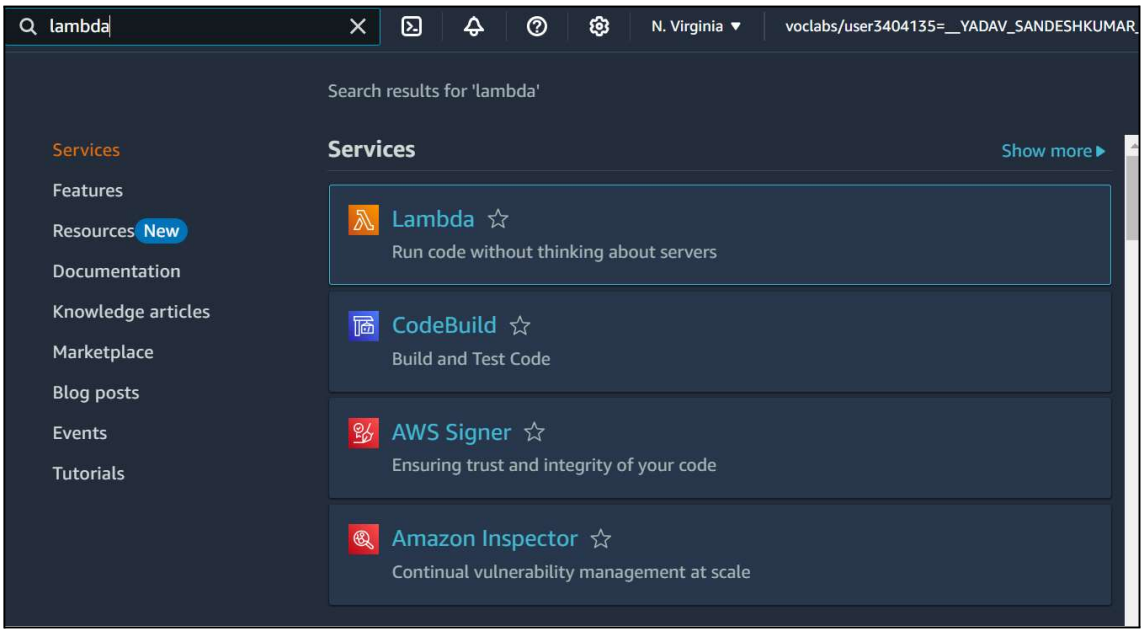
**Name : Sandeshkumar.M.Yadav** **Div : D15C** **Roll : 61**

