 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030

Aim :- Developing REST APIs with API Gateway.

Lab overview and objectives

In this lab, you will create a REST application programming interface (API) by using Amazon API Gateway.

After completing this lab, you should be able to:

- Create simple mock endpoints for REST APIs and use them in your website.
- Enable Cross-Origin Resource Sharing (CORS)

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Scenario

In the *previous lab*, you took on the role of Sofia to build a web application for the café. As part of this process, you created an *Amazon DynamoDB table* that was named *FoodProducts*, where you stored information about café menu items.

You then loaded data that was formatted in JavaScript Object Notation (JSON) into the database table. The table structure looked similar to the following table (one line item of table data is shown as an example):

product_name	description	price_in_cents	product_id	tags	special
apple pie slice	A delicious slice of Frank's homemade pie.	595	a444	[{ "S" : "pie slice" }, { "S" : "on offer" }]	1


In the *previous lab* you also configured code that used the AWS SDK for Python (Boto3) to:

- Scan a DynamoDB table to retrieve product details.
- Return a single item by product name using get-item as a proof of concept
- Create a Global Secondary Index (GSI) called **special_GSI** that you could use to filter out menu items that are on offer and not out of stock.

In *this lab*, you will continue to play the role of Sofia. You will use Amazon API Gateway to configure **mock data endpoints**. There are three that you will create:

- **[GET]** /products (which will *eventually* invoke a DynamoDB table scan)
- **[GET]** /products/on_offer (which will *eventually* invoke a DynamoDB index scan and filter)
- **[POST]** /create_report (which will *eventually* trigger a batch process that will send out a report)

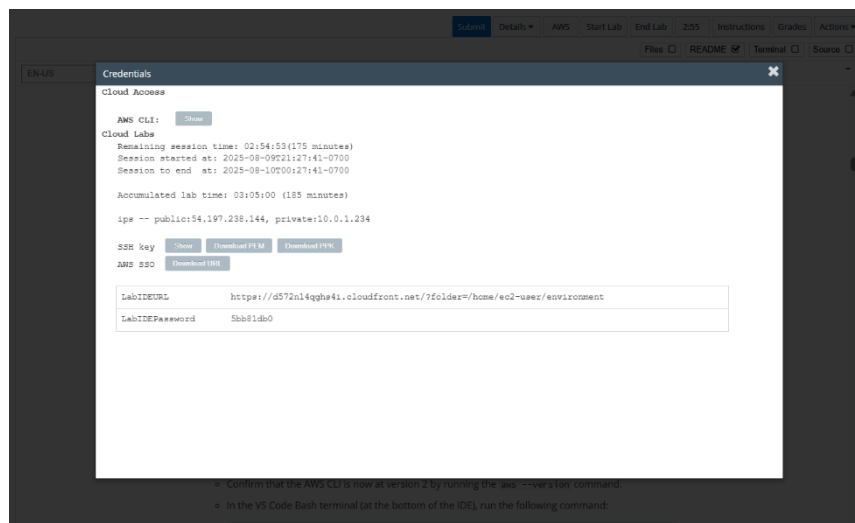
Then in the lab that follows this one, you will replace the mock endpoints with real endpoints, so that the web application can connect to the DynamoDB backend.

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030


Task 1: Preparing the lab

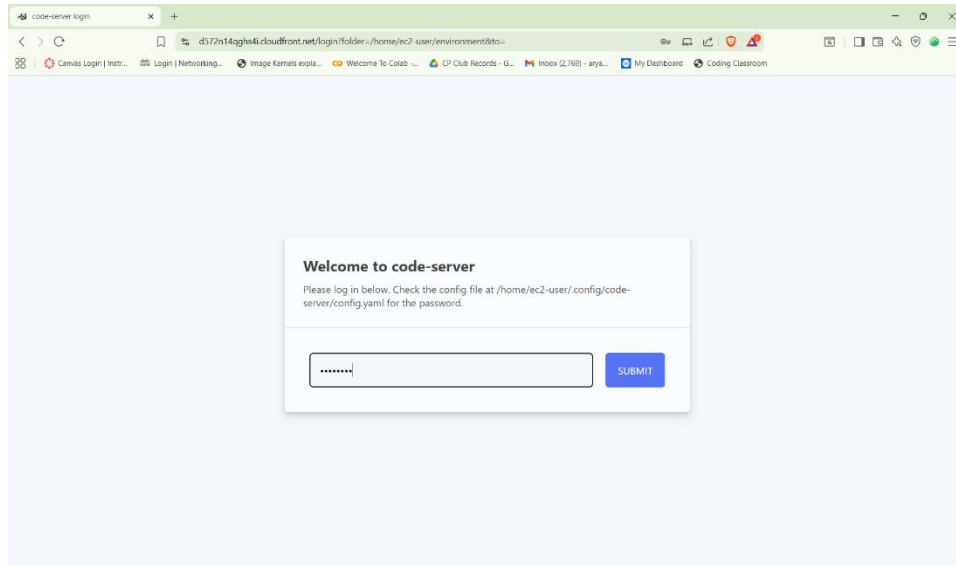
Connect to the VS Code IDE.

