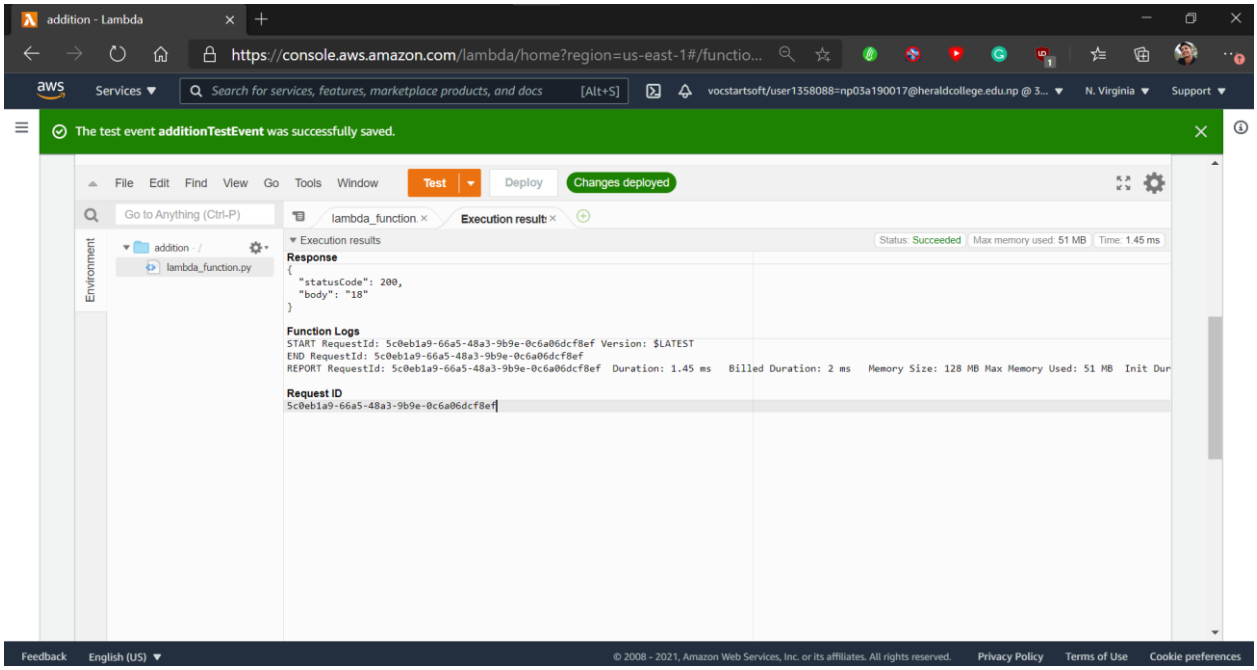
Module	Assessment Type
Distributed and Cloud Systems Programming	Individual Report

Workshop 5

Student Id : 2049867 (NP03A190017)
Student Name : Roshan Parajuli
Section : L5CG3
Module Leader : Rupak Koirala
Lecturer /Tutor : Saroj Sharma
Submitted on : 2020-04-17

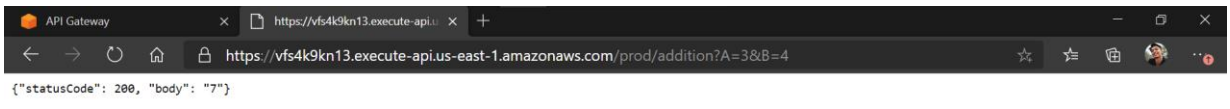
Introduction

For this task, a multi-function calculator is to be hosted on the S3 bucket and with the help of AWS Lambda and Amazon API Gateway, the same calculator REST API is to be deployed. The completion of the tasks will follow through the steps of creating a new lambda function in the lambda console, testing it, creating the rest API for the required operations and then deploying it.

