

Padlet S10 Ejercicio propuesto 2

Create a course folder and a student folder. (Add file archivo01.txt)

Amazon S3 > Buckets > luisfmendezl-sec1 > universidades/ > UTEC/ > CS2032/ > 202110216/

202110216/

Objects | Properties

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 1

Copy S3 URI Copy URL Download Open Delete Actions ▼

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	archivo01.txt	txt	June 4, 2023, 16:20:09 (UTC-05:00)

Modify LeerMetadataArchivo lambda function

Code source Info

File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (Ctrl-P)

Environment

- LeerMetadataArchiv
- lambda_function.py

```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # Entrada (json)
6     archivo_id = event['Records'][0]['s3']['object']['key']
7     key = archivo_id.split('/') # ["universidades", "UTEC", "CS2032", "202110216", "archivo01.txt"]
8     tenant_id = key[1]
9     curso_id = key[2]
10    alumno_id = key[3]
11    archivo_last_modified = event['Records'][0]['eventTime']
12    archivo_size = event['Records'][0]['s3']['object']['size']
13    bucket_name = event['Records'][0]['s3']['bucket']['name']
14    archivo = {
15        'tenant_id': tenant_id,
16        'curso_id': curso_id,
17        'alumno_id': alumno_id,
18        'archivo_id': archivo_id,
19        'archivo_datos': {
20            'last_modified': archivo_last_modified,
21            'size': archivo_size,
22            'bucket_name': bucket_name
23        }
24    }
25    # Publicar en SNS
26    sns_client = boto3.client('sns')
27    response_sns = sns_client.publish(
28        TopicArn = 'arn:aws:sns:us-east-1:896823805584:NuevoArchivo',
29        Subject = 'Nuevo Archivo',
30        Message = json.dumps(archivo),
31        MessageAttributes = {
32            'tenant_id': {'DataType': 'String', 'StringValue': tenant_id }
33        }
34    )
35    # TODO implement
36    return {
37        'statusCode': 200,
```

Modify CrearArchivo_UTEC lambda function

Code sourceInfo

FileEditFindViewGoToolsWindowTestDeploy

Go to Anything (Ctrl-P)

Environment

CrearArchivo_UTEC

lambda_function.py

lambda_function

```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # Entrada (json)
6     print(event) # Revisar en CloudWatch
7     archivo_json = json.loads(event['Records'][0]['Sns']['Message'])
8     # Proceso
9     dynamodb = boto3.resource('dynamodb')
10    table = dynamodb.Table('t_archivos_UTEC')
11    archivo = {
12        'tenant_id': archivo_json['tenant_id'],
13        'curso_id': archivo_json['curso_id'],
14        'alumno_id': archivo_json['alumno_id'],
15        'archivo_id': archivo_json['archivo_id'],
16        'archivo_datos': archivo_json['archivo_datos']
17    }
18    print(archivo) # Revisar en CloudWatch
19    response = table.put_item(Item=archivo)
20    # Salida (json)
21    return {
22        'statusCode': 200,
23        'response': response
24    }
```

Verification

AWS Notifications <no-reply@sns.amazonaws.com>

para mí ▾

inglés ▾ > español ▾ Traducir mensaje

{'tenant_id': 'UTEC', 'curso_id': 'CS2032', 'alumno_id': '202110216', 'archivo_id': 'universidades/UTEC/CS2032/202110216/archivo01.txt', '04T21:17:31.264Z', 'size': 8, 'bucket_name': 'luisfmendezl-sec1'}}

...

← Responder

→ Reenviar

DynamoDB > Items > t_archivos_UTEC

> t_archivos_UTEC

Autopreview

View table details

► Scan or query items

Expand to query or scan items.

✔ Completed. Read capacity units consumed: 2

Items returned (1/4)

Actions ▾

Create item

< 1 > ⚙️ 🔍

	tenant_id	archivo_id	alumn...	archivo_datos
✔	UTEC	universidades/UTEC/CS2032/202110216/archivo01.txt	202110216	{ "size": { "N": "8" }, "last_modified": { "S": "