- At the top of these instructions, choose **Details** followed by **AWS: Show**
- Copy values from the table **similar** to the following and paste it into an editor of your choice for use later.
 - LabIDEURL**
 - LabIDEPassword**



- In a new browser tab, paste the value for **LabIDEURL** to open the VS Code IDE.
- On the prompt window **Welcome to code-server**, enter the value for **LabIDEPassword** you copied to the editor earlier, choose **Submit** to open the VS Code IDE.

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030



5. Download and extract the files that you need for this lab.
 - In the VS Code bash terminal (located at the bottom of the IDE), run the following commands:

```
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCDEV-2-91558/04-lab-api/code.zip -P /home/ec2-user/environment
```

```
[ec2-user@ip-10-0-1-150 environment]$ wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCDEV-2-91558/04-lab-api/code.zip -P /home/ec2-user/environment
--2025-08-10 13:53:43-- https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCDEV-2-91558/04-lab-api/code.zip
Resolving aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)... 52.92.212.146, 3.5.81.118, 3.5.82.245, ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)|52.92.212.146|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5828018 (5.6M) [application/zip]
Saving to: '/home/ec2-user/environment/code.zip'


code.zip                               100%[=====>] 5.56M 8.15MB/s in 0.7s

2025-08-10 13:53:44 (8.15 MB/s) - '/home/ec2-user/environment/code.zip' saved [5828018/5828018]

[ec2-user@ip-10-0-1-150 environment]$
```

6. You should see that the **code.zip** file was downloaded to the VS Code IDE and is now in the left navigation pane.
 - Extract the file by running the following command:

```
unzip code.zip
```


 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030

```
[ec2-user@ip-10-0-1-150 environment]$ unzip code.zip
Archive:  code.zip
  extracting: python_3/create_on_offer_api.py
  extracting: python_3/create_products_api.py
  extracting: python_3/create_report_api.py
  extracting: python_3/update_config.py
  extracting: resources/setup.sh
  extracting: resources/website_security_policy.json
  extracting: resources/permissions.py
  extracting: resources/website/callback.html
  extracting: resources/website/all_products.json
  extracting: resources/website/all_products_on_offer.json
  extracting: resources/website/config.js
  extracting: resources/website/beans.json
  extracting: resources/website/spint.md
  extracting: resources/website/index.html
  extracting: resources/website/favicon.ico
  extracting: resources/website/scripts/main.js
  extracting: resources/website/scripts/pastries.js
  extracting: resources/website/scripts/coffee.js
  extracting: resources/website/scripts/navigation.js
  extracting: resources/website/scripts/jquery-3.6.0.min.js
  extracting: resources/website/styles/beans.css
  extracting: resources/website/styles/main.css
  extracting: resources/website/styles/pastries.css
  extracting: resources/website/styles/reset.css
  extracting: resources/website/styles/navigation.css
  extracting: resources/website/styles/coffee.css
  extracting: resources/website/images/main_coffee_shop.png
  extracting: resources/website/images/main_coffee_shop_1.png
  extracting: resources/website/images/logo.png
  extracting: resources/website/images/not_expanded.png
  extracting: resources/website/images/expanded.png
  extracting: resources/website/images/items/garlic_bagel.jpeg
  extracting: resources/website/images/items/chocolate_cupcake.png
  extracting: resources/website/images/items/chocolate_cake.png
  extracting: resources/website/images/items/chocolate_cake_slice.jpeg
  extracting: resources/website/images/items/chocolate_chip_cupcake.jpeg
  extracting: resources/website/images/items/strawberry_cupcake.jpeg
```

7. Run a script that upgrades the version of the AWS CLI installed on the VS Code IDE.
 - To set permissions on the script and then run it, run the following commands in the Bash terminal:

```
chmod +x ./resources/setup.sh && ./resources/setup.sh
```

The script will prompt you for the **IP address** by which your computer is known to the internet. Use www.whatismyip.com to discover this address and then paste the IPv4 address into the command prompt and finish running the script.

