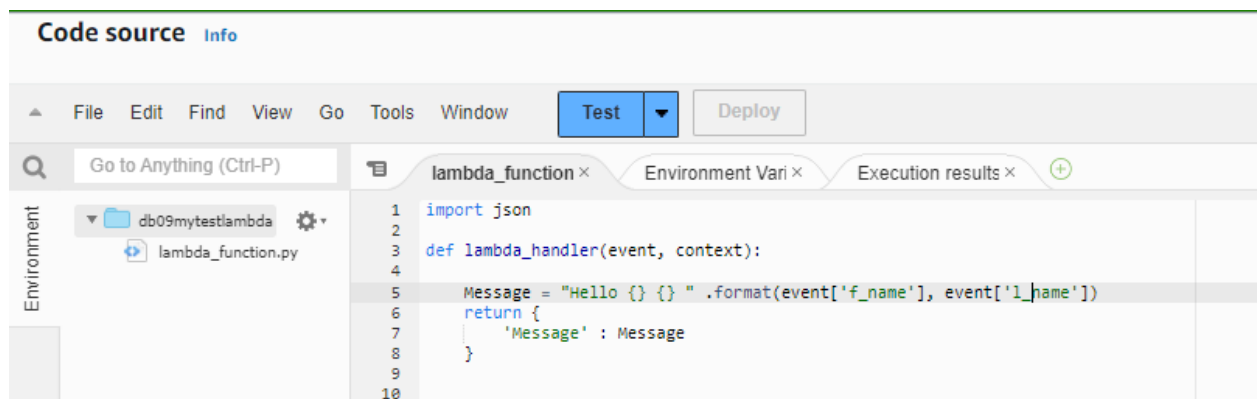
Cancel

- Lambda is created with some basic code.
The main function name is `lambda_handler` which takes two arguments
 - `event`: data from the event that triggered that function

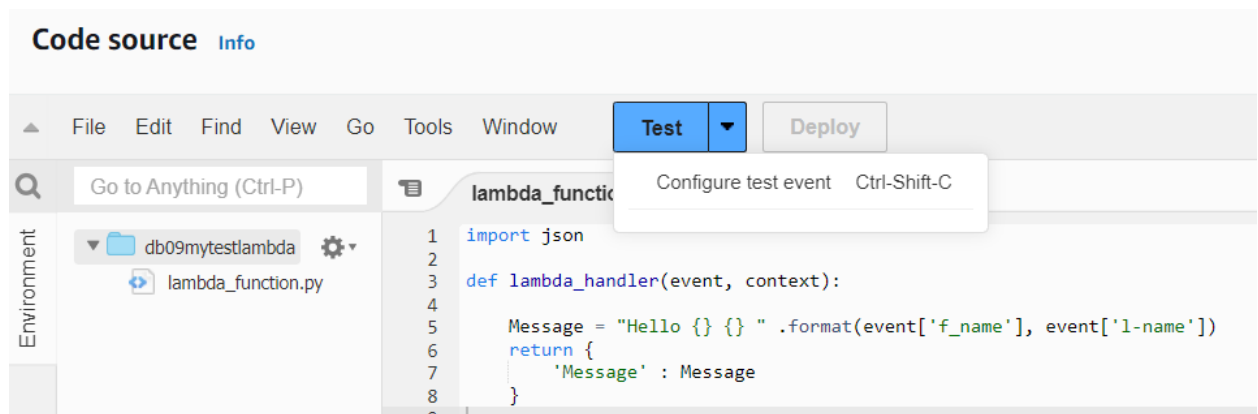
- a. e.g. trigger this lambda when file is uploaded in the s3 bucket
2. context: data about the execution environment of the function.
 - a. Files, code (python), roles, permission etc.