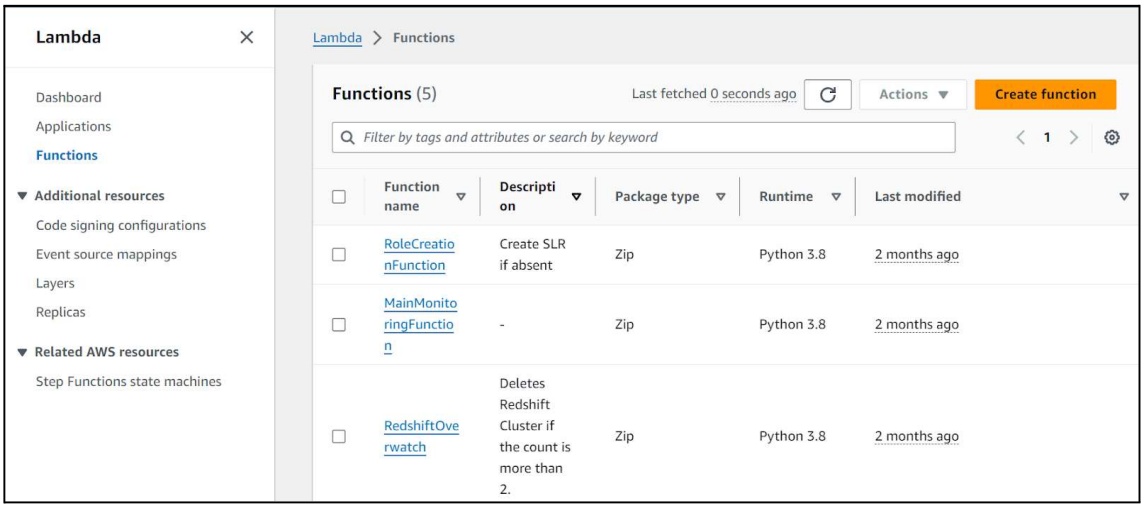
Experiment No: 11

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

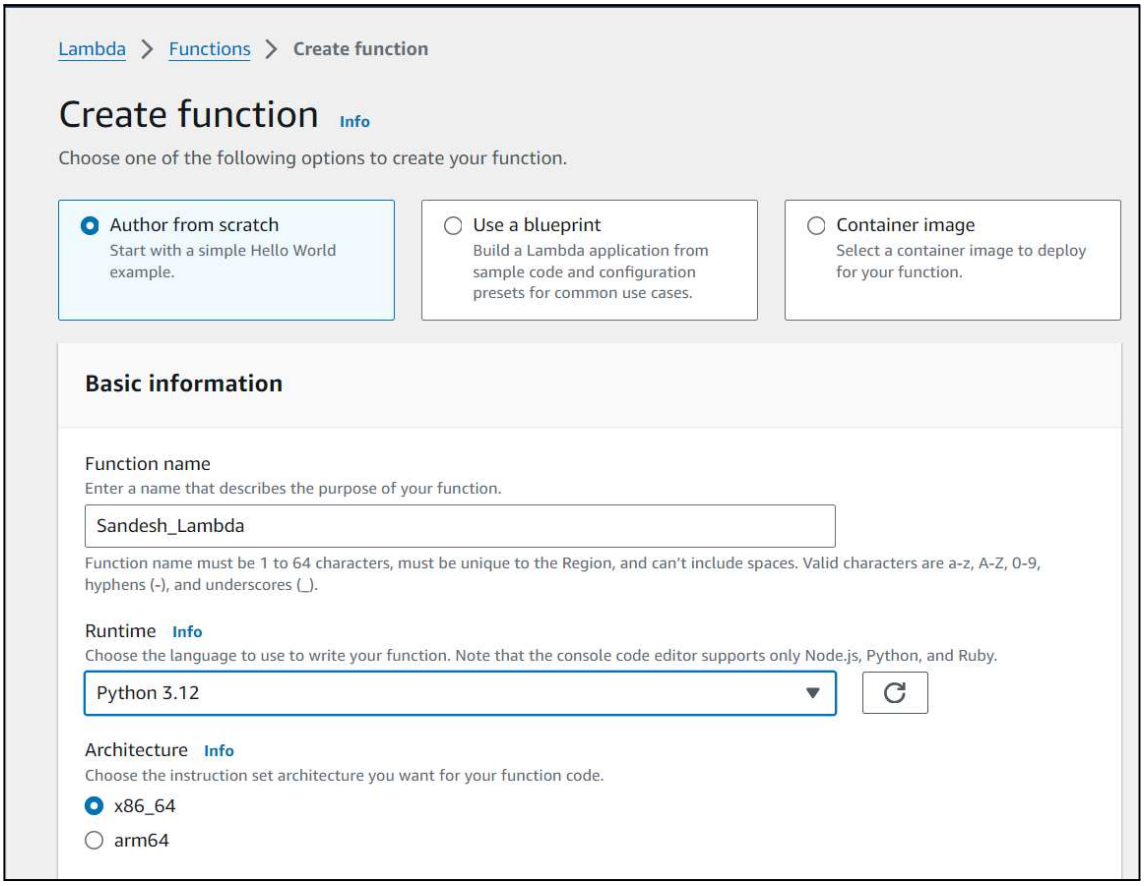
**CREATION OF LAMBDA FUNCTION:**

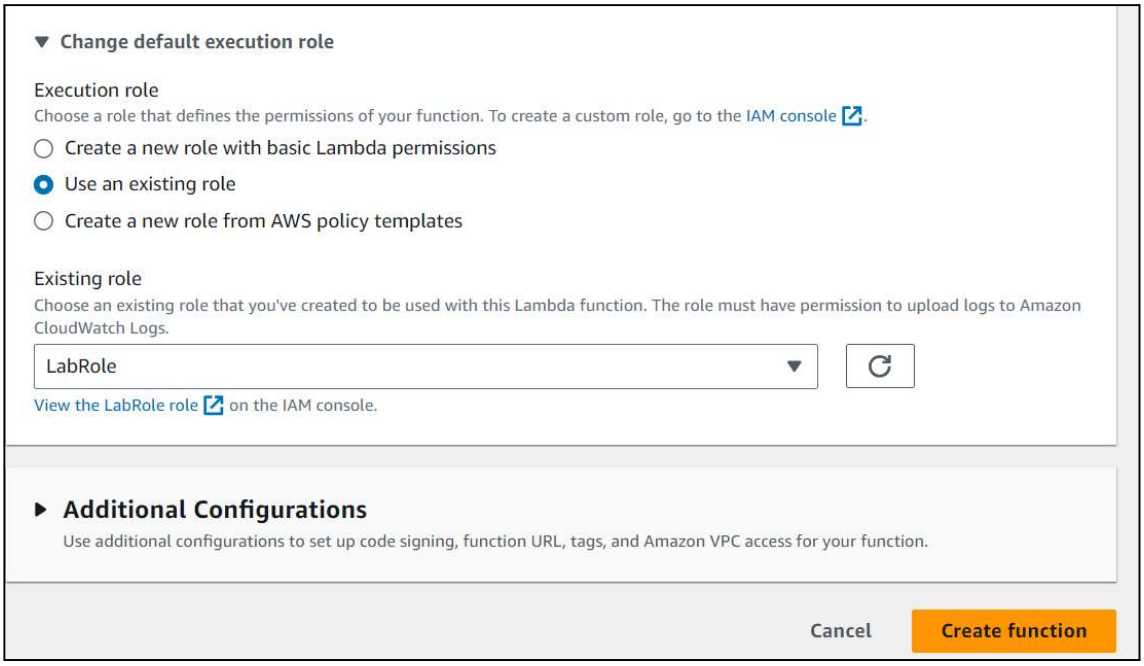
**Step1** : Log in to your AWS Personal or Academy account. Navigate to Lambda, then select the 'Create Function' button

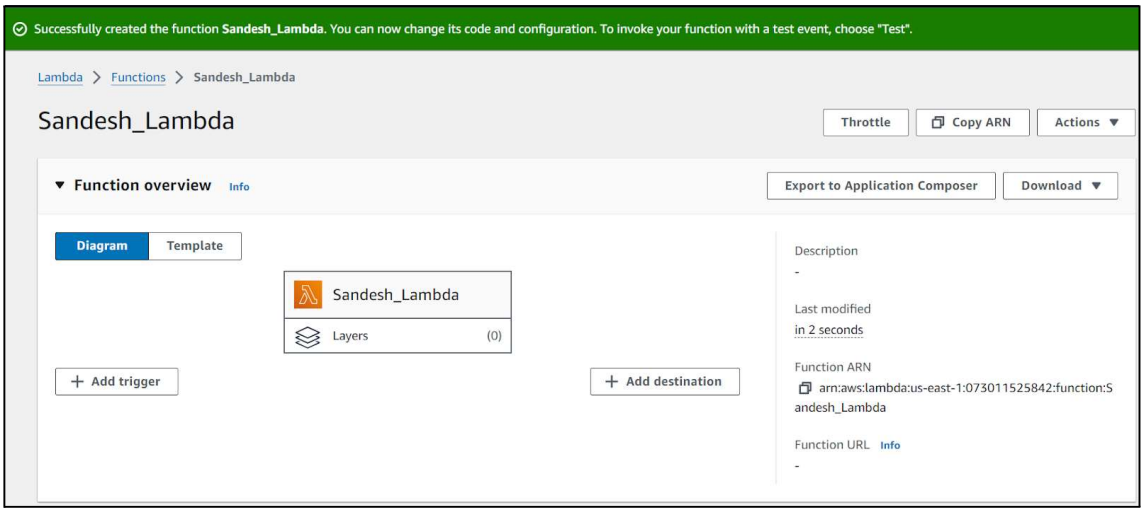


**Step 2** : Give your Lambda function a name and choose a programming language. The code editor only supports Node.js, Python, and Ruby, so in my case I have chosen Python 3.12. Set the architecture to x86. For the execution role, select 'Use an existing role,' then pick 'Lab role' from the dropdown 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)