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030

```

[ec2-user@ip-10-0-1-150 environment]$ chmod +x resources/setup.sh && resources/setup.sh
Please enter a valid IP address:
152.58.63.192
IP address:152.58.63.192
upload: resources/website/all_products_on_offer.json to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/all_products_on_offer.json
upload: resources/website/callback.html to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/callback.html
upload: resources/website/all_products.json to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/all_products.json
upload: resources/website/beans.json to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/beans.json
upload: resources/website/images/beans/excelsa.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/beans/excelsa.png
upload: resources/website/config.js to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/config.js
upload: resources/website/images/items/blueberry_bagel.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/blueberry_bagel.png
upload: resources/website/images/items/apple_pie.jpeg to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/apple_pie.jpeg
upload: resources/website/images/items/blueberry_jelly_doughnut.jpeg to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/blueberry_jelly_doughnut.jpeg
upload: resources/website/images/beans/robusta.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/beans/robusta.png
upload: resources/website/images/items/boston_cream_doughnut.jpeg to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/boston_cream_doughnut.jpeg
upload: resources/website/images/expanded.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/expanded.png
upload: resources/website/images/beans/liberica.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/beans/liberica.png
upload: resources/website/images/items/apple_pie_slice.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/apple_pie_slice.png
upload: resources/website/images/items/apple_pie.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/apple_pie.png
upload: resources/website/images/beans/arabica.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/beans/arabica.png
upload: resources/website/images/items/boston_cream_doughnut.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/boston_cream_doughnut.png
upload: resources/website/favicon.ico to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/favicon.ico
upload: resources/website/images/items/apple_pie_slice.jpeg to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/apple_pie_slice.jpeg
upload: resources/website/images/items/cherry_pie.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/cherry_pie.png
upload: resources/website/images/items/blueberry_bagel.jpeg to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/blueberry_bagel.jpeg
upload: resources/website/images/items/blueberry_jelly_doughnut.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/blueberry_jelly_doughnut.png
upload: resources/website/images/items/cherry_pie_slice.png to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/cherry_pie_slice.png
upload: resources/website/images/items/cherry_pie.jpeg to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/cherry_pie.jpeg
upload: resources/website/images/items/cherry_pie_slice.jpeg to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/cherry_pie_slice.jpeg
upload: resources/website/images/items/chocolate_chip_cupcake.jpeg to s3://c168617a434024811142234t1w184333714729-s3bucket-1wvveyyvhv5l/images/items/chocolate_chip_cupcake.jpeg
Layout US

```


8. Verify the AWS CLI version and also verify that the SDK for Python is installed.

- Confirm that the AWS CLI is now at version 2 by running the **aws --version** command.
- In the VS Code Bash terminal (at the bottom of the IDE), run the following command:
pip3 show boto3

```

[ec2-user@ip-10-0-1-234 environment]$ aws --version
aws-cli/2.28.6 Python/3.13.4 Linux/6.1.147-172.266.amzn2023.x86_64 exe/x86_64.amzn.2023
[ec2-user@ip-10-0-1-234 environment]$ pip3 show boto3
Name: boto3
Version: 1.40.6
Summary: The AWS SDK for Python
Home-page: https://github.com/boto/boto3
Author: Amazon Web Services
Author-email:
License: Apache License 2.0
Location: /usr/local/lib/python3.11/site-packages
Requires: botocore, jmespath, s3transfer
Required-by:
[ec2-user@ip-10-0-1-234 environment]$

