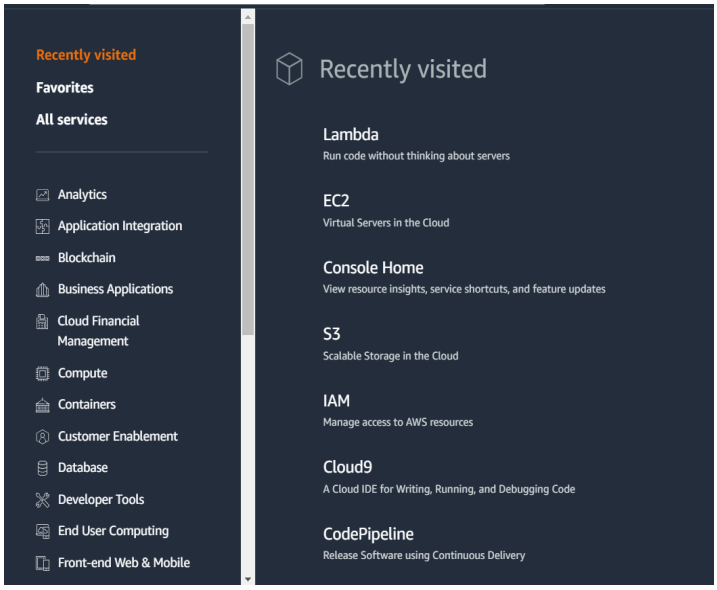
Name: Aditya Dubey **Div: D15C** **Roll No: 10**