5. Change the default code with some simple script and deploy the code.



6. Configure the test event and save it.



Configure test event

×

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event

☐ Edit saved event

Event name

db09mytestevent

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☒ Private
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☐ Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

Event JSON

Format JSON

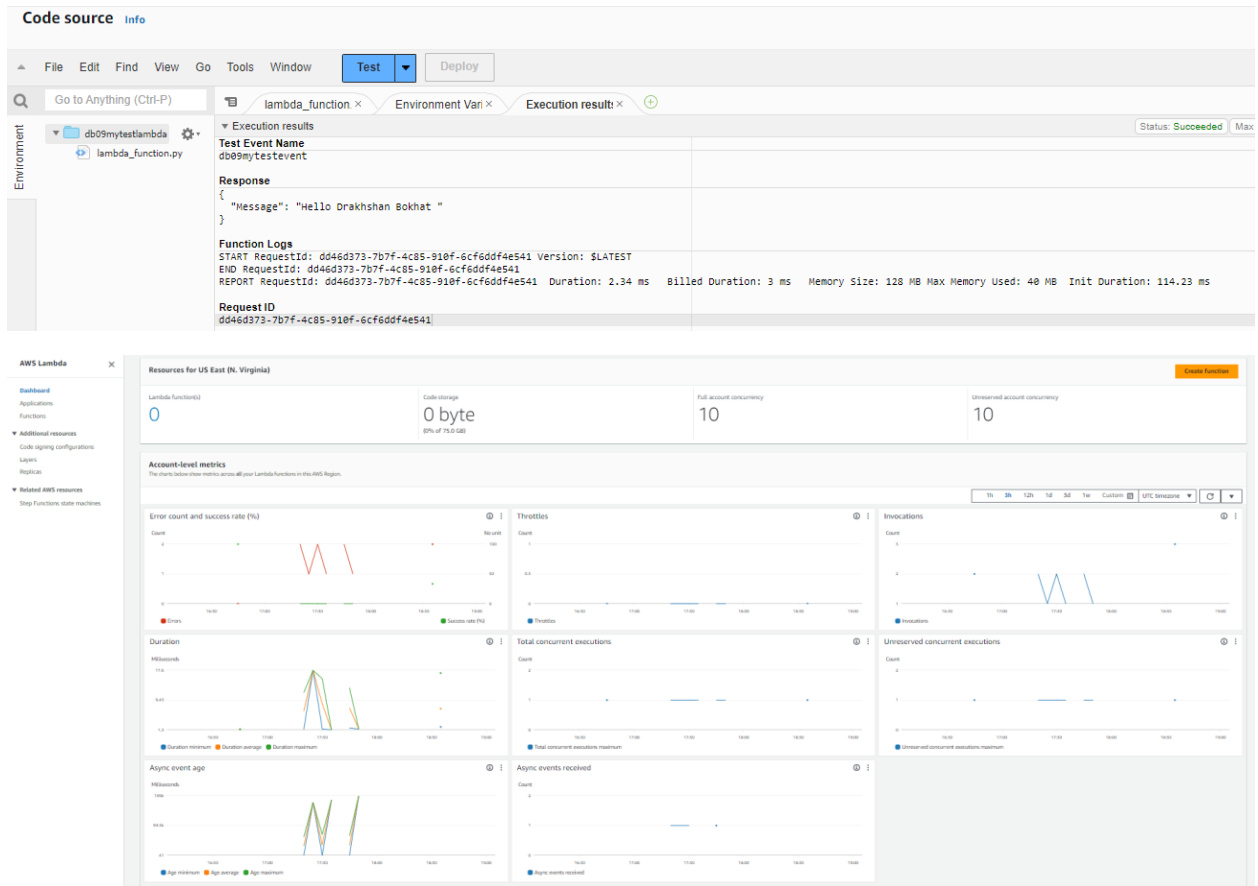
```
1 {  
2   "f_name": "Drakhshan",  
3   "l_name": "Bokhat"  
4 }
```

Cancel

Invoke

Save

7. Now test the event. See the logs through cloudwatch for invocations.



8. You can create a different lambda function for the case: “Whenever a .txt file is uploaded lambda must be triggered” or “Whenever a .txt file is uploaded lambda must be triggered and convert the .txt file in S3 bucket to .pdf and place it in another bucket.”