```

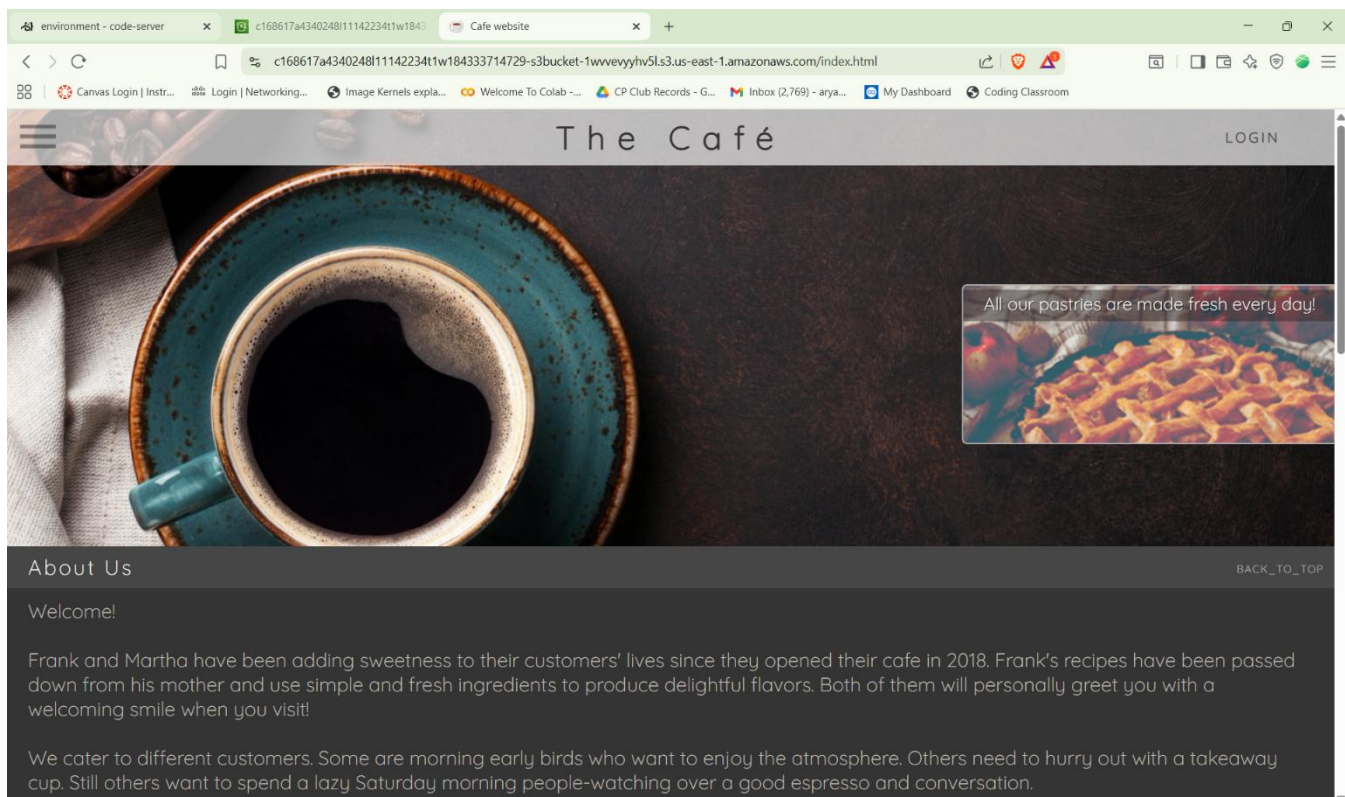

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030


9. Verify that the cafe website can be loaded in a browser tab.

- Load the website in a browser tab.
 - In a browser tab, open the Amazon S3 console.
 - Choose your bucket name, and then choose **Objects**.

If the files that the script just uploaded do not display, choose the refresh icon to view them.

- Choose the **index.html** file.
- Copy the **Object URL**. It will be in the following format. `https://<bucket-name>.s3.amazonaws.com/index.html`
- Verify that the website displays by pasting the full URL into your browser.



 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030

Task 2: Creating the first API endpoint (GET)

10. In the VS Code IDE navigation pane, expand the **python_3** directory and open the file named **create_products_api.py**.

```
python_3 > create_products_api.py
1  import boto3, json
2
3  client = boto3.client('apigateway', region_name='us-east-1')
4
5  response = client.create_rest_api(
6      name='ProductsApi',
7      description='API to get all the food products.',
8      minimumCompressionSize=123,
9      endpointConfiguration={
10         'types': [
11             'REGIONAL',
12         ]
13     }
14 )
15 api_id = response["id"]
```

11. Run the code.

- In the VS Code Bash terminal, go to the directory that contains the Python code file, and run the code.

```
cd python_3
python3 create_products_api.py
```

```
• [ec2-user@ip-10-0-1-150 environment]$ cd python_3
• [ec2-user@ip-10-0-1-150 python_3]$ python3 create_products_api.py
DONE
○ [ec2-user@ip-10-0-1-150 python_3]$
```

12. Return to the **AWS Management Console** browser tab, and open the API Gateway console.
13. Open the **ProductsApi** that you just created by choosing the link.
14. Choose the **GET** method that you defined.



Aim: Developing REST APIs with API Gateway.

Enrolment No: 92200133030

API Gateway

APIs

Custom domain names

Domain name access associations

VPC links

▼ API: ProductsApi

Resources

Stages

Authorizers

Gateway responses

Models

Resource policy

Documentation

Dashboard

API settings

Usage plans

API keys

Client certificates

Settings

Resources

Create resource

/

/products

GET

/products - GET - Method execution

ARNarn:aws:execute-api:us-east-1:184333714729:hir1htenf/*/GET/products

Resource ID1cxzao

Update documentation

Delete

Client

Method request

Integration request

Mock

Method response

Integration response

Mock integration

Method requestIntegration requestIntegration responseMethod responseTest

Method request settings

AuthorizationNONE

API key requiredFalse


Request validatorNone

SDK operation nameGenerated based on method and path

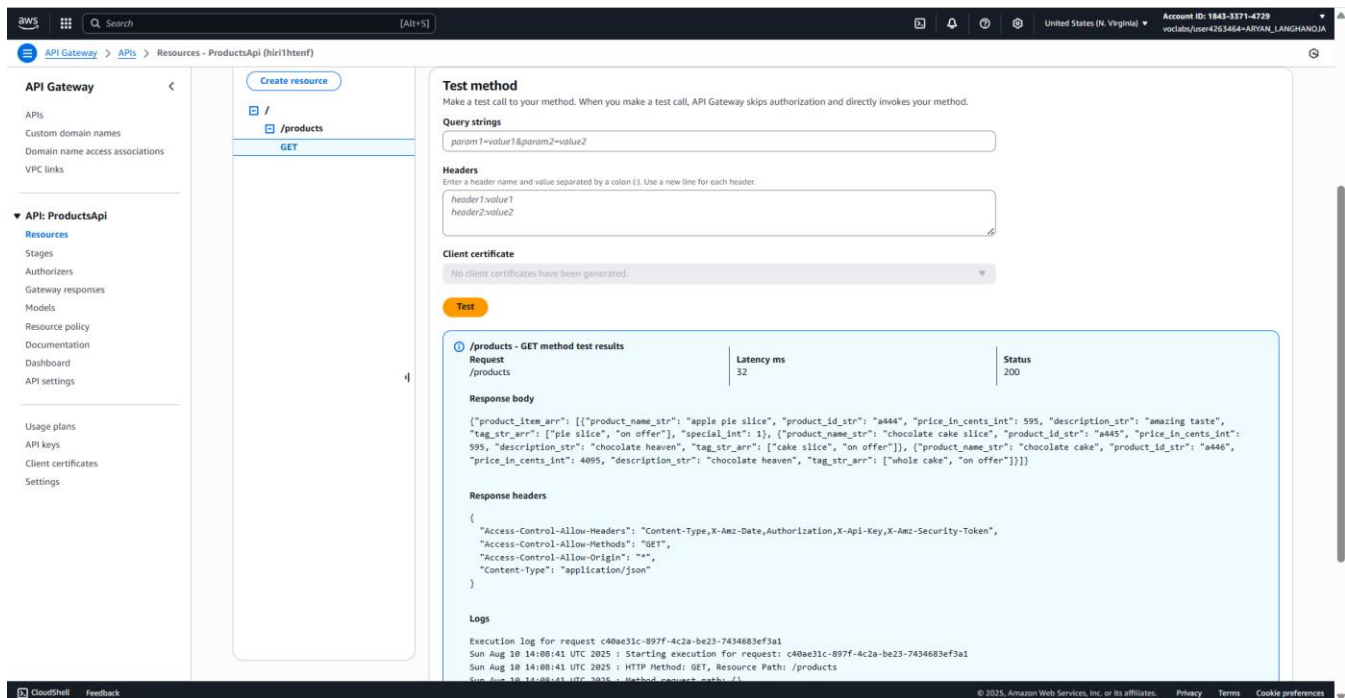
Request paths (0)