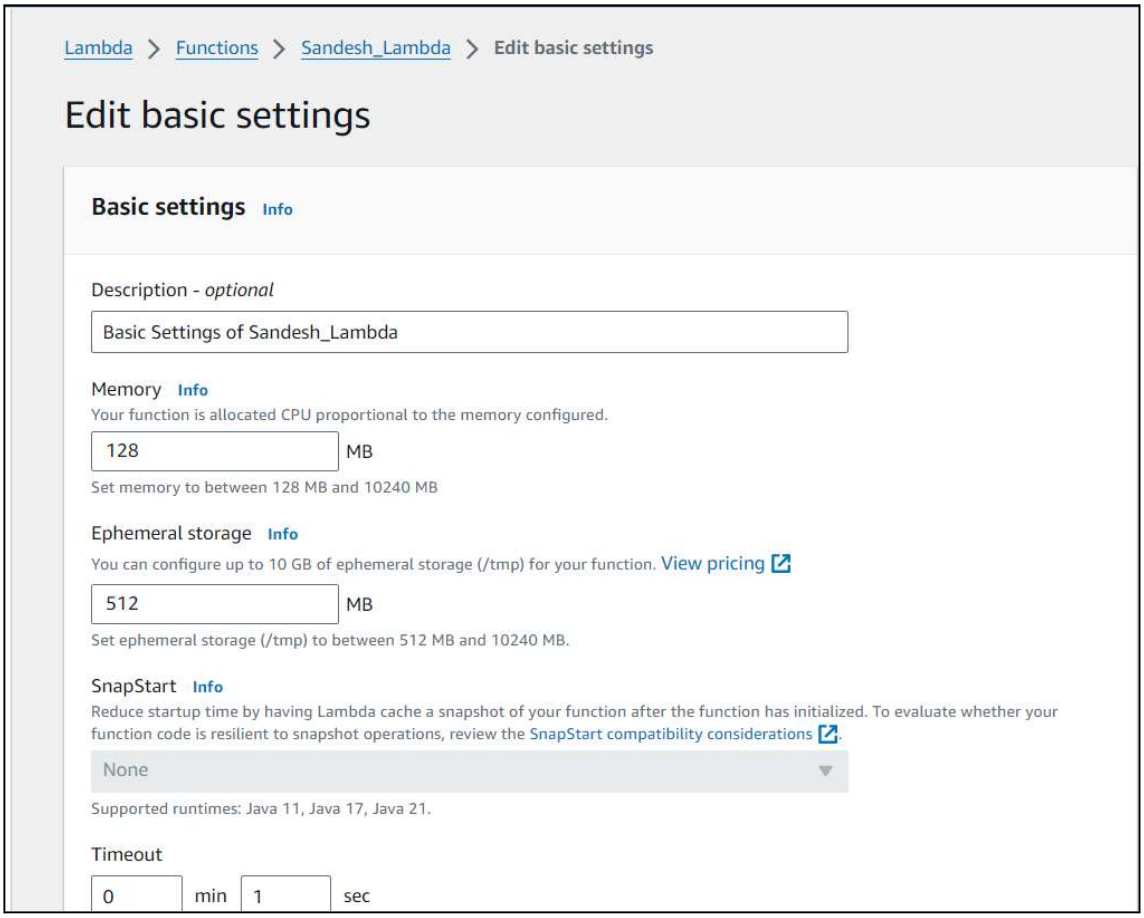


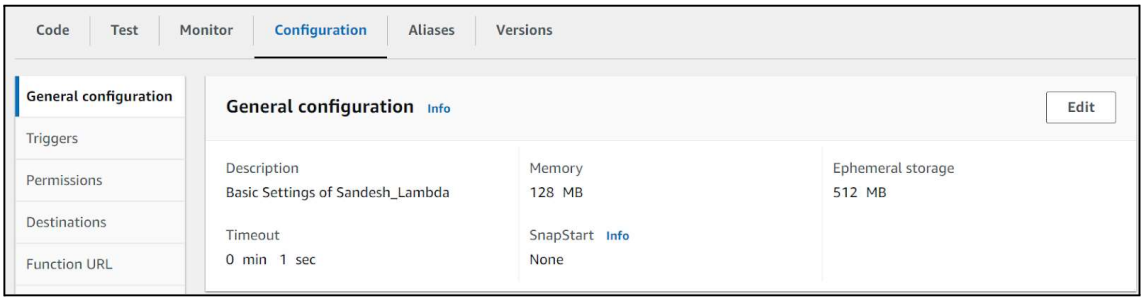




Successfully created Lambda function

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