You need to write the python code for this.

- ✿ [a] Code for printing the name of bucket in which file is uploaded (copied code).
- ✿ [b] Another source: <https://repost.aws/knowledge-center/lambda-copy-s3-files>

```

1 import json
2 import urllib.parse
3 import boto3
4
5 print('loading function')
6
7 s3 = boto3.client('s3')
8
9 def lambda_handler(event, context):
10     #print("Received event: " + json.dumps(event, indent=2))
11
12     # Get the object from the event and show its content type
13     bucket = event['Records'][0]['s3']['bucket']['name']
14     print(bucket)
15     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
16     try:
17         response = s3.get_object(Bucket=bucket, Key=key)
18         print("CONTENT TYPE: " + response['ContentType'])
19         return response['ContentType']
20     except Exception as e:
21         print(e)
22         print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function')
23         raise e
24

```


[Lambda](#) > [Functions](#) > db09myLamdaS3handler

db09myLamdaS3handler

ThrottleCopy ARNActions

Function overviewInfo

Export to Application ComposerDownload function

DiagramTemplate

db09myLamdaS3handler

Layers(0)

+ Add trigger+ Add destination

Description-

Last modified25 seconds ago

Function ARNarn:aws:lambda:us-east-1:717571058165:function:db09myLamdaS3handler

Function URLInfo-

Code sourceInfo

Upload from

FileEditFindViewGoToolsWindowTestDeploy

Go to Anything (Ctrl-P)

Environment

db09myLamdaS3ha
lambda_function.py

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO Implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')
8     }
9
```

Code propertiesInfo

| | | |
|----------------------------|---|--|
| Package size 299.0 byte | SHA256 hash HAPq9EReJVECSgLavtc/gyd5vZtd9eiUGF932t0jBxY= | Last modified November 24, 2023 at 09:40 PM GMT+5 |
|----------------------------|---|--|

Runtime settingsInfo

EditEdit runtime management configuration

| | | |
|------------------------|--|-----------------------------|
| Runtime Python 3.11 | Handler Info lambda_function.lambda_handler | Architecture Info x86_64 |
|------------------------|--|-----------------------------|

Runtime management configuration

LayersInfo

EditAdd a layer

| Merge order | Name | Layer version | Compatible runtimes | Compatible architectures | Version ARN |
|------------------------------|------|---------------|---------------------|--------------------------|-------------|
| There is no data to display. | | | | | |

9. Now set up the trigger for S3.

[Lambda](#) > Add trigger


Add trigger


Trigger configuration [Info](#)


Select a source


Q |


APIs/interactive/web


 **Alexa**
alexa iot voice

 **API Gateway**
aws api application-services backend HTTP REST
serverless

 **Application Load Balancer**
aws HTTP load-balancing server web

 **CloudFront**
cdn edge

 **CodeCommit**
aws asynchronous developer-tools git

 **Cognito Sync Trigger**
aws authentication frontend identity mobile sync

Cancel


Add

E & OE
Handouts: Drakhshan Bokhat

[Lambda](#) > Add trigger

Add trigger

Trigger configuration [Info](#)



S3
aws asynchronous storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

Bucket must be in 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 ✕

10. Add the buckets if not added.

Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket
Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

[Amazon S3](#) > Buckets

Account snapshot
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

Buckets (1) [Info](#)
Buckets are containers for data stored in S3. [Learn more](#)

| | Name | AWS Region | Access | Creation date |
|-----------------------|--------------|---------------------------------|-------------------------------|---|
| <input type="radio"/> | db09mybucket | US East (N. Virginia) us-east-1 | Bucket and objects not public | November 24, 2023, 21:46:32 (UTC+05:00) |

11. Select the bucket created.

Add trigger

Trigger configuration [Info](#)



S3

aws asynchronous storage

Bucket

Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

×
↺

Use: "s3/db09mybucket"

db09mybucket

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 ×

12. Add suffix .txt, lambda will be triggered for these types of files only. Also acknowledge that lambda will not loop (will cost more).

Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

.txt

Recursive invocation

If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

- ☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel

Add

13. Trigger is added.

db09myLamdaS3handler

ThrottleCopy ARNActions

The trigger db09mybucket was successfully added to function db09myLamdaS3handler. The function is now receiving events from the trigger.

Function overviewInfo

Export to Application ComposerDownload function

DiagramTemplate

db09myLamdaS3handler

Layers(0)

S3

+ Add trigger

+ Add destination

Description

-

Last modified

10 minutes ago

Function ARN

arn:aws:lambda:us-east-1:717571058165:function:db09myLamdaS3handler

Function URL

Info

14. You can test the code by creating the test events and running the test events. The default code of lambda handler will show the “hello from lambda”. The test event can also be deleted.

Configure test event



A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event

☐ Edit saved event

Event name

db09mytestevent

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☒ Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☐ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

Event JSON

Format JSON