NameCaching

No request paths

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030


15. From the API Gateway console **lower panel**, choose the **TEST** link, then scroll to the bottom and choose the **Test** button.



Task 3: Creating the second API endpoint (GET)

16. In the VS Code IDE navigation pane, expand the **python_3** directory and open the file named **create_on_offer_api.py**.
17. Replace **<FMI_1>** and **<FMI_2>** with the correct values so that this code file will add another resource to the API that you defined in the previous task.
 - In a browser tab, go to the **API Gateway** console and choose the **ProductsApi** API that you created a moment ago.
 - In the panel on the left, choose **Resources**.
 - Choose **GET** under products
 - In the breadcrumb navigation at the top of the screen (above the Actions menu), you can see APIs > **ProductsAPI** followed by an id in parenthesis.
 - This is the **api_id**.
 - On the same line, you will see / at the top, you will noticed **Resource ID**.
 - This is the resource **parent_id**
18. Create the API resource.

- Close the file by choosing **X** from the top. (Your changes are saved automatically).
- Then in the Bash terminal, verify that the current directory is **python_3** and run the code.

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030

python3 create_on_offer_api.py

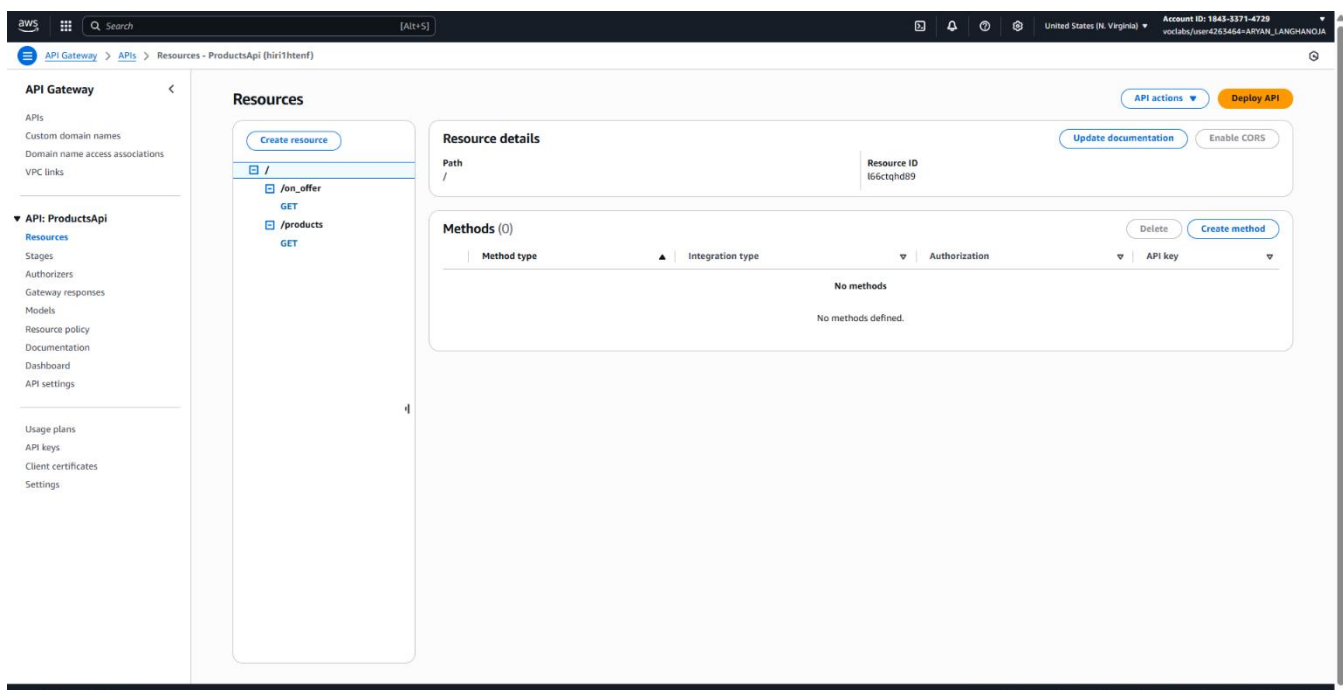
```

[ec2-user@ip-10-0-1-150 python_3]$ python3 create_on_offer_api.py
DONE
[ec2-user@ip-10-0-1-150 python_3]$

```


19. Observe the results.

- Return to the **AWS Management Console** browser tab, and open the **API Gateway** console.
- Choose the **APIs** link in the breadcrumb navigation above, then on the left, open the **ProductsApi** by choosing the link.
- Notice that there is now a nested resource called `/on_offer` under the `/products` resource.