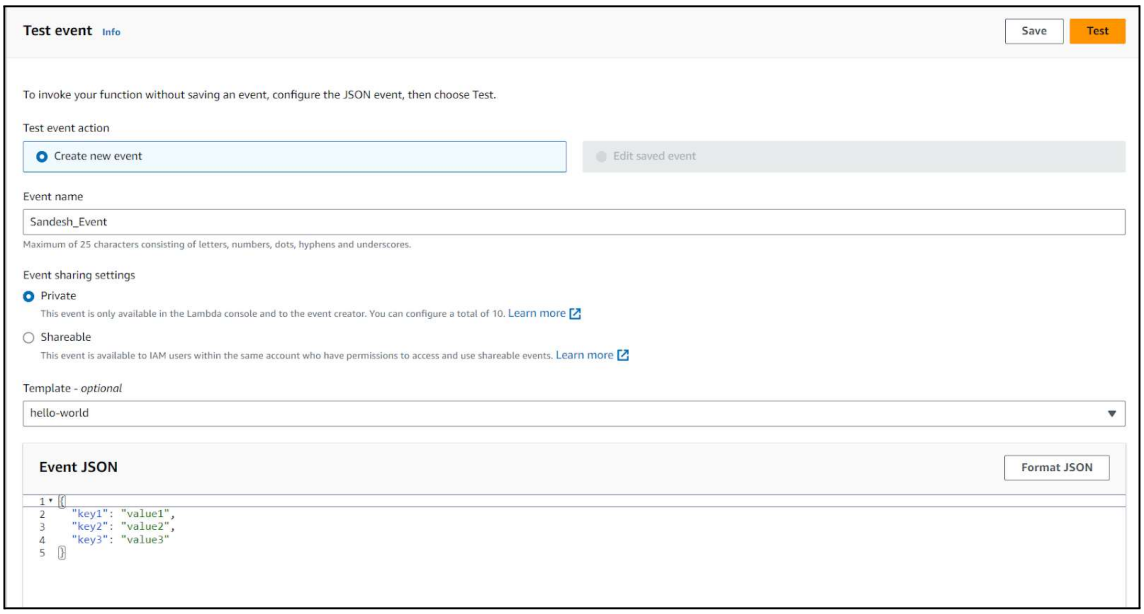

You can add a description and adjust the memory and timeout settings. I've changed the timeout to 1 second, as that's enough for now.