```
1 {  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3"  
5 }
```

Cancel

Invoke

Save

Code source Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

db09myLambdaS3h
lambda_function.py

Execution results

Test Event Name
db09mytestevent

Response
{
 "statusCode": 200,
 "body": "\\Hello from Lambda!\\"
}

Function Logs
START RequestId: 91f21b4f-7dd9-4a61-b54c-29cfad6fc47b Version: \$LATEST
END RequestId: 91f21b4f-7dd9-4a61-b54c-29cfad6fc47b
REPORT RequestId: 91f21b4f-7dd9-4a61-b54c-29cfad6fc47b Duration: 1.46 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 40 MB Init Duration: 159.23 ms

Request ID
91f21b4f-7dd9-4a61-b54c-29cfad6fc47b

Status: Succeeded

Configure test event

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

☐ Create new event

☒ Edit saved event

Event name

db09mytestevent



Delete



Delete event db09mytestevent from db09myLambdaS3handler?

Confirm deletion of test event db09mytestevent to delete.

Permanently delete

Event JSON

Format JSON

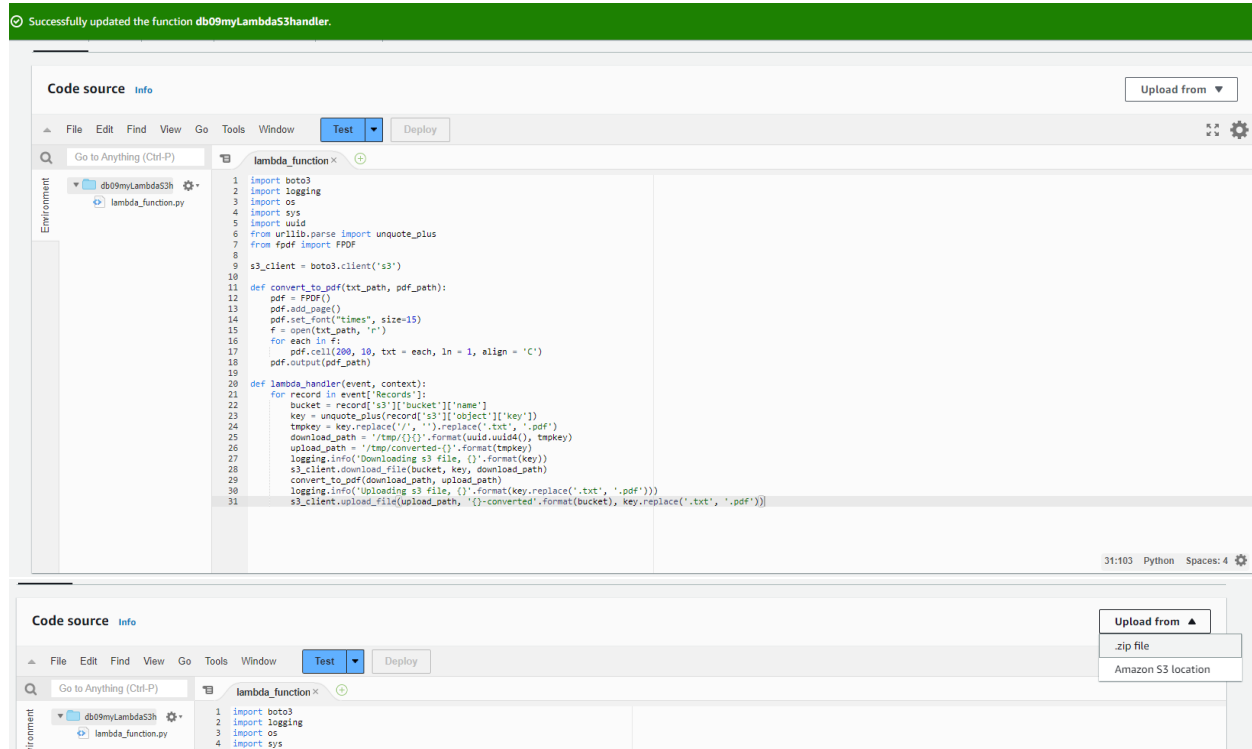
```
1 {  
2   "key1": "value1",  
3   "key2": "value2",  
4   "key3": "value3"  
5 }
```

Cancel

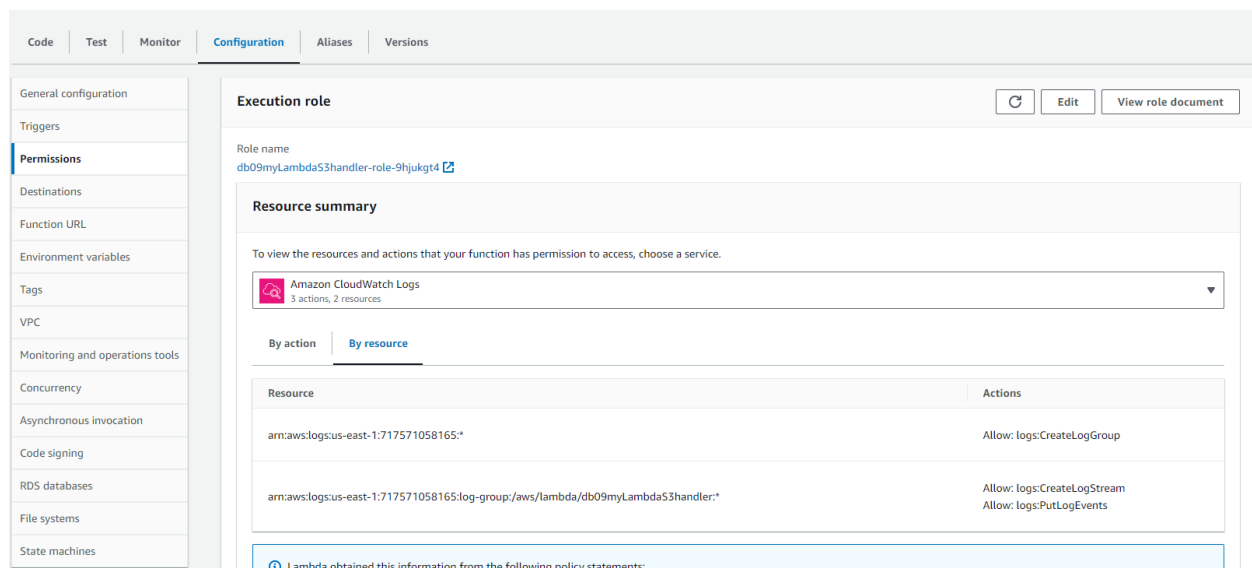
Invoke

Save

15. Add the lambda function as per the use case. (use case: Convert the .txt file into .pdf when uploaded). Change the lambda handler and deploy the code. You can also upload the code (python code) .zip file.