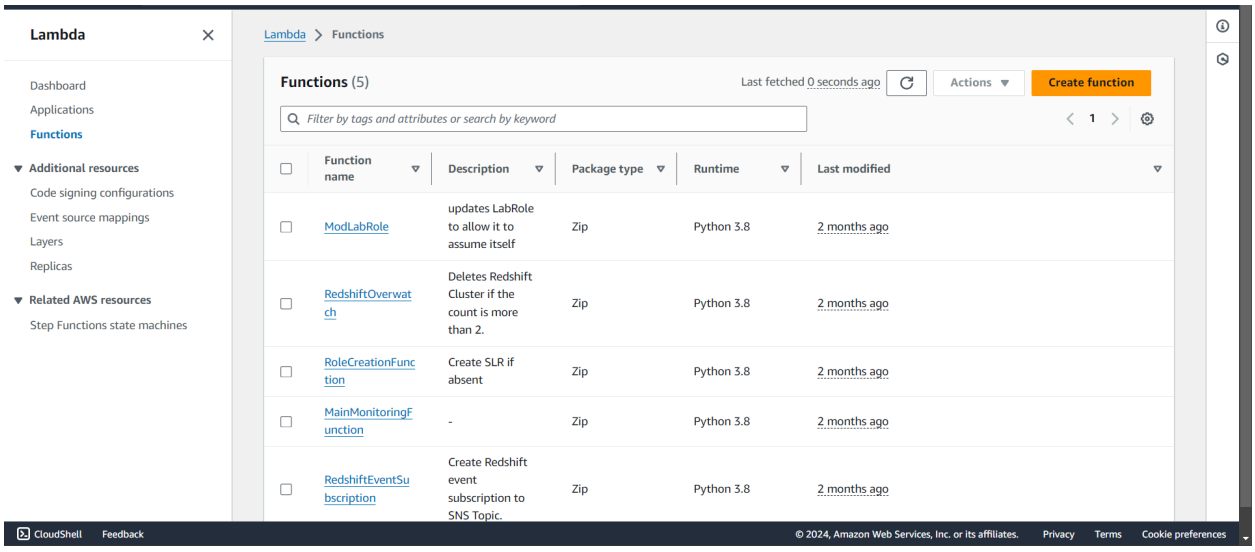
Experiment No: 11

AIM: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

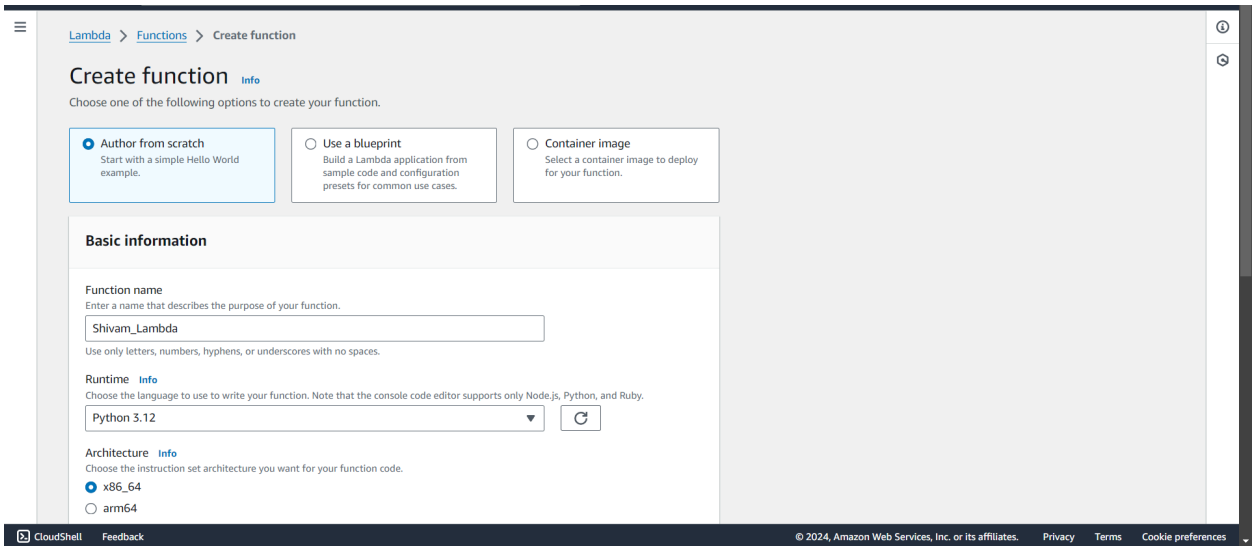
CREATION OF LAMBDA FUNCTION:

Step1: Log in to your AWS Personal or Academy account. Navigate to Lambda, then select the 'Create Function' button

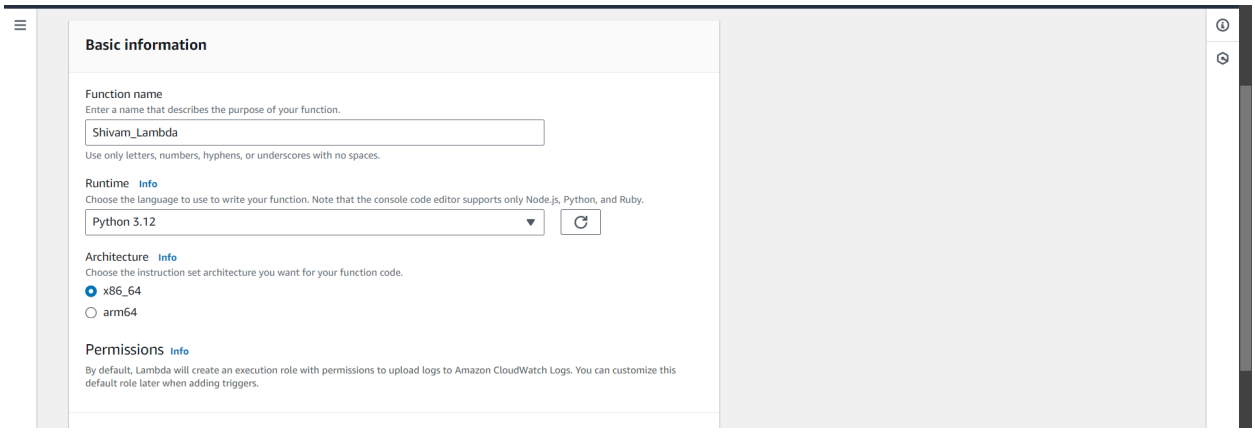




Step 2: Give your Lambda function a name and choose a programming language. The code editor only supports Node.js, Python, and Ruby, so in my case I have chosen **Python 3.12**. Set the **architecture to x86**. For the execution role, select '**Use an existing role**,' then pick '**Lab role**' from the dropdown menu under existing roles .
(This is because the Lab role already has the permissions needed for Lambda to run properly, so you don't need to create a new role from scratch. It's a quicker and more convenient option)