20. Test the `/on_offer` resource.

- From the lower pane **Test**, choose the **Test** button to test the same way you tested the first resource in the previous task.
- You should receive a **200** HTML status code response.

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030

🔍 /on_offer - GET method test results

Request
/on_offer

Latency ms
53

Status
200

Response body

```
{
  "product_item_arr": [
    {
      "product_name_str": "apple pie slice",
      "product_id_str": "a444",
      "price_in_cents_int": 595,
      "description_str": "amazing taste",
      "tag_str_arr": [
        "pie slice",
        "on offer"
      ],
      "special_int": 1
    }
  ]
}
```

Response headers

```
{
  "Access-Control-Allow-Headers": "Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token",
  "Access-Control-Allow-Methods": "GET",
  "Access-Control-Allow-Origin": "*",
  "Content-Type": "application/json"
}
```

Logs

```
Execution log for request 8e69d5d6-a512-4d1c-9647-1ea2fc728f79
Sun Aug 10 14:16:55 UTC 2025 : Starting execution for request: 8e69d5d6-a512-4d1c-9647-1ea2fc728f79
Sun Aug 10 14:16:55 UTC 2025 : HTTP Method: GET, Resource Path: /on_offer
Sun Aug 10 14:16:55 UTC 2025 : Method request path: {}
Sun Aug 10 14:16:55 UTC 2025 : Method request query string: {}
Sun Aug 10 14:16:55 UTC 2025 : Method request headers: {}
Sun Aug 10 14:16:55 UTC 2025 : Method request body before transformations:
Sun Aug 10 14:16:55 UTC 2025 : Method response body after transformations: {"product_item_arr": [{"product_name_str": "apple pie slice", "product_id_str": "a444", "price_in_cents_int": 595, "description_str": "amazing taste", "tag_str_arr": ["pie slice", "on offer"], "special_int": 1}]}
Sun Aug 10 14:16:55 UTC 2025 : Method response headers: {Access-Control-Allow-Headers=Content-Type,X-Amz-Date,Authorization,X-Api-Key,X-Amz-Security-Token, Access-Control-Allow-Methods=GET, Access-Control-Allow-Origin=*, Content-Type=application/json}
Sun Aug 10 14:16:55 UTC 2025 : Successfully completed execution
Sun Aug 10 14:16:55 UTC 2025 : Method completed with status: 200
```


Task 4: Creating the third API endpoint (POST)

21. In the VS Code IDE, if the **create_products_api.py** file is not already open, open it (you ran this file in Task 2).
22. Next, in the **python_3** directory, also open the **create_report_api.py** file.
23. In the main code editor window, *right-click* the **create_report_api.py** file tab and choose **Split Pane in Two Columns**.
24. Analyze and update the **create_report_api.py** code. Be sure to compare the code in this file to the **create_products_api.py** code while you do the analysis and updates.
 - Replace the <FMI_1> that appears on line 5 with the correct value.

25. In the terminal, confirm that you are in the **python_3** directory, and then run the code to create the third endpoint.

python3 create_report_api.py

```
• [ec2-user@ip-10-0-1-150 python_3]$ python3 create_report_api.py
DONE
○ [ec2-user@ip-10-0-1-150 python_3]$
```


