

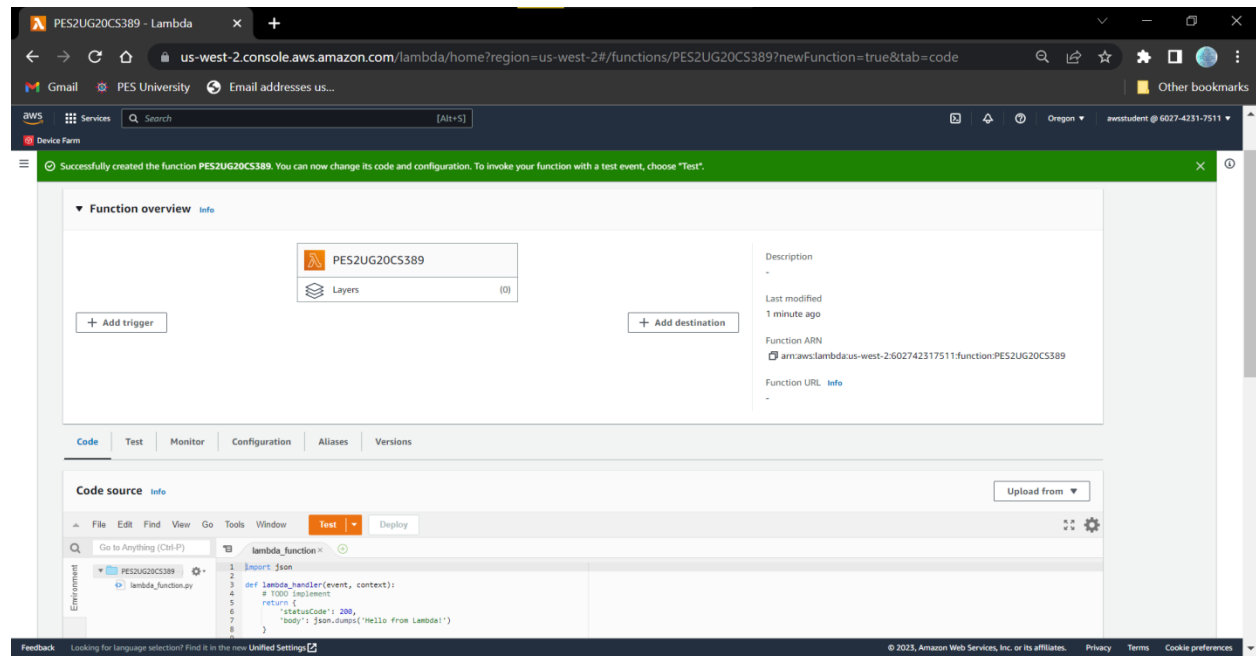
Cloud Computing: UE20CS352

Assignment 1 - Introduction to AWS Lambda with API Gateway

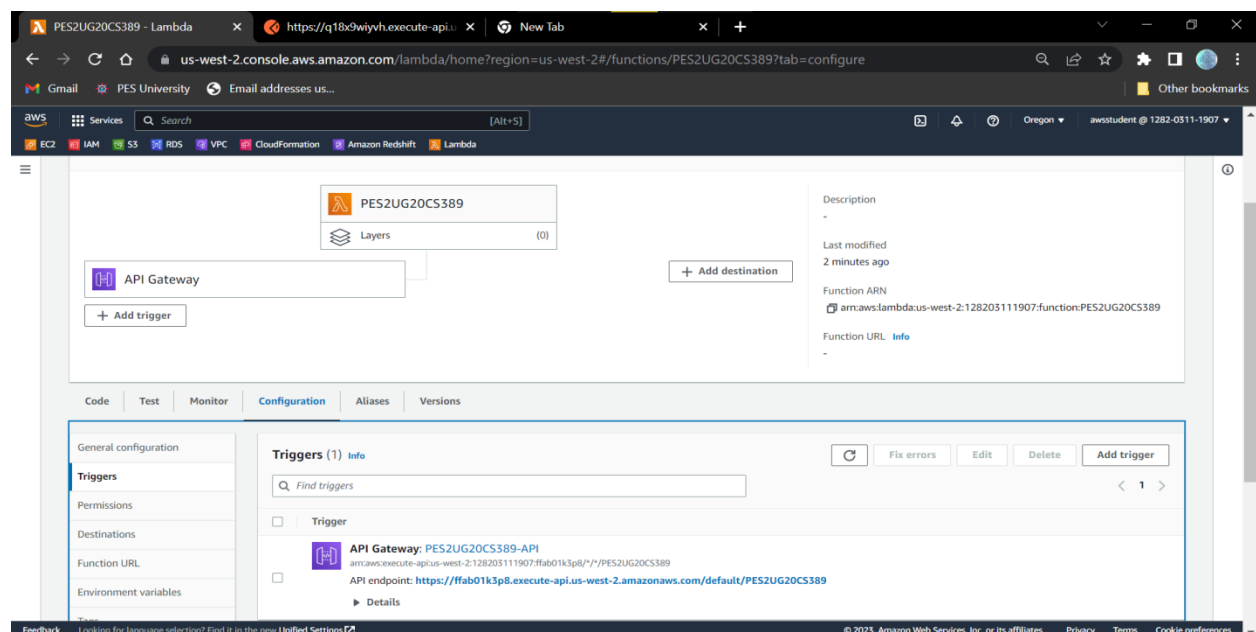
-Vishwa Mehul Mehta, Section F

-PES2UG20CS389

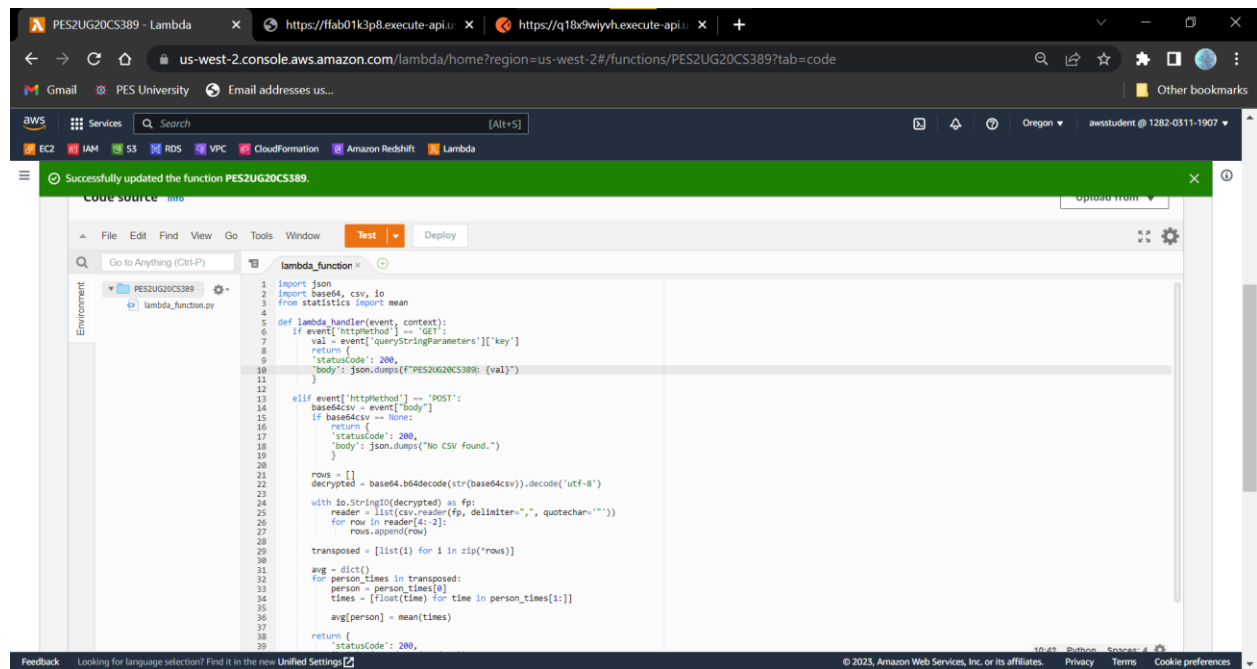
1a: Lambda Function Created



1b: API Gateway created, trigger added



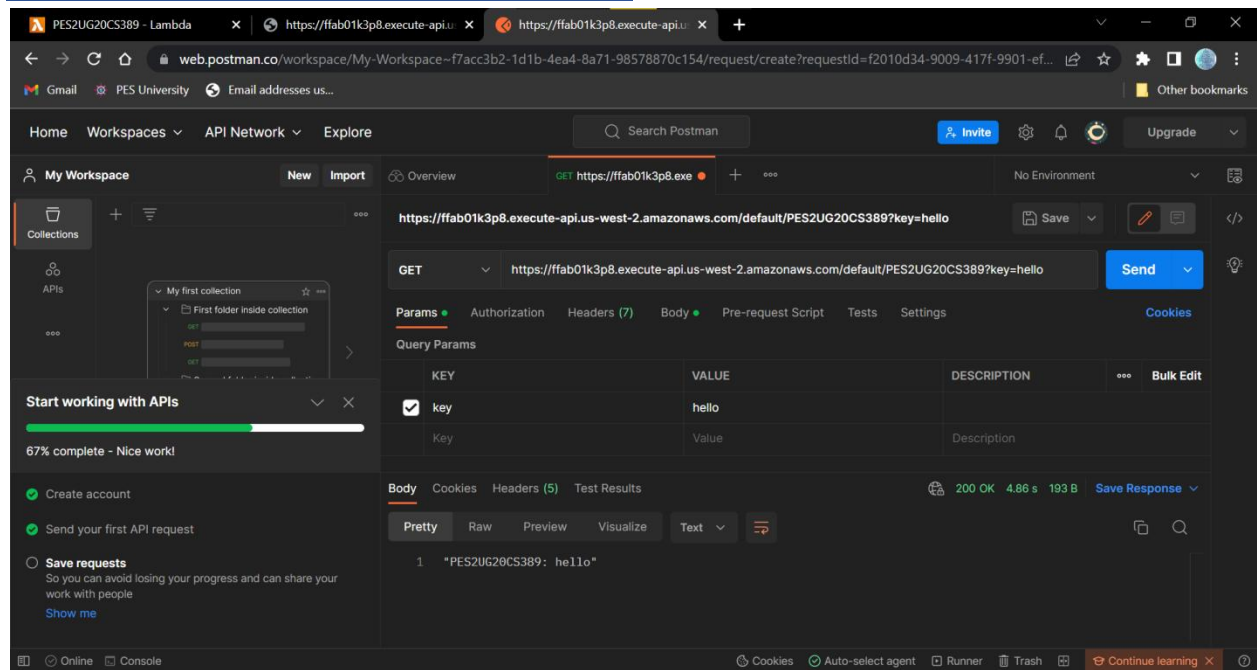
1c: Modified Lambda function



The screenshot shows the AWS Lambda console for the function PES2UG20CS389. A green banner at the top indicates "Successfully updated the function PES2UG20CS389." The code source is displayed in a text editor. The code is a Python lambda function that handles GET and POST requests. For GET requests, it returns a 200 status code and a JSON body containing the function name. For POST requests, it decodes the base64 body, reads a CSV file, and returns a 200 status code with a JSON body containing the CSV data.

```
1 import json
2 import base64, csv, io
3 from statistics import mean
4
5 def lambda_handler(event, context):
6     if event['httpMethod'] == 'GET':
7         val = event['queryStringParameters']['key']
8         return {
9             'statusCode': 200,
10            'body': json.dumps(f"PES2UG20CS389: {val}")
11        }
12
13     elif event['httpMethod'] == 'POST':
14         base64csv = event['body']
15         if base64csv == None:
16             return {
17                 'statusCode': 200,
18                 'body': json.dumps("No CSV found.")