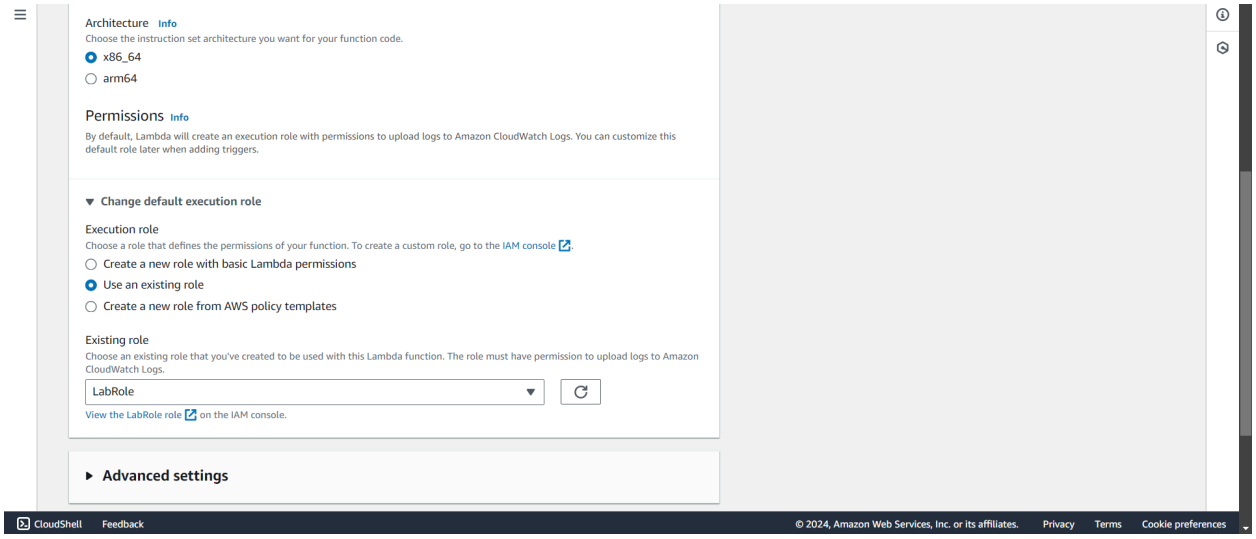


Give the function name and select required language for lambda function

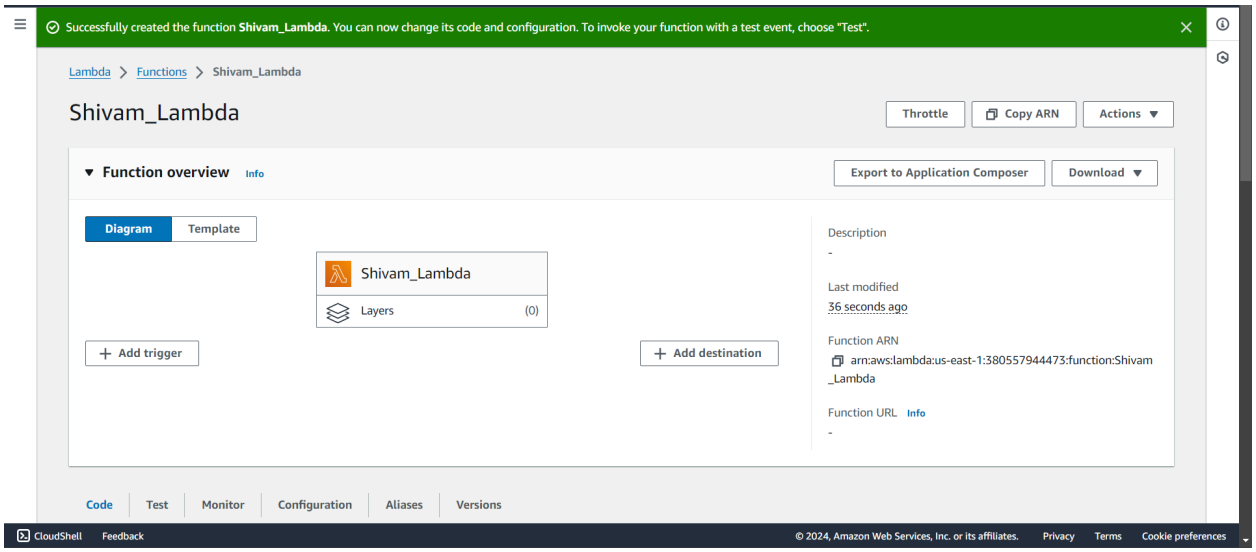


Architecture will be x86_64

Name: Aditya Dubey **Div: D15C** **Roll No: 10**



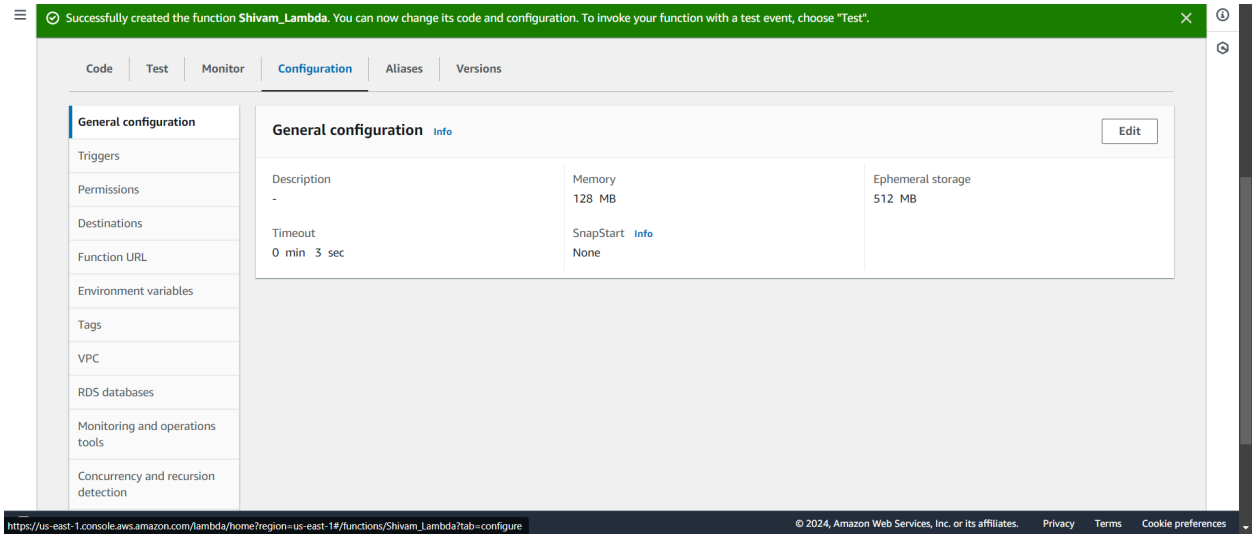
