

Code source [Info](#)

File Edit Find View Go Tools Window

Test

Deploy

Changes deployed



Go to Anything (Ctrl-P)

Environment

CourseLambda - /
index.js

index.js

```
1 const AWS = require ("aws-sdk");
2 var dynamodb = new AWS.DynamoDB();
3 const sns = new AWS.SNS();
4 const s3 = new AWS.S3()
5 // var dynamodbDoc = new AWS.DynamoDB().DocumentClient;
6 const courseTable = process.env.COURSE_TABLE
7
8
9 exports.handler = async(event, context) => {
10   console.log("Event is " +JSON.stringify(event))
11
12   const Text_Event = JSON.stringify(event, null, 2)
13   console.log("Text_Event "+Text_Event)
14
15   const params={
16     "Message": Text_Event,
17     "Subject": "Test SNS from Lambda",
18     "TopicArn": "arn:aws:sns:us-east-1:317586427235:EmailList"
19   };
20   sns.publish(params, context.done);
21
22   // var s3params = {
23   //   Bucket : "cs516assignment5",
24   //   Key : "1",
25   //   Body : JSON.stringify(event)
26   // }
27   // await s3.putObject(s3params).promise()
28
29
```



Execution results x

Execution results

Status: Succeeded Max memory used

Test Event Name

MyEvent

Response

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

Function Logs

```
t":{"accountId":"123456789012","resourceId":"123456","stage":"prod","requestId":"2021-11-12T04:30:48.597Z 55bce115-699b-4ebe-a9d5-ff399cbcb50c INFO
"body": "eyJ0ZXN0IjoiYm9keS39",
"resource": "/{proxy+}",
"path": "/path/to/resource",
"httpMethod": "POST",
"isBase64Encoded": true,
"queryStringParameters": {
  "foo": "bar"
},
"multiValueQueryStringParameters": {
  "foo": [
    "bar"
  ]
},
"pathParameters": {
  "proxy": "/path/to/resource"
},
}
```



SQS

+ Add trigger

+ Add destination

Function ARN

arn:aws:lambda:us-east-1:317586427235:function:MintesLambda

Code

Test

Monitor

Configuration

Aliases

Versions

Code source [Info](#)

Upload from

File Edit Find View Go Tools Window

Test

Deploy

Changes deployed



Go to Anything (Ctrl-P)

Environment

MintesLambda - /

index.js

```
index.js
10 const saveParams = {
11   TableName: mintesTable,
12   Item: {
13     "id": {S: "1"},
14     "name": {S: "Mintes Geb"},
15     "profession": {SS: ["Architect", "Industrial-Designer", "programmer"]},
16     "date": {S: Math.floor(new Date().getTime()/1000) + ""}
17   }
18 };
19
20
21 await dynamodb.putItem(saveParams).promise();
22
23 console.log("TRIGGERED 9")
24 // console.log("myContext " + JSON.stringify(context))
25 // console.log("myEvent " + JSON.stringify(event))
26 console.log("myData " + (event.Records[0].body))
27 console.log("myData " + JSON.stringify(event.Records[0].body.Type))
28
29 var s3params = {
30   Bucket : "cs516assignment5",
31   Key : "myData",
32   Body : event.Records[0].body
33 }
34 await s3.putObject(s3params).promise()
35
36
37
38
```

Execution results x

Execution results

No execution results yet

- 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
- AWS Marketplace for S3

Amazon S3 > cs516assignment5 > myData

myData Info

Copy S3 URI Download Open

- Properties
- Permissions
- Versions

Object overview

Owner
awslabsc0w3024764t1635117566

AWS Region
US East (N. Virginia) us-east-1

Last modified
November 11, 2021, 23:36:45 (UTC-05:00)

Size
2.2 KB

Type

Key
myData

S3 URI
s3://cs516assignment5/myData

Amazon Resource Name (ARN)
arn:aws:s3:::cs516assignment5/myData

Entity tag (Etag)
63a49ebbd23d557b1c1500b7ae401a5f

Object URL
https://cs516assignment5.s3.amazonaws.com/myData

Object management overview

The following bucket properties and object management configurations impact the behavior of this object.

Code Test Monitor Configuration Aliases Versions

Code source Info

Upload from

File Edit Find View Go Tools Window

Test

Deploy

Changes deployed

Go to Anything (Ctrl-P)

Environment
CourseLambda - /
index.js

```
5
6
7 exports.handler = async(event, context) => {
8   console.log("Event is " + JSON.stringify(event))
9
10  const Text_Event = JSON.stringify(event, null, 2)
11  console.log("Text_Event " + Text_Event)
12  const sns = new AWS.SNS();
13  const params={
14    "Message": Text_Event,
15    "Subject": "Test SNS from Lambda",
16    "TopicArn": "arn:aws:sns:us-east-1:317586427235:mysnstopic"
17  };
18  sns.publish(params, context.done);
19
20  const saveParams = {
21    TableName: courseTable,
22    Item: {
23      "courseCode": {
24        S: "CS517"
25      },
26      "courseName": {
27        S: "test"
28      },
29      "teacherName": {
30        S: "Mintes"
31      },
32      "students": {
33        SS: [
34          "Bipin",
35          "Ryans",
36          "Michael"
37        ]
38      },
39      "monthYear": {
40        S: "Nov, 2021"
41      }
42    }
43  }
```

Execution result

Execution results Status: Succeeded Max memory used: 79 MB Time: 934.34 ms

Test Event Name

MyEvent

Response

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

Function Logs

```
9012", "resourceId": "123456", "stage": "prod", "requestId": "c6af9ac6-7b61-11e6-9a41-93e8deadbeef",
2021-11-12T02:30:15.159Z 9520d17e-e4bf-481f-a894-e290b7329eea INFO Text_Event {
  "body": "eyJ0ZXN0IjoieiYm9keSJ9",
  "resource": "/{proxy+}",
  "path": "/path/to/resource",
  "httpMethod": "POST",
  "isBase64Encoded": true,
  "queryStringParameters": {
    "foo": "bar"
  },
  "multiValueQueryStringParameters": {
    "foo": [
      "bar"
    ]
  },
  "pathParameters": {
    "proxy": "/path/to/resource"
  },
  "stageVariables": {
    "baz": "qux"
  },
  "headers": {
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",
    "Accept-Encoding": "gzip, deflate, sdch",
    "Accept-Language": "en-US,en;q=0.8",
```