19             }
20
21         rows = []
22         decrypted = base64.b64decode(str(base64csv)).decode('utf-8')
23
24         with io.StringIO(decrypted) as fp:
25             reader = list(csv.reader(fp, delimiter=";", quotechar='"'))
26             for row in reader[4:2]:
27                 rows.append(row)
28
29         transposed = [list(i) for i in zip(*rows)]
30
31         avg = dict()
32         for person_times in transposed:
33             person = person_times[0]
34             times = [float(time) for time in person_times[1:]]
35             avg[person] = mean(times)
36
37         return {
38             'statusCode': 200,
39             'body': json.dumps(avg)
```

1d: Successful Postman GET request



The screenshot shows the Postman interface with a GET request to the AWS Lambda function. The request URL is `https://fab01k3p8.execute-api.us-west-2.amazonaws.com/default/PES2UG20CS389?key=hello`. The response is a 200 OK status with a body of `"PES2UG20CS389: hello"`.

Params

KEY	VALUE	DESCRIPTION
key	hello	
Key	Value	Description

Body

```
1 "PES2UG20CS389: hello"
```

1e: Successful Postman POST request

The screenshot shows the Postman interface with a POST request configured. The URL is `https://ffab01k3p8.execute-api.us-west-2.amazonaws.com/default/PES2UG20CS389?key=hello`. The request body is a JSON object: `{ "Inno": 12.93258, "Vation": 18.99218, "Lab": 16.91436, "Hashco": 16.89154, "Deleven": 11.088 }`. The response status is 200 OK, and the response body is empty. The interface includes a sidebar with collections, a top navigation bar, and a bottom status bar.