Select proper Execution role



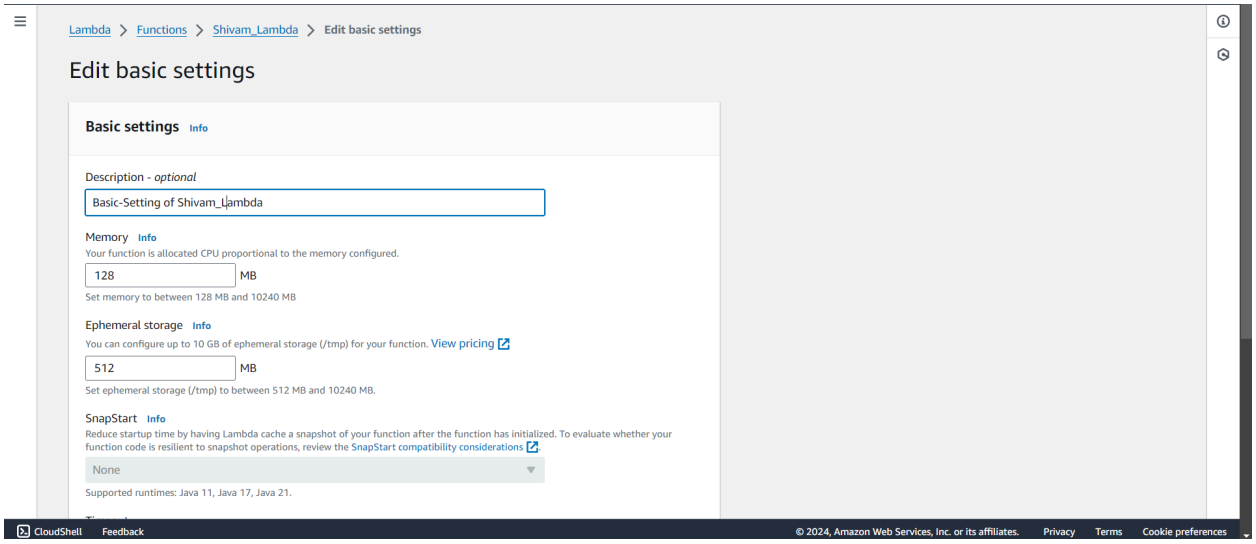
Successfully created Lambda function

Step 3: To view or change the basic settings, go to the 'Configuration' tab and click 'Edit' under 'General settings.' (THIS STEP IS OPTIONAL)