16. To see the IAM role attached to lambda in the permission section. You can see the default role created at the time of lambda creation.



17. Modify the IAM role to get the object from the S3 bucket and put it into other S3 bucket.

[IAM](#) > [Roles](#) > db09myLambdaS3handler-role-9hjukt4

db09myLambdaS3handler-role-9hjukt4 [Info](#)

Introducing Amazon Q
Receive guidance, get troubleshooting tips, and learn about AWS services and capabilities.

Edit

Summary

Creation date
December 07, 2023, 21:42 (UTC+05:00)

Last activity
-

ARN
arn:aws:iam::717571058165:role/service-role/db09myLambdaS3handler-role-9hjukt4

Maximum session duration
1 hour

[Permissions](#) | [Trust relationships](#) | [Tags](#) | [Access Advisor](#) | [Revoke sessions](#)

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Filter by Type

All types

Search

< 1 > ⚙

☐

Policy name

▲

Type

▼

Attached entities

▼

☐

☒ AWSLambdaBasicExecutionRole-9b9b04db-7509-44c1-af95...

Customer managed

1

► Permissions boundary (not set)

18. Click on the policy name shown and add the permissions. The default permission created can be seen. You can see the JSON format as well for S3 handler.

[IAM](#) > [Policies](#) > AWSLambdaBasicExecutionRole-9b9b04db-7509-44c1-af95-b272781910c9