POST `https://ffab01k3p8.execute-api.us-west-2.amazonaws.com/default/PES2UG20CS389?key=hello`

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
key	BvxfxrOsF/times.csv	
Key	Value	Description

Body Cookies Headers (5) Test Results

200 OK 9.54 s 272 B Save Response

Pretty Raw Preview Visualize Text

```
1 { "Inno": 12.93258, "Vation": 18.99218, "Lab": 16.91436, "Hashco": 16.89154, "Deleven": 11.088 }
```

1f: Failed Postman PUT/DELETE/HEAD request

The screenshot shows the Postman interface with a PUT request configured. The URL is `https://ffab01k3p8.execute-api.us-west-2.amazonaws.com/default/PES2UG20CS389?key=hello`. The request body is a JSON object: `{ "Inno": 12.93258, "Vation": 18.99218, "Lab": 16.91436, "Hashco": 16.89154, "Deleven": 11.088 }`. The response status is 200 OK, and the response body is `"Only GET and POST are supported"`. The interface includes a sidebar with collections, a top navigation bar, and a bottom status bar.

PUT `https://ffab01k3p8.execute-api.us-west-2.amazonaws.com/default/PES2UG20CS389?key=hello`

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
key	BvxfxrOsF/times.csv	
Key	Value	Description

Body Cookies Headers (5) Test Results

200 OK 5.69 s 209 B Save Response

Pretty Raw Preview Visualize Text

```
1 "Only GET and POST are supported"
```