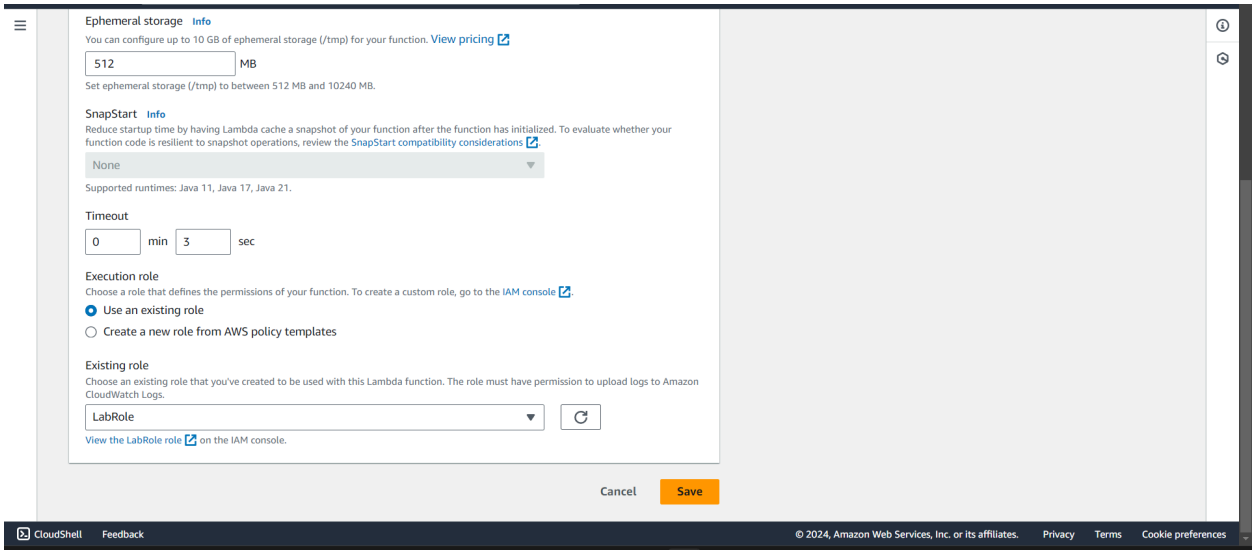
Name: Aditya Dubey **Div: D15C** **Roll No: 10**



You can add a description and adjust the memory and timeout settings. I've changed the timeout to 1 second, as that's enough for now.



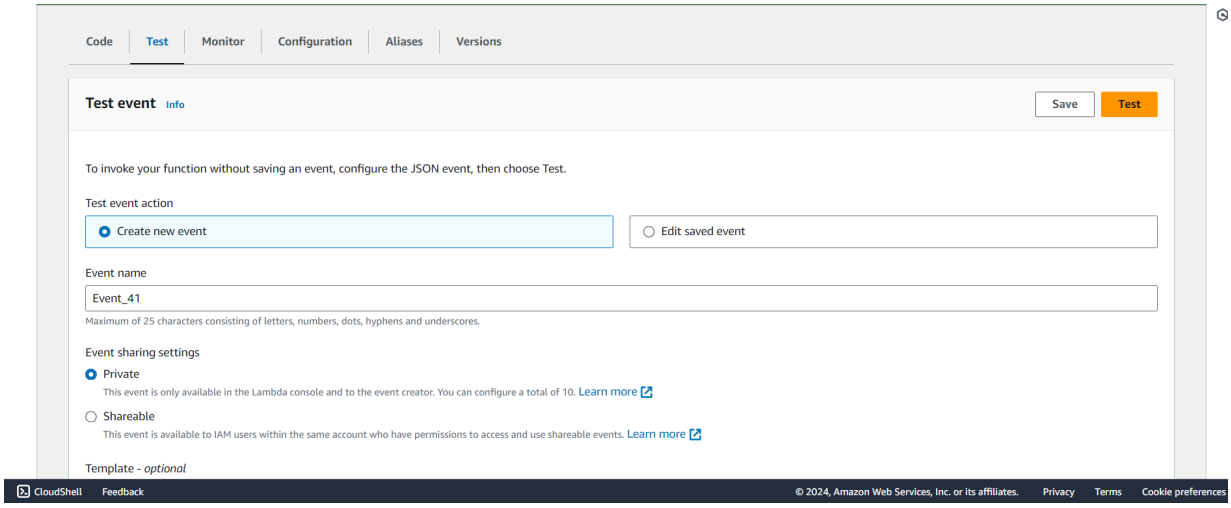
Name: Aditya Dubey **Div: D15C** **Roll No: 10**