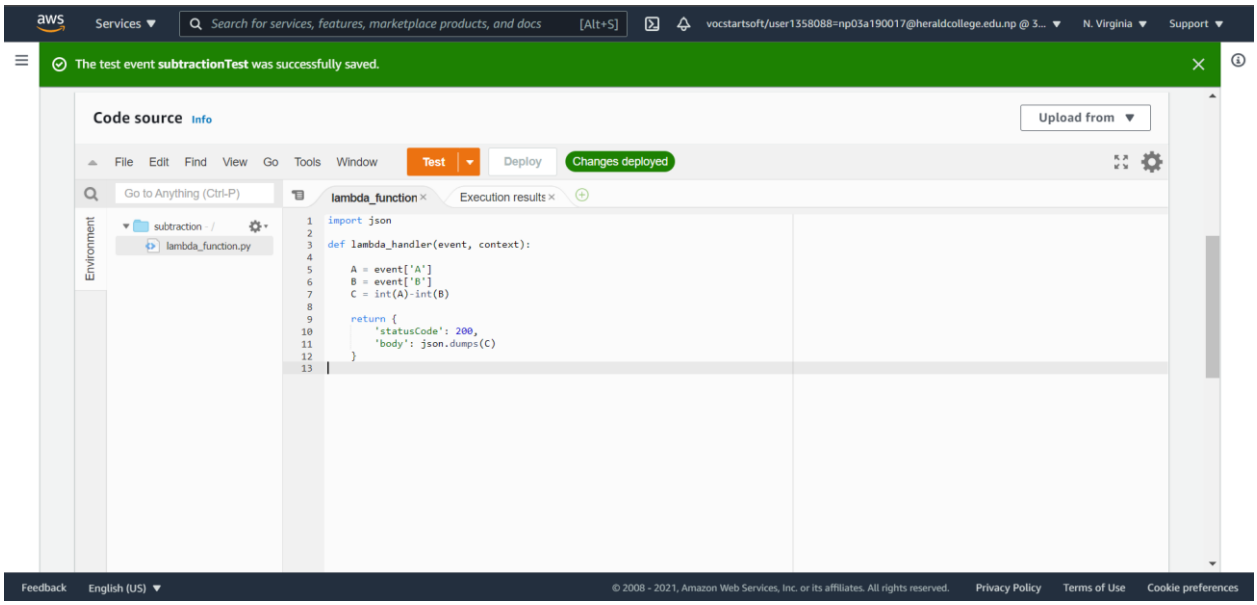


After the lambda function is created for addition, it was tested with the help of an event. Here, the test event for addition is made and invoked which successfully returned the status code of 200 and body key corresponds to the correct outcome for the addition of two given input numbers. **This is the first required screenshot of the workshop.**

After that, a new rest API was created, and the “addition” resource was created and attached with the function with the same name. In addition, a mapping template was created with the codes provided in the workshop document.



When two parameters are provided via URL, i.e. A and B will the values 3 and 4 respectively, the object returned as the outcome correctly displayed the addition value.

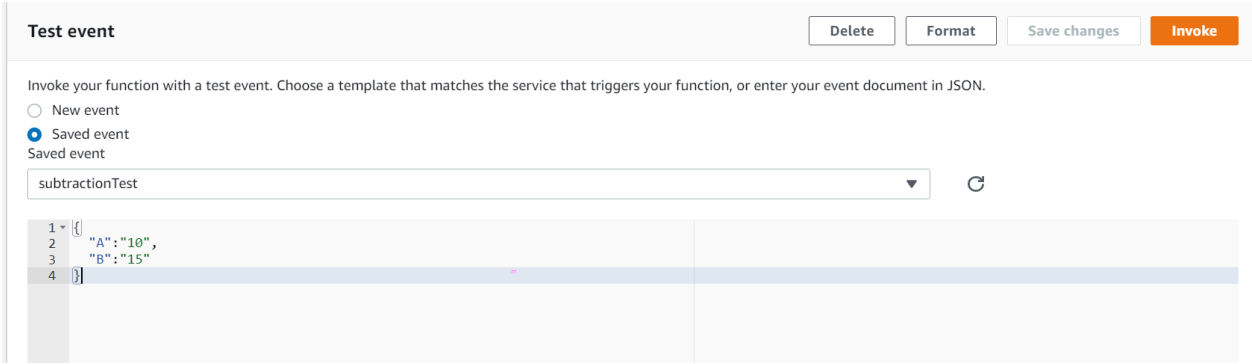


Lambda function for addition

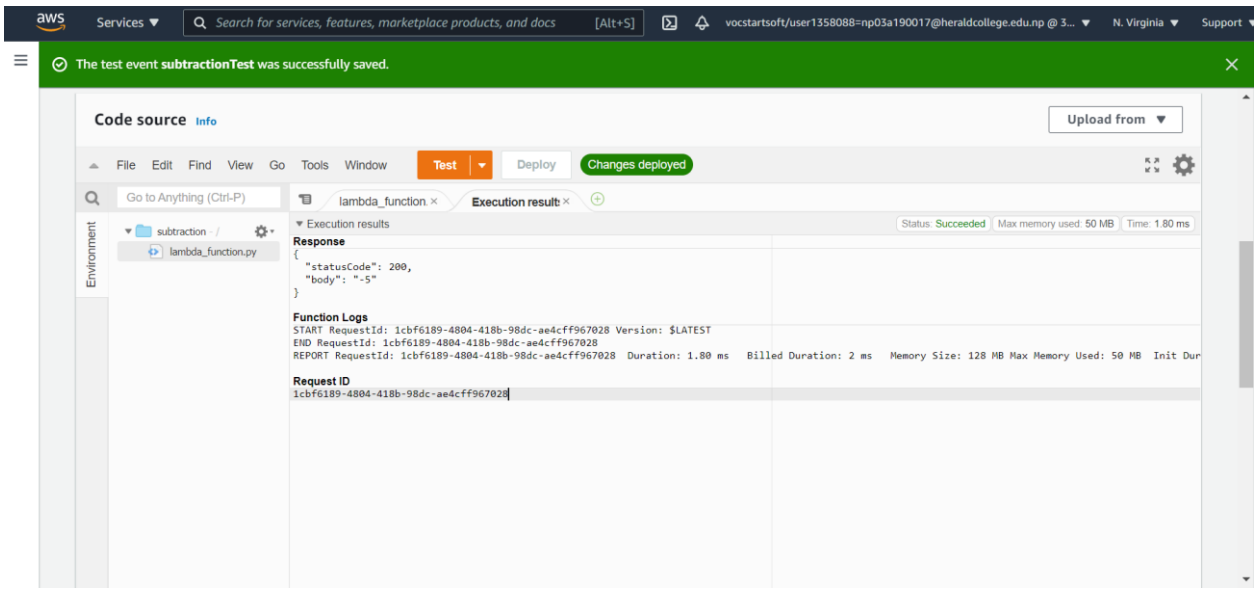
```
1. import json
2. def lambda_handler(event, context):
3.         A = event['A']
4.         B = event['B']
5.         C = int(A) + int(B)
6.         return {
7.          'statusCode': 200,
8.          'body': json.dumps(C)
9.           }
```

This is the lambda function for the addition operation which take two parameters A and B from input and returns the object containing the result of addition between those two as well as the status code.