AWSLambdaBasicExecutionRole-9b9b04db-7509-44c1-af95-b272781910c9 [Info](#)

Delete

Policy details

Type
Customer managed

Creation time
December 07, 2023, 21:42 (UTC+05:00)

Edited time
December 07, 2023, 21:42 (UTC+05:00)

ARN
arn:aws:iam::717571058165:policy/service-role/AWSLambdaBasicExecutionRole-9b9b04db-7509-44c1-af95-b272781910c9

[Permissions](#) | [Entities attached](#) | [Tags](#) | [Policy versions \(1\)](#) | [Access Advisor](#)

Permissions defined in this policy [Info](#)

Edit | Summary | JSON

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

Search

Show remaining 401 services

Allow (1 of 402 services)

| Service | Access level | Resource | Request condition |
|---------------------------------|----------------|----------|-------------------|
| CloudWatch Logs | Limited: Write | Multiple | None |

Permissions defined in this policy [info](#)

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "logs:CreateLogGroup",
7       "Resource": "arn:aws:logs:us-east-1:717571058165:*"
8     },
9     {
10      "Effect": "Allow",
11      "Action": [
12        "logs:CreateLogStream",
13        "logs:PutLogEvents"
14      ],
15      "Resource": [
16        "arn:aws:logs:us-east-1:717571058165:log-group:/aws/lambda/db09myLambdaS3handler:*"
17      ]
18    }
19  ]
20 }