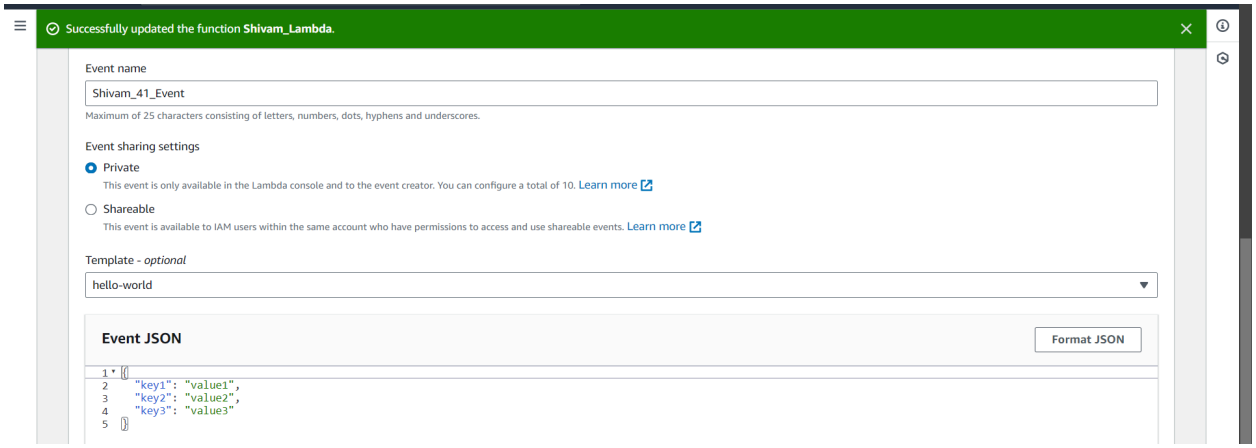
Click on Save

Step 4: Go to the 'Test' tab and click 'Create a new event.' Give the event a name, set 'Event Sharing' to private, and choose the 'hello-world' template.

We basically create a new event to test and verify your Lambda function; setting Event Sharing to private keeps it secure and choosing the "hello-world" template provides a simple structure for testing without complex inputs.