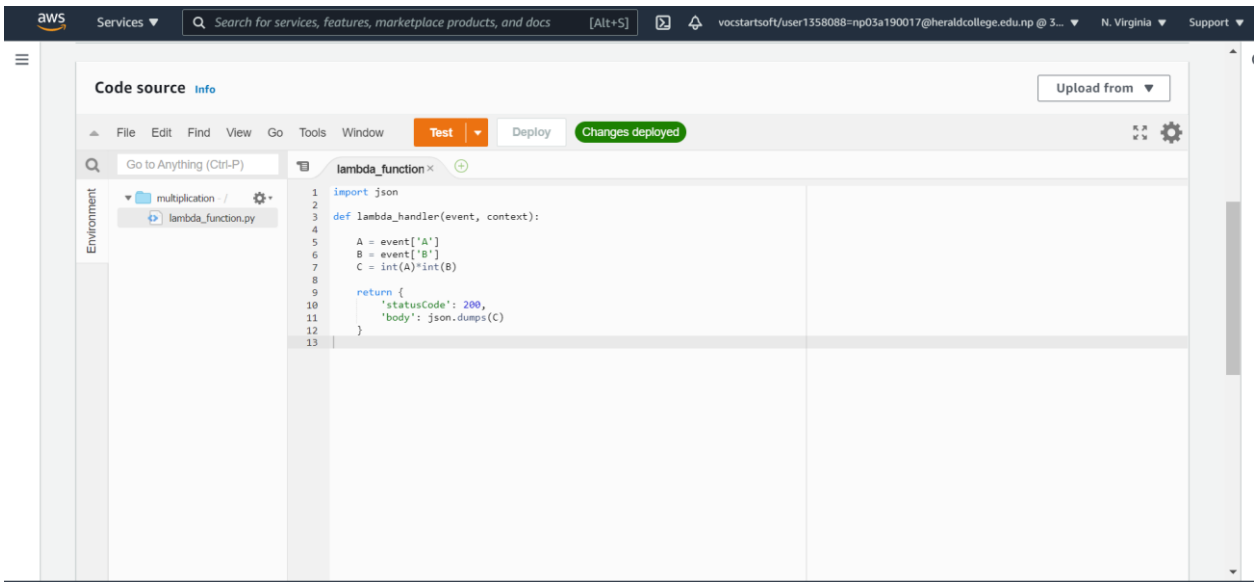
For other operations such as: subtraction, multiplication and division the sign is changed in line number 5 correspondingly.



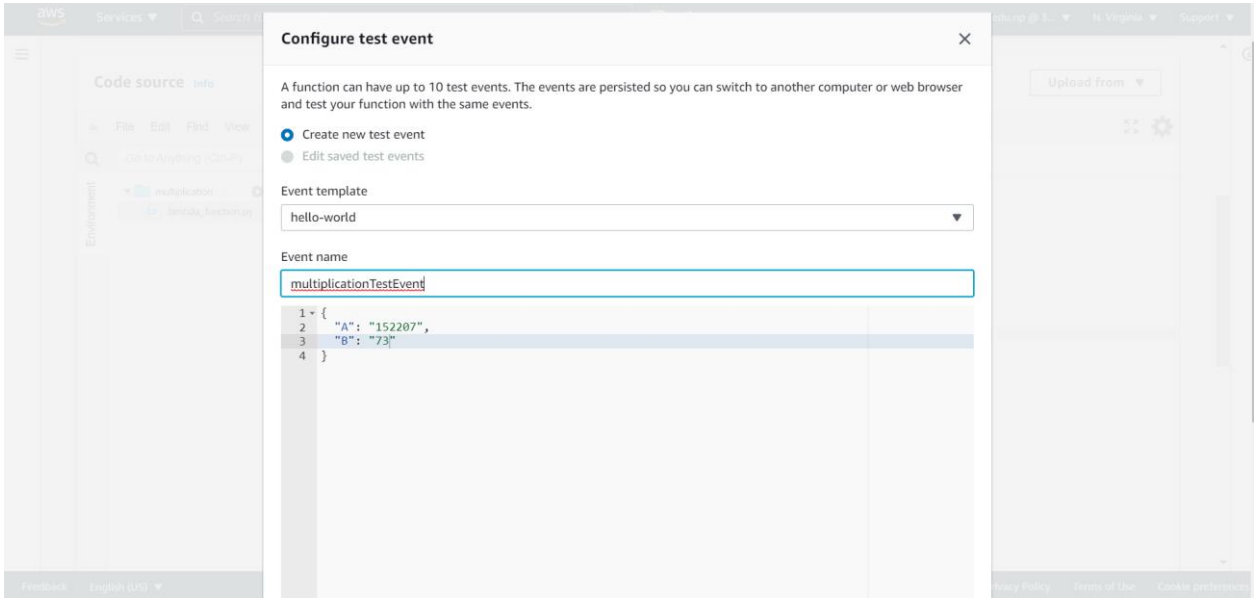
Here, a new test event is created which defines a set of values for the keys A and B and the addition of values of those two keys is expected. After the test event is invoked, the following outcome (shown in the screenshot below) can be seen.