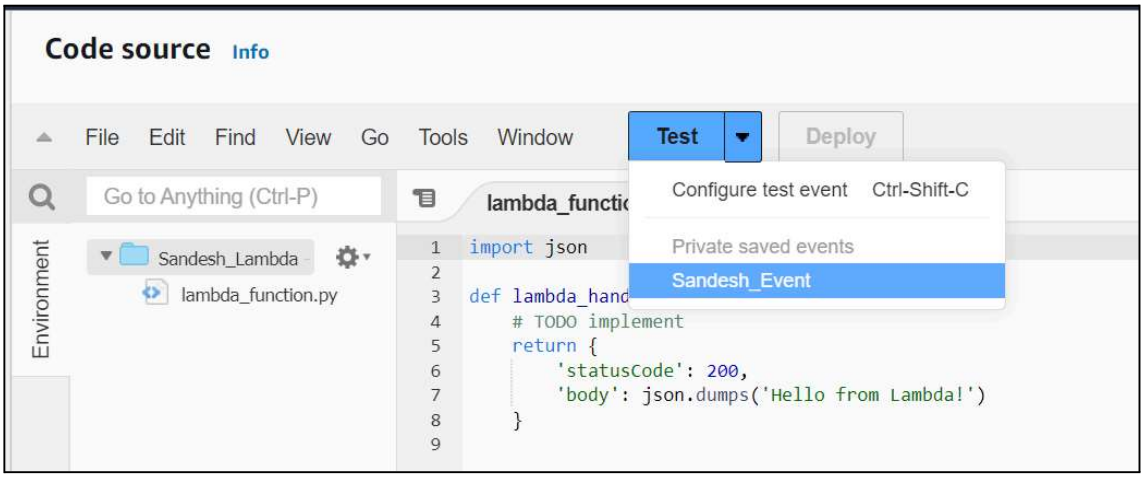



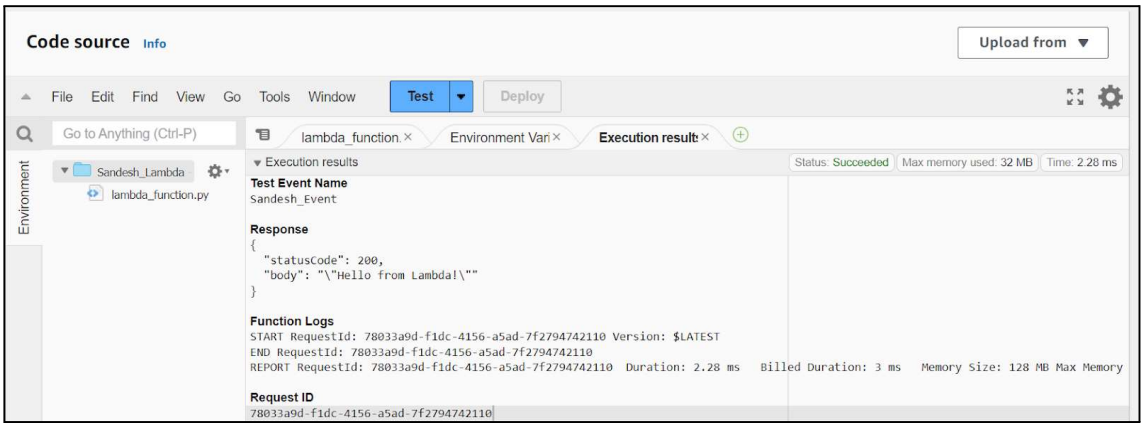
Step 4: Go to the 'Test' tab and click 'Create a new event.' Give the event a name, set 'Event Sharing' to private, and choose the 'hello-world' template.

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.