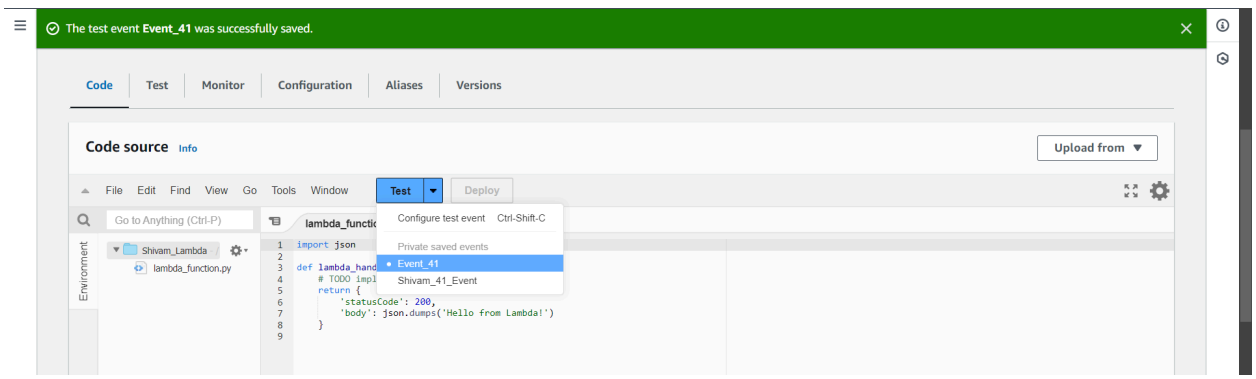


Name: Aditya Dubey **Div: D15C** **Roll No: 10**

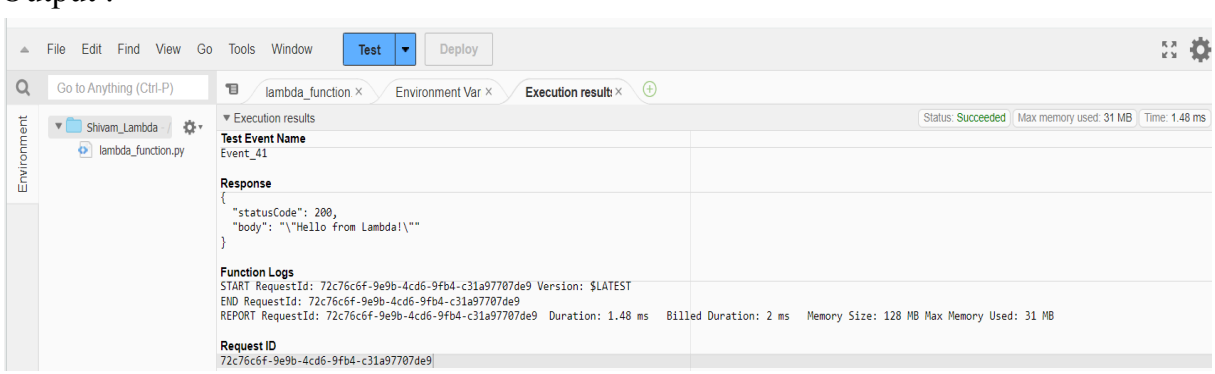


Click on save button above

Step 5: In the Code section, select the event you created from the dropdown menu under 'Test,' then click 'Test.' You should see the output below."



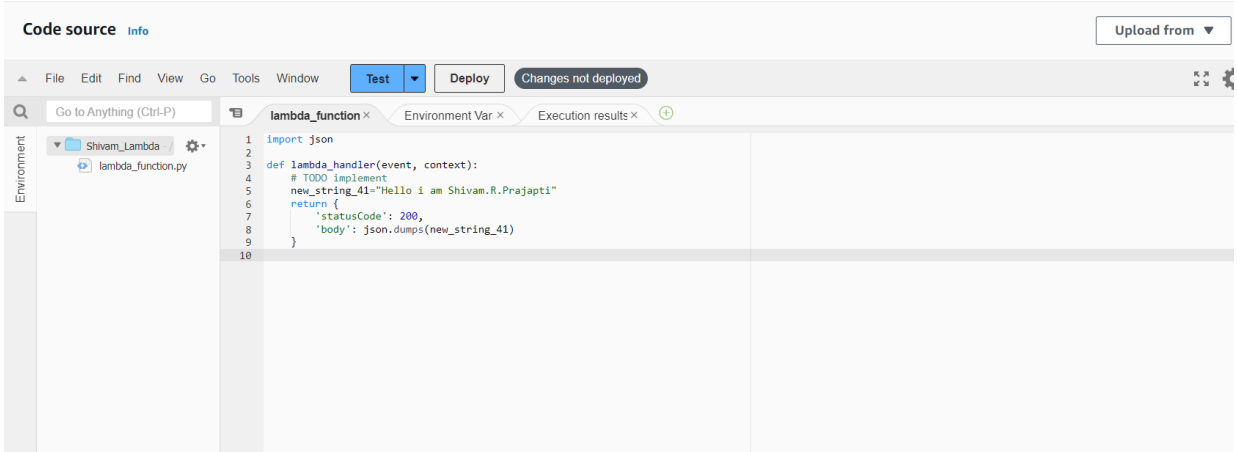
Output :