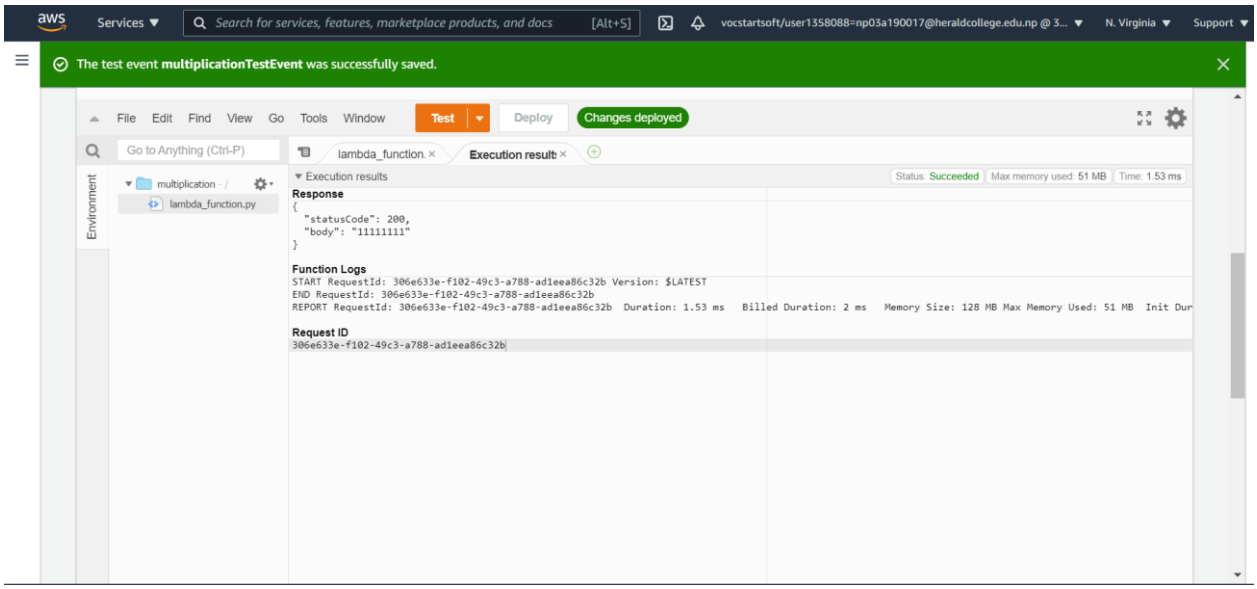
The expected outcome matched up with the actual result as the value of the body key is shown to be -5 and the input provided (as seen in the above screenshot) is 10-15. The lambda function did it job accurately.



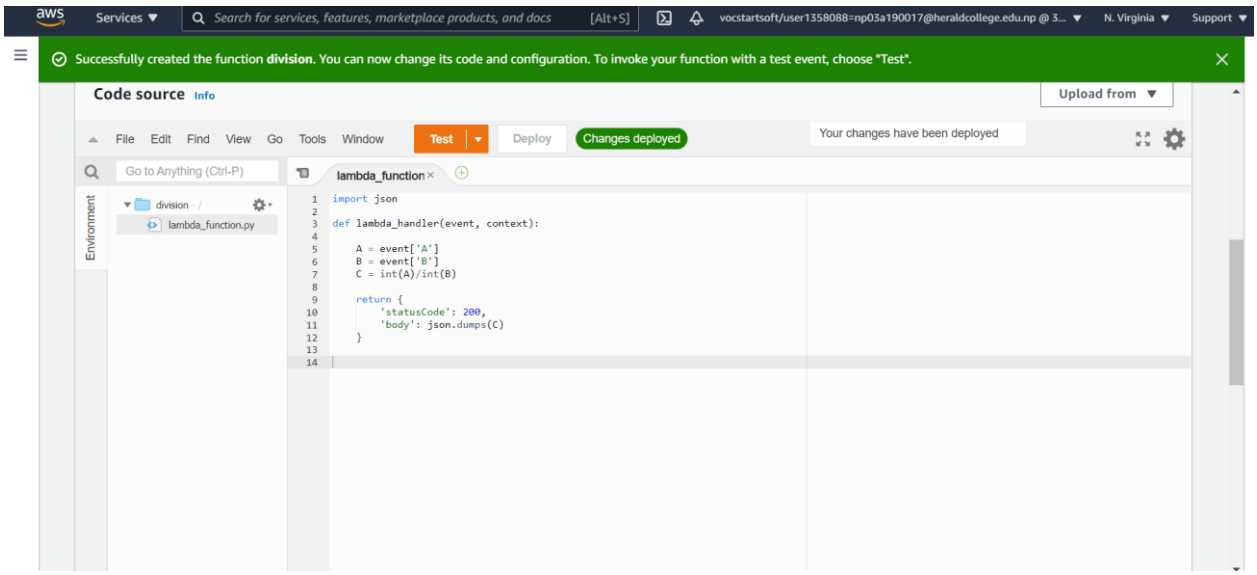
Here, the lambda function for multiplication is created with the same logic as the addition function.