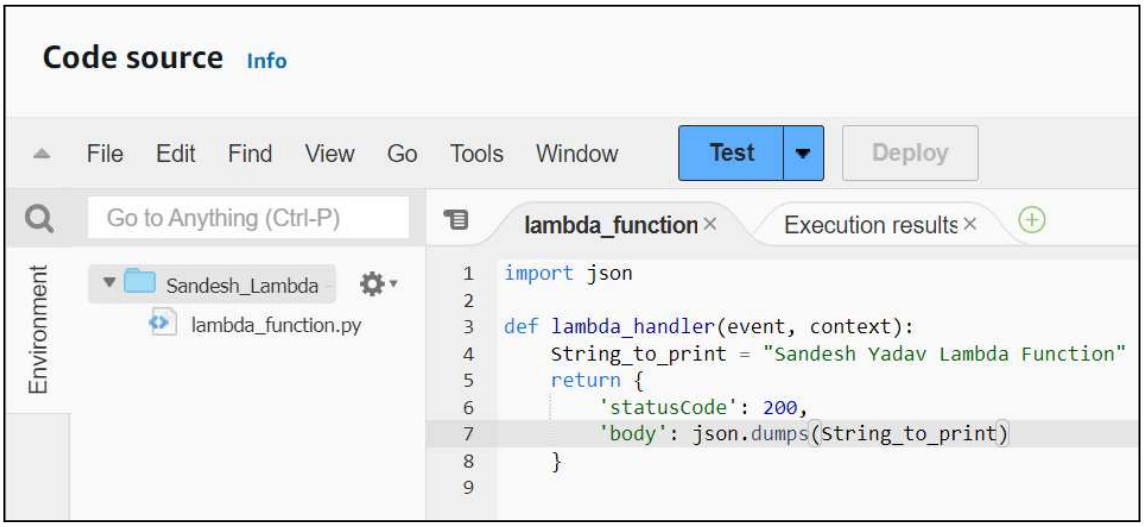

**Step 5**: In the Code section, select the event you created from the dropdown menu under 'Test,' then click 'Test.' You should see the output below."

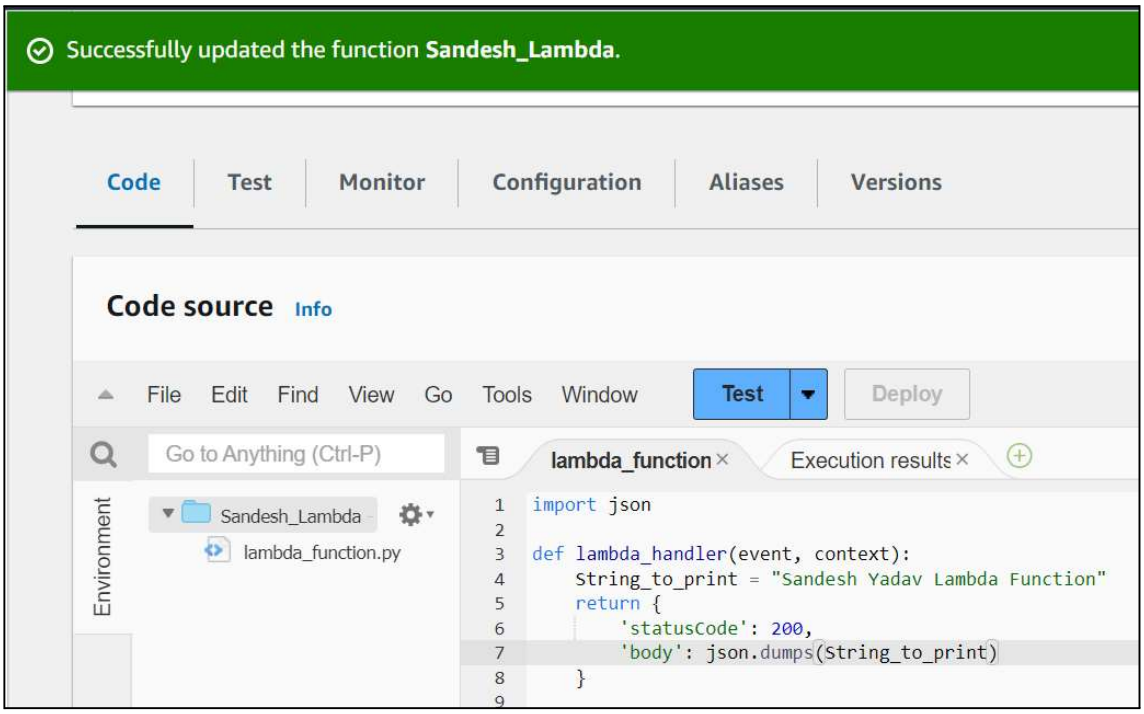





You select the created event to run the specific test you set up, and clicking 'Test' executes your Lambda function to check if it works as expected and produces the desired output

**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






**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 created and deployed an AWS Lambda function using Python. Key steps included setting up the function, adjusting the timeout, creating a test event, and modifying the code. AWS Lambda simplifies development by handling infrastructure, allowing developers to focus on code, while automatically scaling and managing servers.