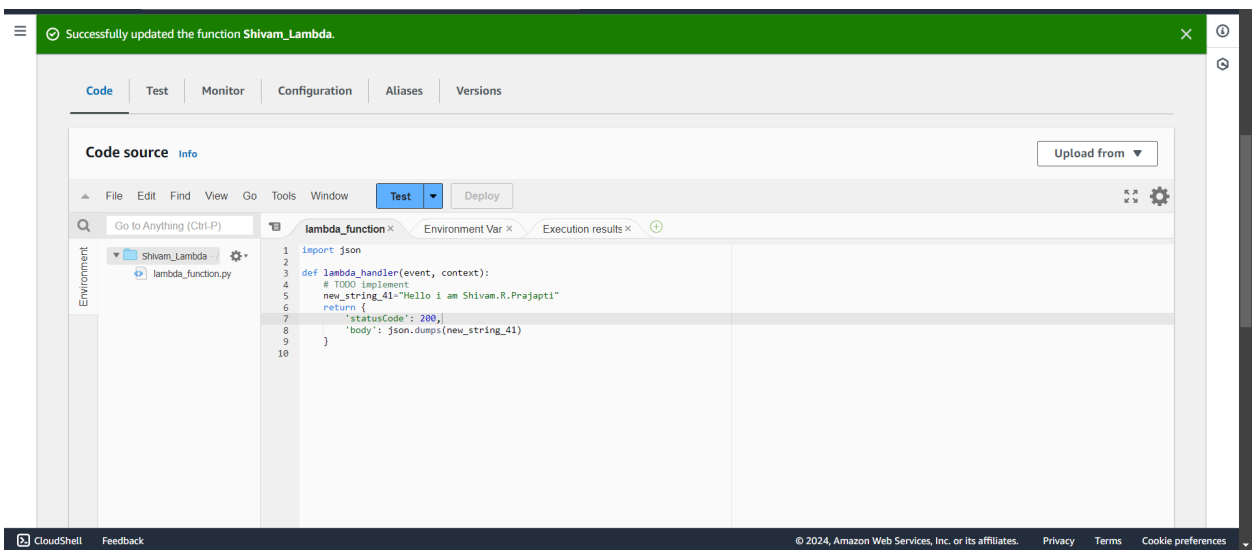


You select the created event to run the specific test you set up, and clicking 'Test' executes your Lambda function to check if it works as expected and produces the desired output.

Name: Aditya Dubey **Div: D15C** **Roll No: 10**

Step 6: You can edit your lambda function code. I have changed the code to display the new String. After Changing save it by Control + S and click on Deploy . Make sure you have internet connectivity while deploying or else it will show failed deployment

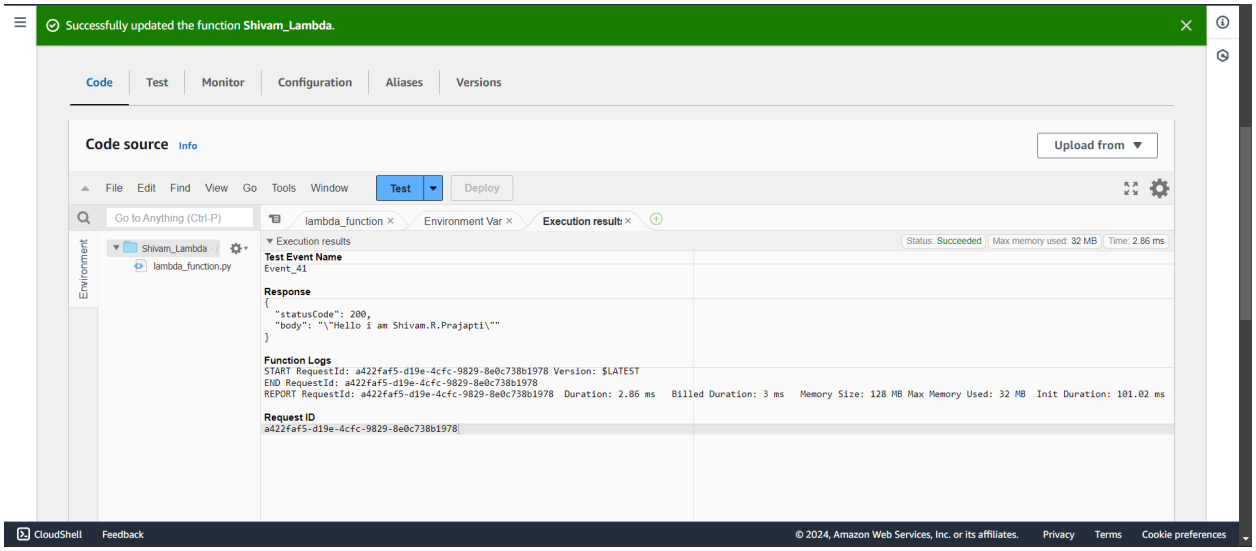




Successfully changed the function.

Name: Aditya Dubey **Div: D15C** **Roll No: 10**

Step 7: Click on '**Test**' to see the output. You'll get a status code of **200** which means "OK" and indicates that the request was successful, your string output, and the function logs, showing that it was deployed successfully.



CONCLUSION:

In this experiment, we successfully created an AWS Lambda function and followed the important steps involved. First, we set up the function using Python and adjusted the timeout setting to 1 second. Then, we created a test event to see how the function works and checked the output to ensure it was correct. We also modified the function's code and redeployed it to see the changes in real-time. So Lambda Function allows you to concentrate on writing code while AWS manages the infrastructure and automatically scales the service as needed. This makes it easier to develop and run applications without worrying about server management.