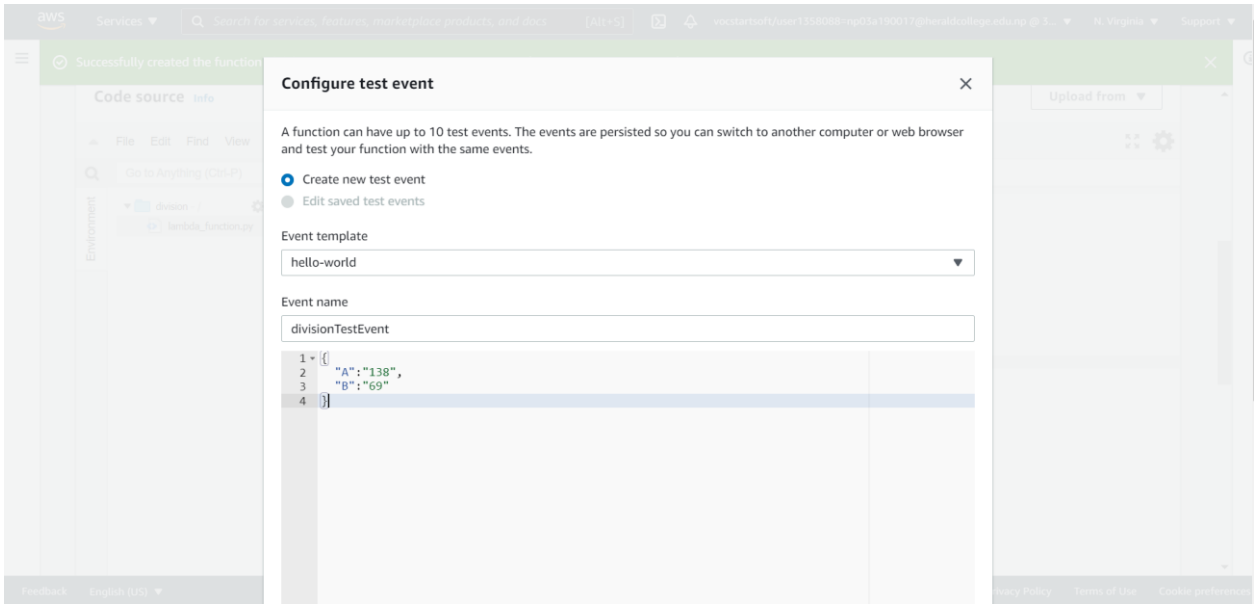
Here, a new test event is created which defines a set of values for the keys A and B and the multiplication of values of those two keys is expected. After the test event is invoked, the following outcome (shown in the screenshot below) can be seen.



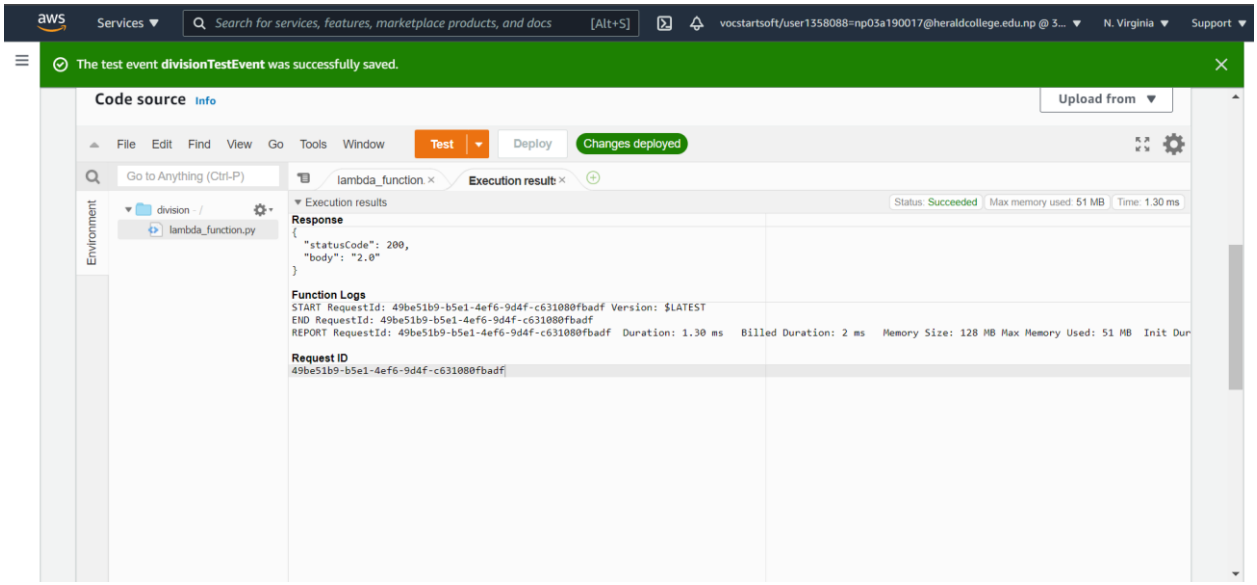
The expected outcome matched up with the actual result as the value of the body key is shown to be 11111111 and the input provided (as seen in the above screenshot) is 152207*73. The lambda function did it job accurately again.



