Transforming Your Data with Amazon S3 Object Lambda

- Amazon Simple Storage Service (S3) allows you to store and share data to be used by multiple applications. However, certain applications require specific formats or views of your data, In order to accommodate the various applications accessing your data, you could either store multiple copies of your data in different formats or provide an additional layer to the object retrieval process that transforms your data to match the required format

- With Amazon S3 Object Lambda, you can now add your own custom AWS Lambda function code to intercept data being retrieved from S3 before returning it to an application. This Lambda function can transform your data to meet your application's requirements without modifying the original object stored in your S3 bucket

- In this hands-on , I will configure an Amazon S3 Object Lamdba Access Point and write a custom Lambda function to retrieve CSV files stored in an S3 bucket and transform that data into JSON before returning it to the application

Hands-on Steps:

1- Configuring an Amazon S3 Object Lambda Access Point
2- Writing a Lambda Function for an S3 Object Lambda Access Point
3- Retrieving Transformed Data from an S3 Object Lambda Access Point

1- Configuring an Amazon S3 Object Lambda Access Point

- In this lab step, you will create an Amazon S3 Access Point and Object Lambda Access Point which will allow you to retrieve and transform data from an S3 bucket

- The data that you will be working with comes in the form of comma-separated value (CSV) files. The data below is from the dev-team.csv file that is stored in my S3 bucket

1- Click the bucket, You will notice the single CSV file in this S3 bucket mentioned previously
2- Click the Access Points tab

(Amazon S3 Access Points are network endpoints that are attached to buckets that you can use to perform S3 object operations. In this case, your S3 Object Lambda

access point will use this access point to retrieve your team data before transforming it)

3- Click Create access point
4- Configure the following properties for your access point
5- Access point name: Enter lab-ap, Network origin: Select Internet,Block Public Access settings for this Access Point
6- Click Create access point
7- Click Object Lambda Access Points
8- Configure the following properties for your Object Lambda access point:

- Object Lambda Access Point name: Enter lab-olap
- Supporting Access Point: Click Browse In the resulting Choose supporting Access Point popup, select lab-ap, then click Choose supporting Access Point
- S3 APIs: GetObject
- Invoke Lambda function: Ensure Choose from functions in your account is selected
- Lambda function: Select the cloudacademylabs-CsvToJsonConverter-XXXX function

10. Scroll to the bottom of the page and click Create Object Lambda Access Point

11. Click the lab-olap access point name

12. Click the Properties tab

13. From the Object Lambda Access Point overview section, copy the Amazon Resource Name (ARN) value onto your clipboard

With your access points created, you will next write the accompanying Lambda function code needed to transform and return your S3 object data to your users

2- Writing a Lambda Function for an S3 Object Lambda Access Point:

1. In the AWS Management Console, in the search bar at the top, enter Lambda, and under Services, click the Lambda result
2. Click the cloudacademylabs-CsvToJsonConverter-XXXX function name from the list:
3. Replace the code in index.py with the following code:

```
import boto3
import urllib3
import csv
import json

def lambda_handler(event, context):

    object_get_context = event["getObjectContext"]
    request_route = object_get_context["outputRoute"]
    request_token = object_get_context["outputToken"]
    s3_url = object_get_context["inputS3Url"]

    # Get object from S3
    http = urllib3.PoolManager()
    response = http.request('GET', s3_url)
    original_object = response.data.decode('utf-8').splitlines()
    csv_reader = csv.DictReader(original_object)
```

```
data = {}

for rows in csv_reader:
    id = rows['Identifier']
    data[id] = rows

json_object = json.dumps(data, indent=4)

# Write object back to S3 Object Lambda
s3 = boto3.client('s3')
s3.write_get_object_response(
    Body=json_object,
    RequestRoute=request_route,
    RequestToken=request_token)

return {'status_code': 200}
```

3- Retrieving Transformed Data from an S3 Object Lambda Access Point

In this lab step, you will test your newly created S3 Object Lambda Access Point and accompanying CSV converter function to retrieve and transform data from your S3 bucket. You will utilize a Lambda function preconfigured for this S3 GetObject operation test

1. In the AWS Lambda console, click the Functions link
2. Click the cloudacademylabs-RetrieveTeamData-XXXX function name
3. Review the existing index.py file defined in the Code source section

```
import boto3
s3 = boto3.client('s3')

def lambda_handler(event, context):

    response = s3.get_object(
      Bucket='{REPLACE-WITH-S3-OBJECT-LAMBDA-ACCESS-POINT}',
      Key= 'dev-team.csv'
    )

    return response['Body'].read()
```

The function makes a simple S3 API get_object call, but instead of an S3 bucket name passed into the Bucket parameter, you will pass in the S3 Object Lambda Access Point ARN. This will retrieve the S3 object using this newly created access point, transform the dev-team.csv file into a JSON object, and return it to this Lambda function

4. Replace the string {REPLACE-WITH-S3-OBJECT-LAMBDA-ACCESS-POINT} with the Object Lambda Access ARN
5. Click Deploy above the code editor then test
6. In the resulting Execution results tab, the function will return in json