 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030

Task 5: Deploying the API

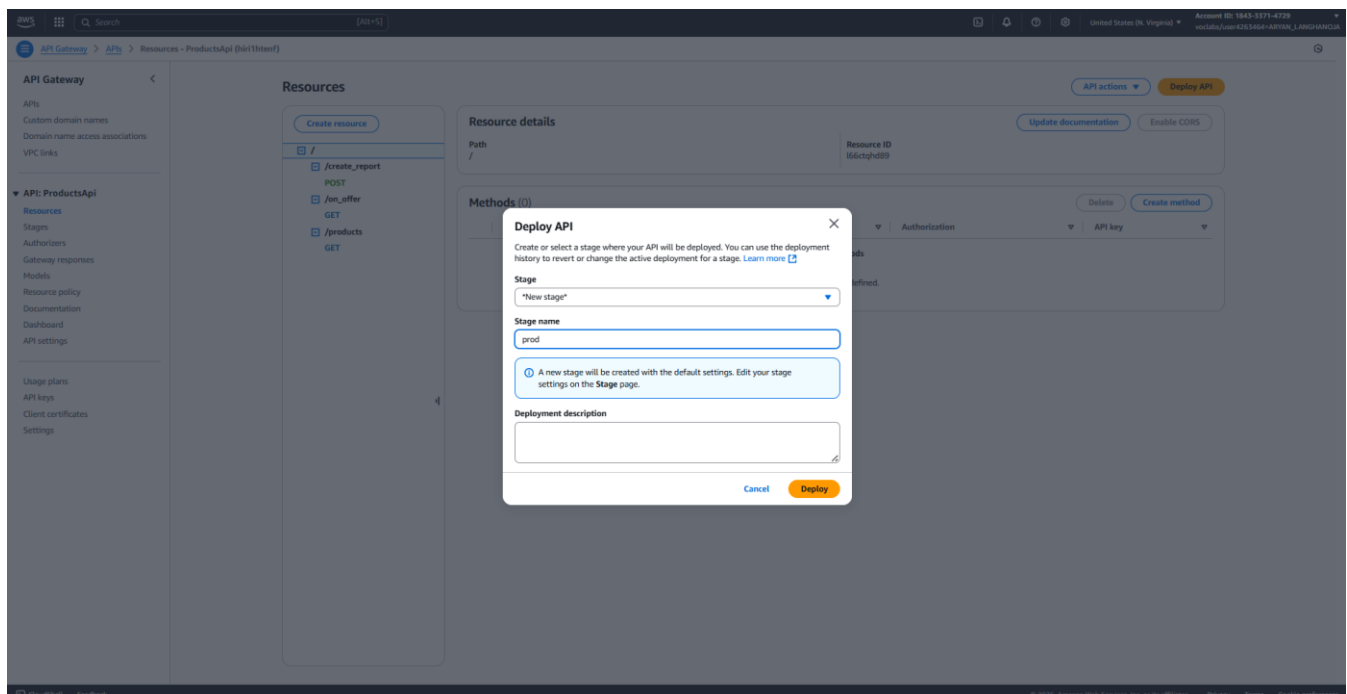
Now that you have defined all three resources in the API, the next step is to deploy the API.


28. Deploy the API.

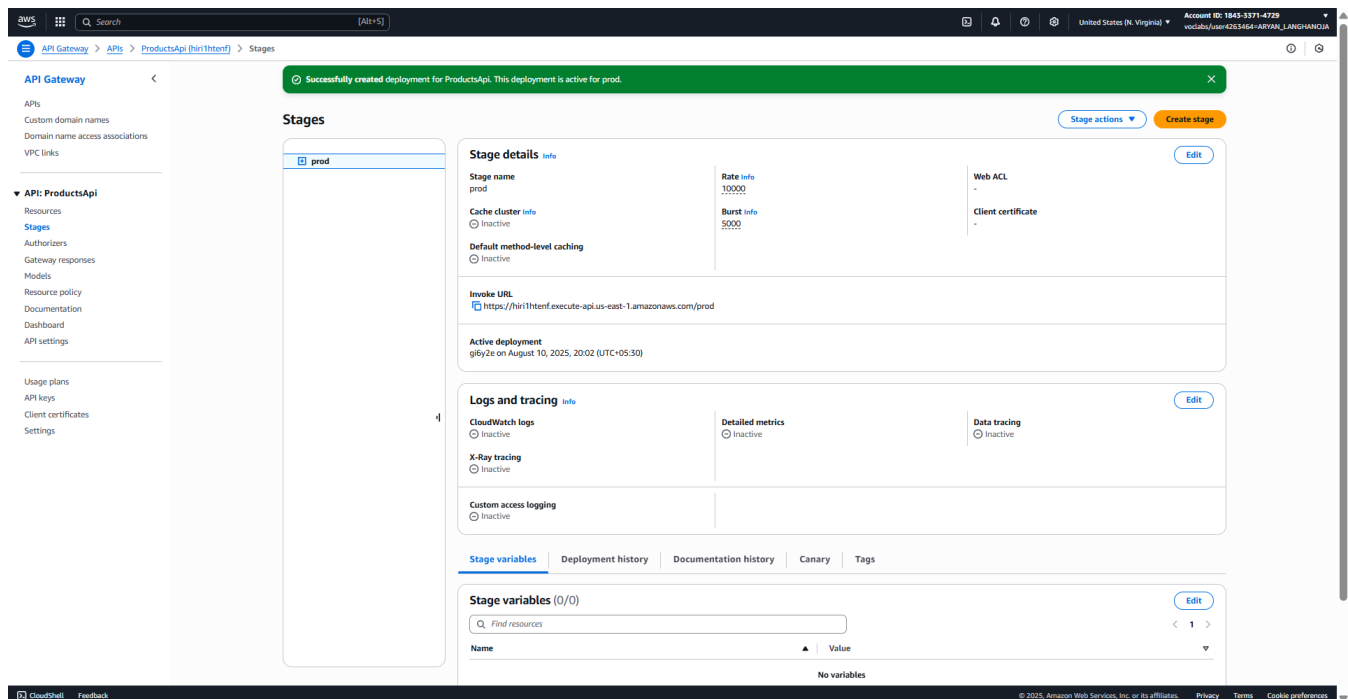
- Still in the API Gateway console where you have the ProductsApi details open, under **Resources** select the root /
- From the top, choose **Deploy API** button and then fill in the details:..
 - Deployment stage: ***New Stage***.
 - Stage name: **prod**
 - Stage description: (leave blank)
 - Deployment description: (leave blank)
- Choose **Deploy**

Tip: If you see a warning that you do not have ListWebACLs and AssociateWebACL permissions for Web Application Firewall (WAF Regional), you can ignore the message and close it.