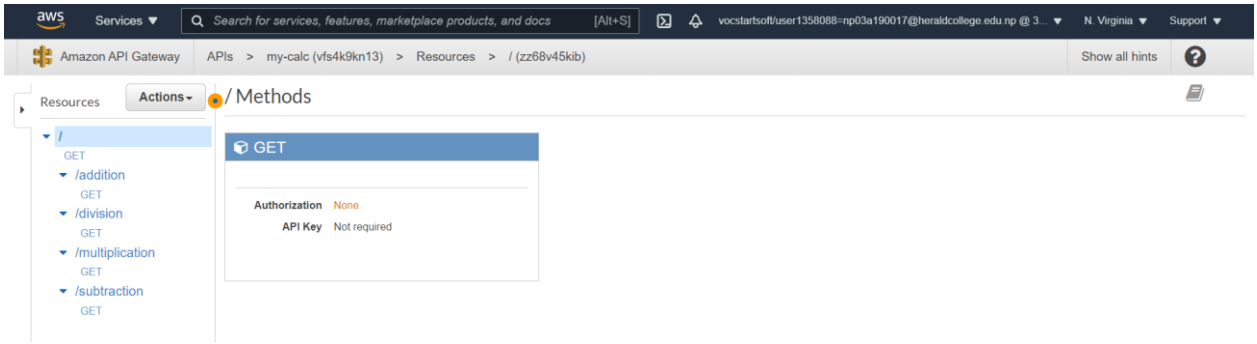
Here, the lambda function for division is created with the same logic as the addition function.



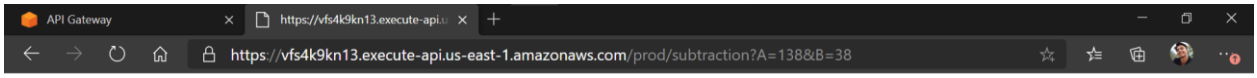
Here, a new test event is created which defines a set of values for the keys A and B and the division of the values of A by B is expected. After the test event is invoked, the following outcome (shown in the screenshot below) can be seen.



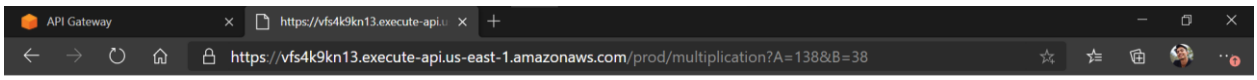
The expected outcome matched up with the actual result as the value of the body key is shown to be 2.0 and the input provided (as seen in the above screenshot) is 138/69. The lambda function did it job accurately again.



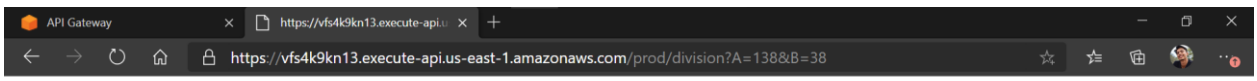
Here, a new resource is created which is denoted by root or slash(/) and four new methods addition, division, multiplication and subtraction were created for the required operations.