```

Copy Edit Summary JSON

19. Edit the policy. You can see the cloudwatch log monitoring Allowed in the policy.

Modify permissions in AWSLambdaBasicExecutionRole-9b9b04db-7509-44c1-af95-b272781910c9 [info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

Visual JSON Actions

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "logs:CreateLogGroup",
7       "Resource": "arn:aws:logs:us-east-1:717571058165:*"
8     },
9     {
10      "Effect": "Allow",
11      "Action": [
12        "logs:CreateLogStream",
13        "logs:PutLogEvents"
14      ],
15      "Resource": [
16        "arn:aws:logs:us-east-1:717571058165:log-group:/aws/lambda/db09myLambdaS3handler:*"
17      ]
18    }
19  ]
20 }

```

Edit statement Remove

Add actions

Choose a service

Filter services

Included

CloudWatch Logs

Available

AMP

API Gateway

API Gateway V2

ASC

Access Analyzer

20. Add new statement for additional permissions. You can use visual editor as well. Create permission for S3 and action is getObject and add the resource by copying the ARN of the created bucket. The permission will be added to the policy for all the objects of the bucket.

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "logs:CreateLogGroup",
7       "Resource": "arn:aws:logs:us-east-1:717571058165:*"
8     },
9     {
10      "Effect": "Allow",
11      "Action": [
12        "logs:CreateLogStream",
13        "logs:PutLogEvents"
14      ],
15      "Resource": [
16        "arn:aws:logs:us-east-1:717571058165:log-group:/aws/lambda/db09myLambdaS3handler:*"
17      ]
18    }
19  ]
20 }

```

```
18 },
19 {
20   "Sid": "Statement1",
21   "Effect": "Allow",
22   "Action": [ ],
23   "Resource": [ ]
24 }
25 ]
26 }
```

+ Add new statement

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

+ Add new statement

Edit statement

Statement1 Remove

Add actions

Choose a service

Q S3 X

Available

- S3
- S3 Express
- S3 Object Lambda
- S3 Outposts

Edit statement

Statement1 Remove

Add actions

All services > S3

Q getObj X

Access level - list

Access level - read

- ☒ GetObject Info
- ☐ GetObjectAd Info

Add a resource


Add

Add a condition (optional)

Add

5652 of 6144 characters remaining

► **Account snapshot**

Storage lens provides visibility into storage usage and activity trends. [Learn more](#) 

`"logs:PutLogEvents"`

Add resource ✕

Specify the resource type and ARN to add for the selected service.

Service

S3 ▼

Resource type

object ▼

Resource ARN

`arn:aws:s3:::db09mybucket`

Cancel **Add resource**

y: 0 ⊗ Errors: 0 ⚠ Warnings: 0 💡 Suggestions: 1

```
{
  "Sid": "Statement1",
  "Effect": "Allow",
  "Action": [
    "s3:GetObject"
  ],
  "Resource": [
    "arn:aws:s3:::db09mybucket"
  ]
},
```

21. Create another bucket for the converted files and add the permission `putObject()` with ARN of the new bucket created.

Account snapshot View Storage Lens dashboard

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

General purpose buckets | Directory buckets

✔ Bucket ARN copied ↻ Copy ARN Empty Delete Create bucket

General purpose buckets (2) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

| Name | AWS Region | Access | Creation date |
|--|---------------------------------|-------------------------------|--|
| <input type="radio"/> db09mybucket | US East (N. Virginia) us-east-1 | Bucket and objects not public | December 7, 2023, 21:44:05 (UTC+05:00) |
| <input checked="" type="radio"/> db09mybucketpdf | US East (N. Virginia) us-east-1 | Bucket and objects not public | December 7, 2023, 22:13:35 (UTC+05:00) |

Policy editor Visual JSON Actions 📄

```

10     "Effect": "Allow",
11     "Action": [
12         "logs:CreateLogStream",
13         "logs:PutLogEvents"
14     ],
15     "Resource": [
16         "arn:aws:logs:us-east-1:717571058165:log-group:/aws/lambda/db09myLambdaS3handler:*"
17     ]
18 },
19 {
20     "Sid": "Statement1",
21     "Effect": "Allow",
22     "Action": [
23         "s3:GetObject"
24     ],
25     "Resource": [
26         "arn:aws:s3:::db09mybucket"
27     ]
28 },
29 {
30     "Sid": "Statement2",
31     "Effect": "Allow",
32     "Action": [
33         "s3:PutObject"
34     ],
35     "Resource": [
36     ]
37 }
38 ]