29. Copy the **Invoke URL** value to your clipboard. You will use it next.



 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030



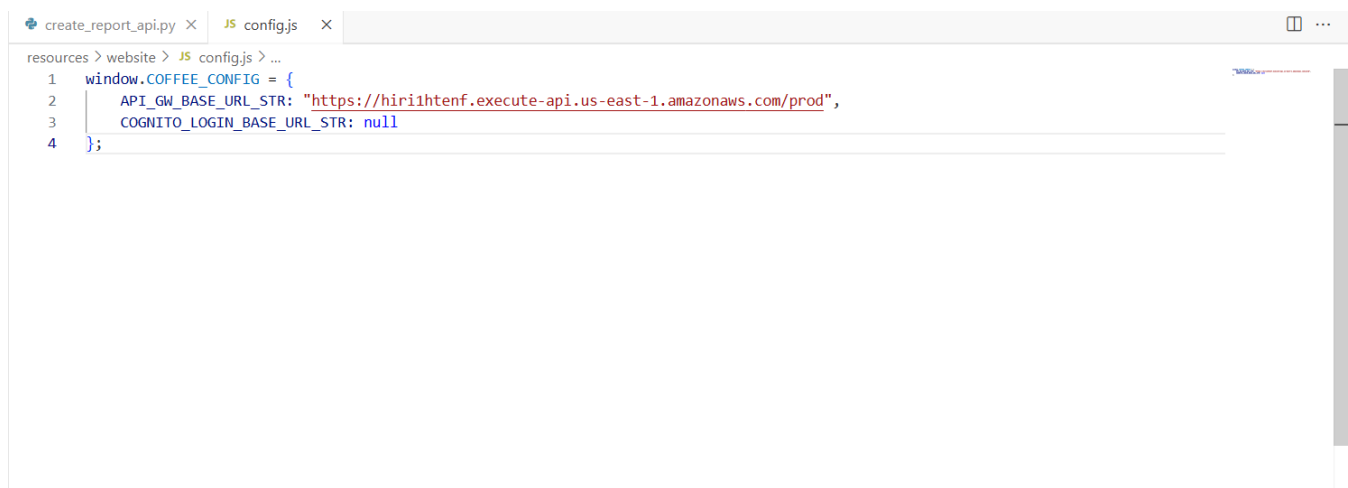
The screenshot shows the AWS API Gateway console for the 'ProductsApi' resource. A green banner at the top indicates 'Successfully created deployment for ProductsApi. This deployment is active for prod.' The left sidebar shows the navigation menu with 'API Gateway' selected. The main content area displays the 'Stages' section for the 'prod' stage. The 'Stage details' tab is active, showing the stage name 'prod', cache cluster 'inactive', default method-level caching 'inactive', and the invoke URL 'https://h1ri1htenf.execute-api.us-east-1.amazonaws.com/prod'. The 'Logs and tracing' section shows 'CloudWatch logs', 'X-Ray tracing', and 'Custom access logging' all as 'inactive'. The 'Stage variables' section is currently empty (0/0).

Task 6: Updating the website to use the APIs

In this final task in the lab, you will update and then test the website files that are hosted on Amazon S3. After you complete these updates, the website will invoke the REST API that you just created.

30. Update the website's **config.js** file.


- In the VS Code IDE, open resources/website/**config.js**



The screenshot shows the VS Code IDE with the 'config.js' file open. The file content is as follows:

```

1 window.COFFEE_CONFIG = {
2   API_GW_BASE_URL_STR: "https://h1ri1htenf.execute-api.us-east-1.amazonaws.com/prod",
3   COGNITO_LOGIN_BASE_URL_STR: null
4 };
  
```


 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	Aim: Developing REST APIs with API Gateway.	
Experiment No: 05	Date:	Enrolment No: 92200133030

31. Update and then run the **update_config.py** script.

- Open **python_3/update_config.py** in the text editor.
- Replace the <FMI_1> placeholder with the name of your bucket.

Tip: You can find the bucket name in the S3 console, or by running this command:

aws s3 ls

- Notice that this script uploads the config.js file that you just editing the previous step, and uploads it to the S3 bucket.
- Close the file by choosing **X** from the top. (Your changes are saved automatically).

python3 update_config.py

```
[ec2-user@ip-10-0-1-150 python_3]$ aws s3 ls
2025-08-10 13:46:24 c168617a4340248111142234t1w184333714729-s3bucket-1wvveyyvhv5l
[ec2-user@ip-10-0-1-150 python_3]$
```

Ln 4, Col 3 Tab Size: 4 UTF-8 LF { } JavaScript Layout: US

```
[ec2-user@ip-10-0-1-150 python_3]$ python3 update_config.py
DONE
[ec2-user@ip-10-0-1-150 python_3]$
```

Conclusion:-

- In this Lab I Learned the AWS API Gateway Service.
- I had created the REST API of GET Method /product to retrieve all the product.
- I had created the REST API of GET Method /product/on_offer to retrieve all the product having offer.
- I had created the REST API of POST Method /create_report to create the report.
- I had deployed that API