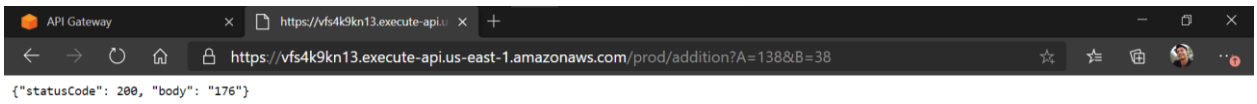
{"statusCode": 200, "body": "100"}



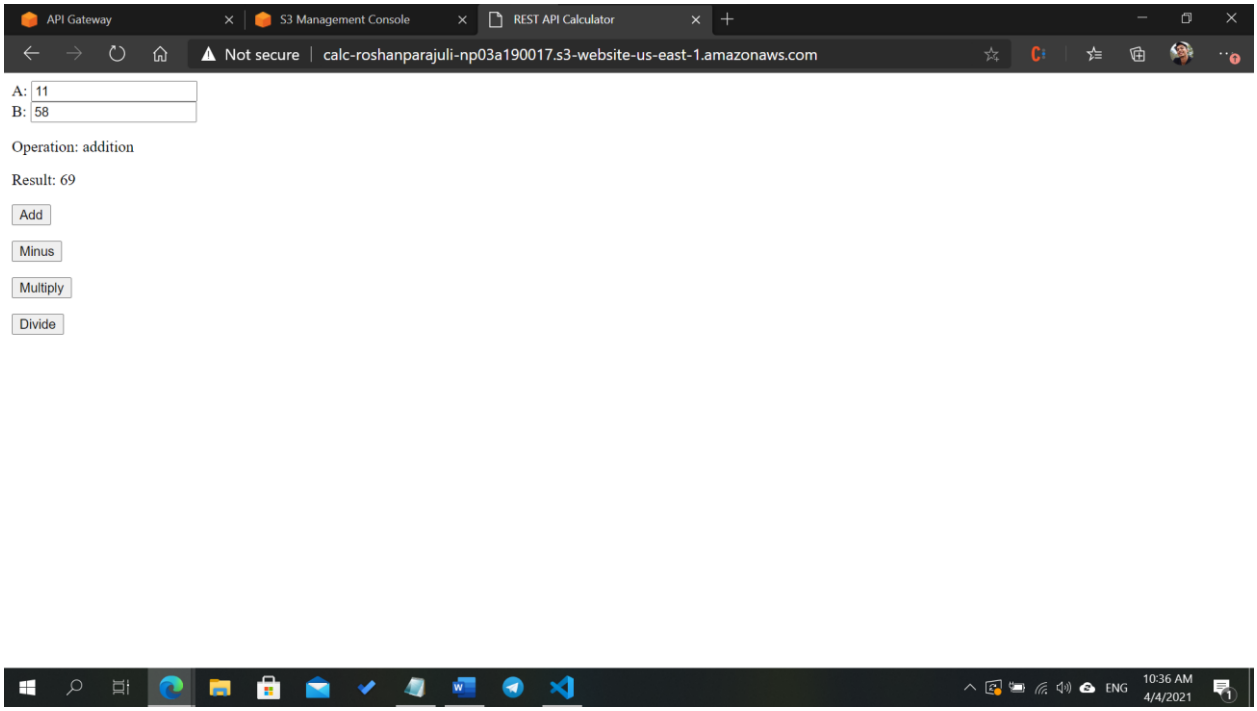
{"statusCode": 200, "body": "5244"}



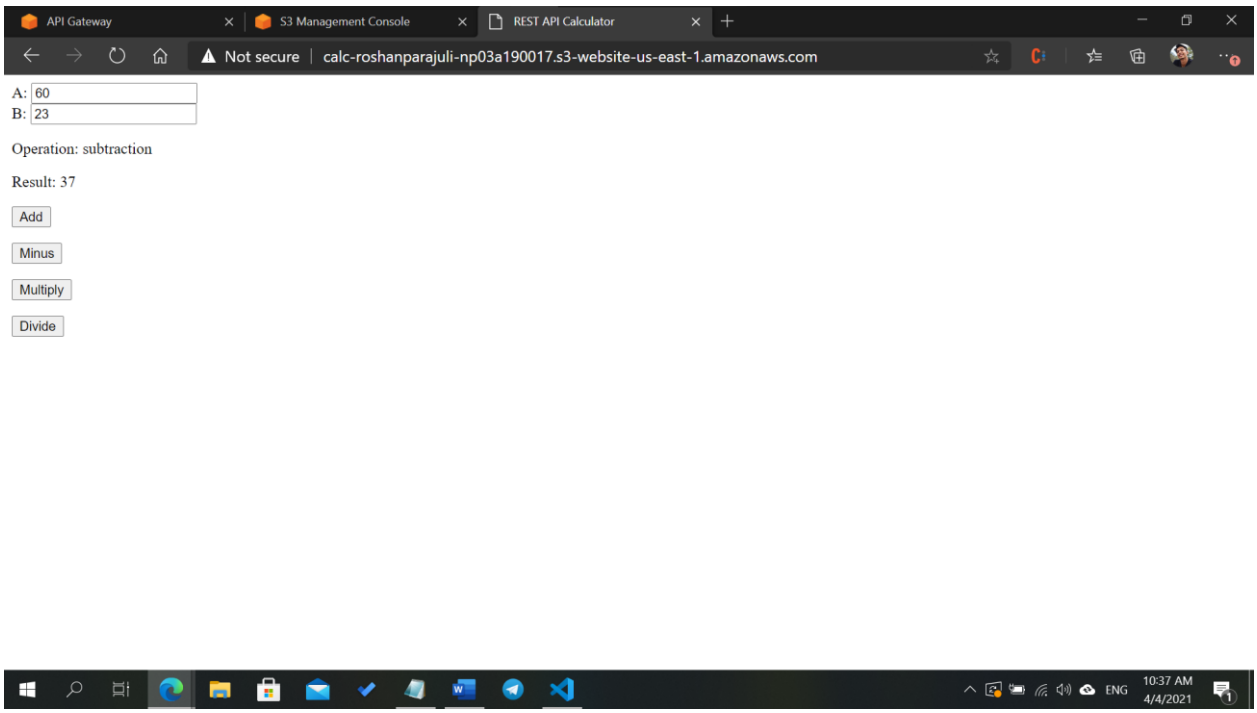
{"statusCode": 200, "body": "3.6315789473684212"}