```

Edit statement Remove

Statement2

Add actions

All services > S3

Q puto X

Access level - list

Access level - read

Access level - read or write

☒ PutObject [Info](#)

☐ PutObjectLegalHold [Info](#)

☐ PutObjectRetention [Info](#)

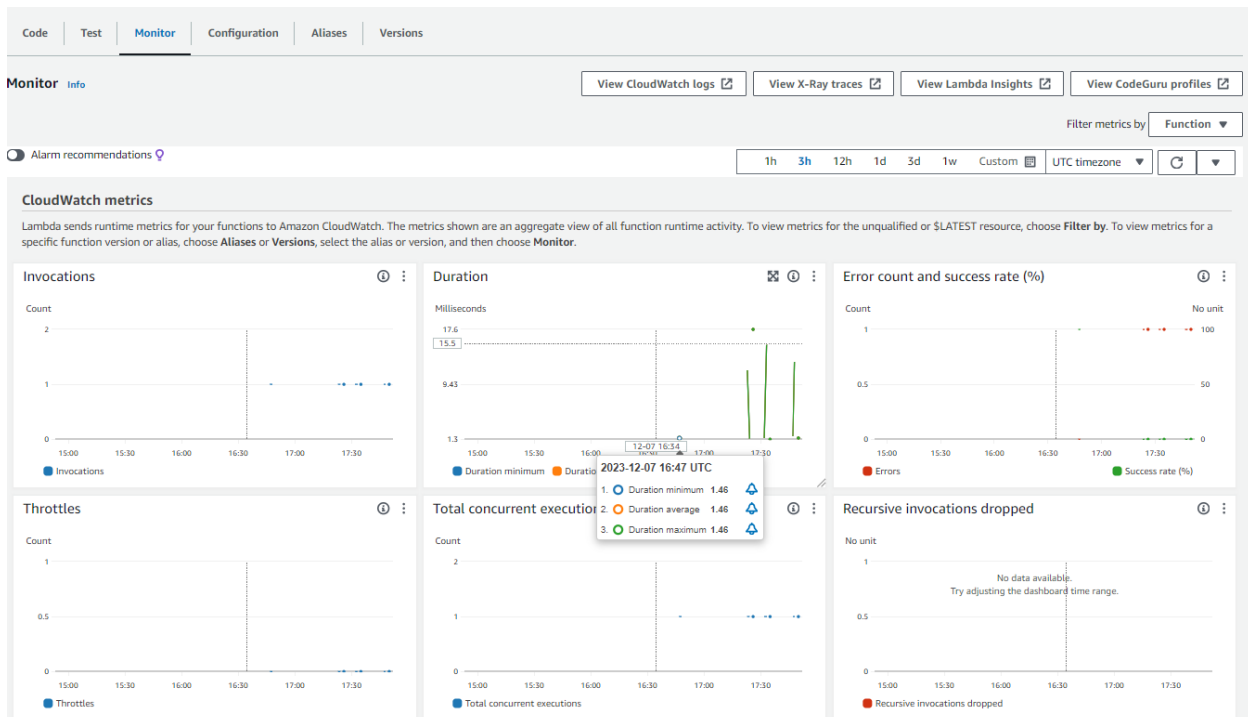
Access level - permissions

☐ PutObjectAcl [Info](#)

Add a resource Add

22. Save and review the policies.

23. Upload the files in bucket and see the logs of cloud watch for lambda triggers and *other*.



24. Delete the lambda function.

The screenshot shows the 'Delete 1 functions' dialog box in the AWS Lambda console. The dialog box has a title bar with a close button (X). Below the title bar, there is a warning message: 'Deleting a function permanently removes the function code. The related logs, roles, test event schemas, and triggers are retained in your account.' Below the warning message, the function name 'db09myLambdaS3handler' is displayed. Below the function name, there is a confirmation field with the text 'delete'. At the bottom right of the dialog box, there are two buttons: 'Cancel' and 'Delete'.

<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>