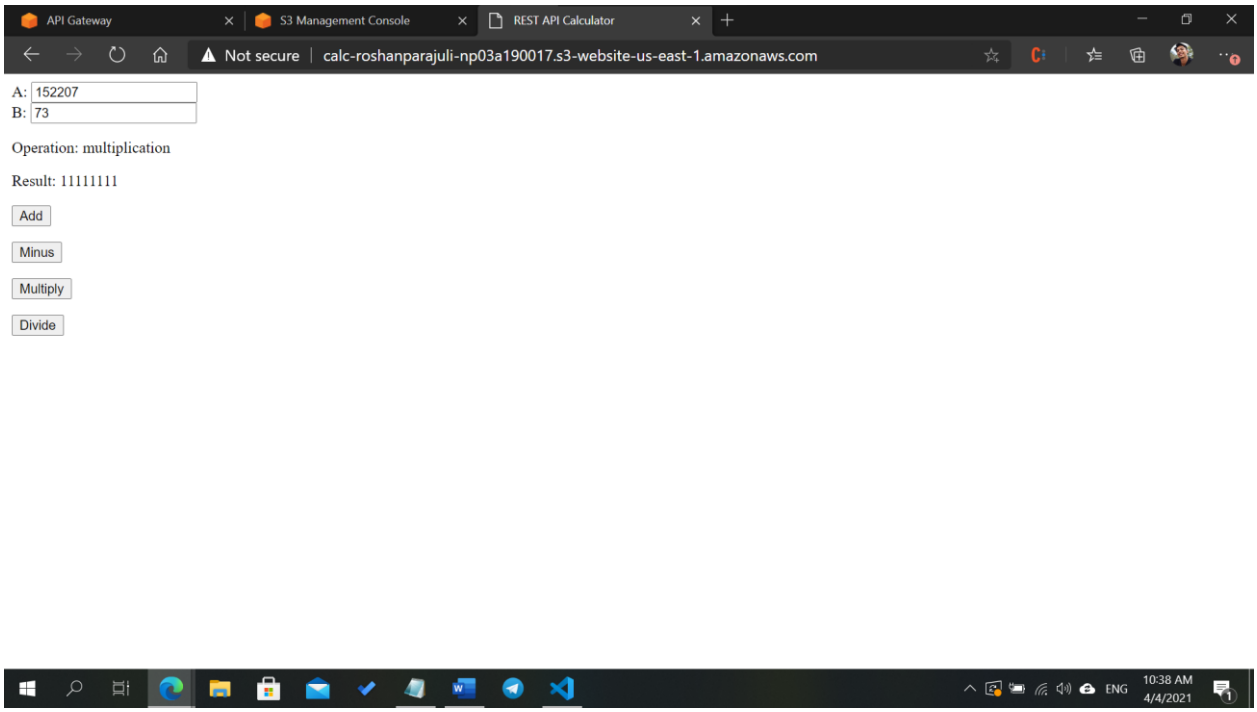
These final four screenshots prove the accuracy of the workshop as all the four operations are performing their job accurately. After this step is completed, CORS policy was enabled on the each resource and the website codes for the calculator app were put into the S3 bucket for hosting purpose.



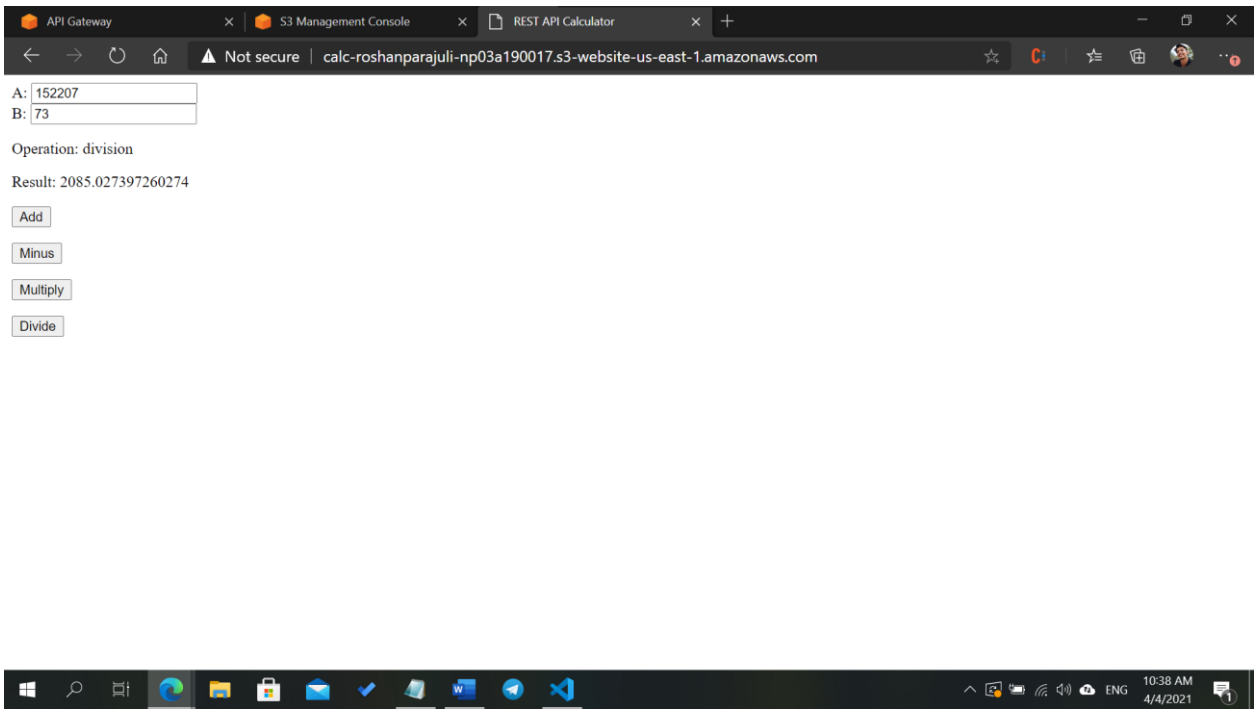
Proof of the addition function working well.



Proof of the subtraction function working well.



Proof of the multiplication function working well.



Proof of the division function working well.

Conclusion

To summarize the tasks that I have successfully done in this workshop, the multifunctional calculator was successfully turned into a working piece as the deployment of the REST API with AWS Lambda and Amazon API Gateway was done following the document.