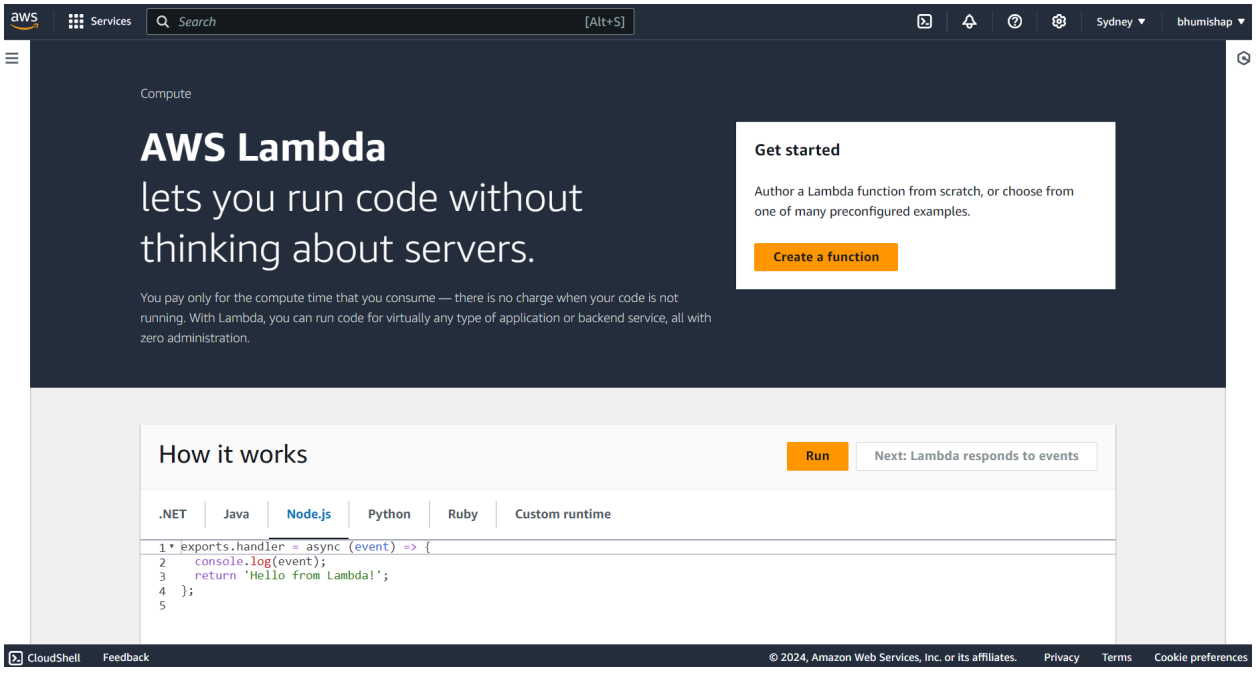
EXPERIMENT NO. 11

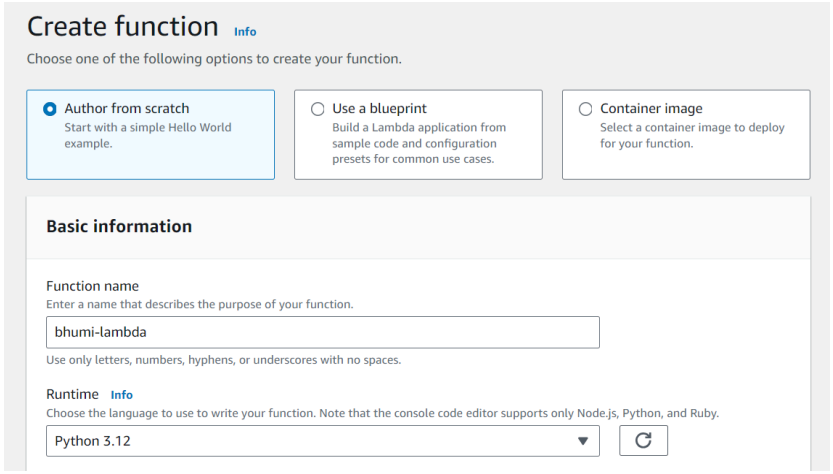
Aim: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

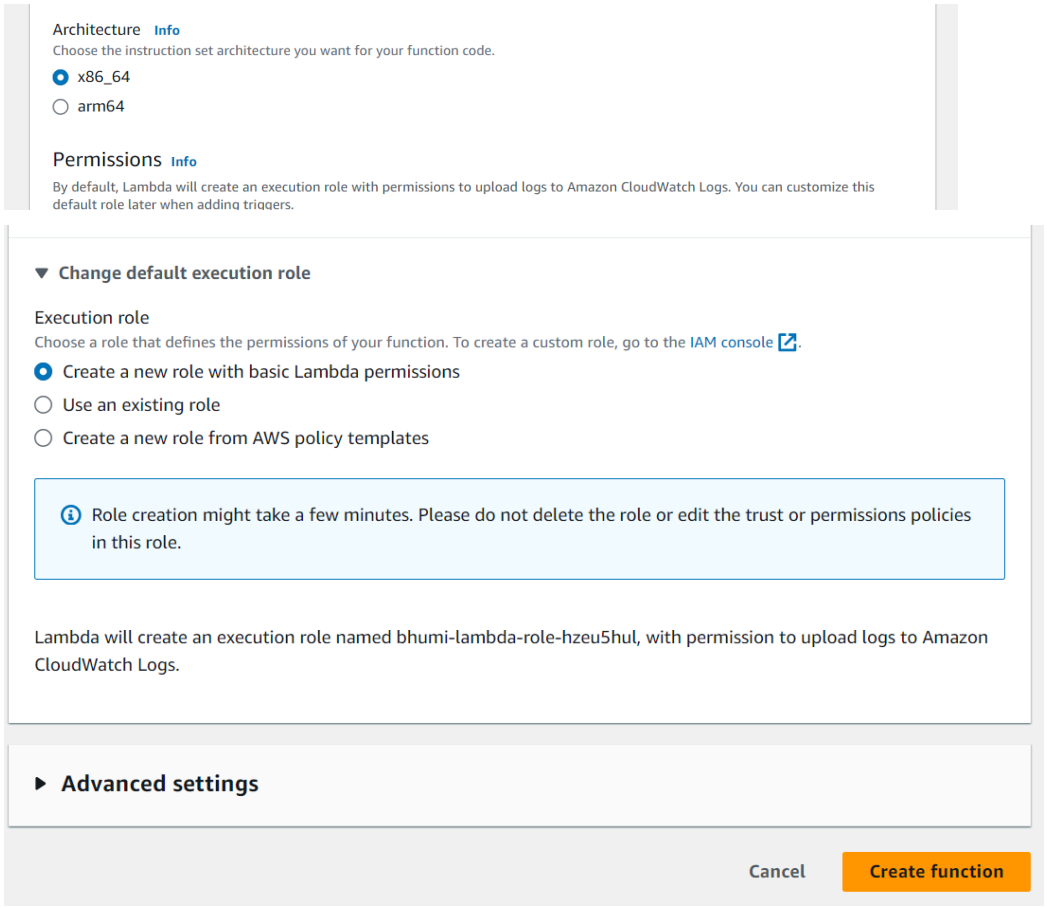
Steps To create the lambda function:

1. Login to your AWS Personal/Academy Accout.Open lambda and click on the create function button.

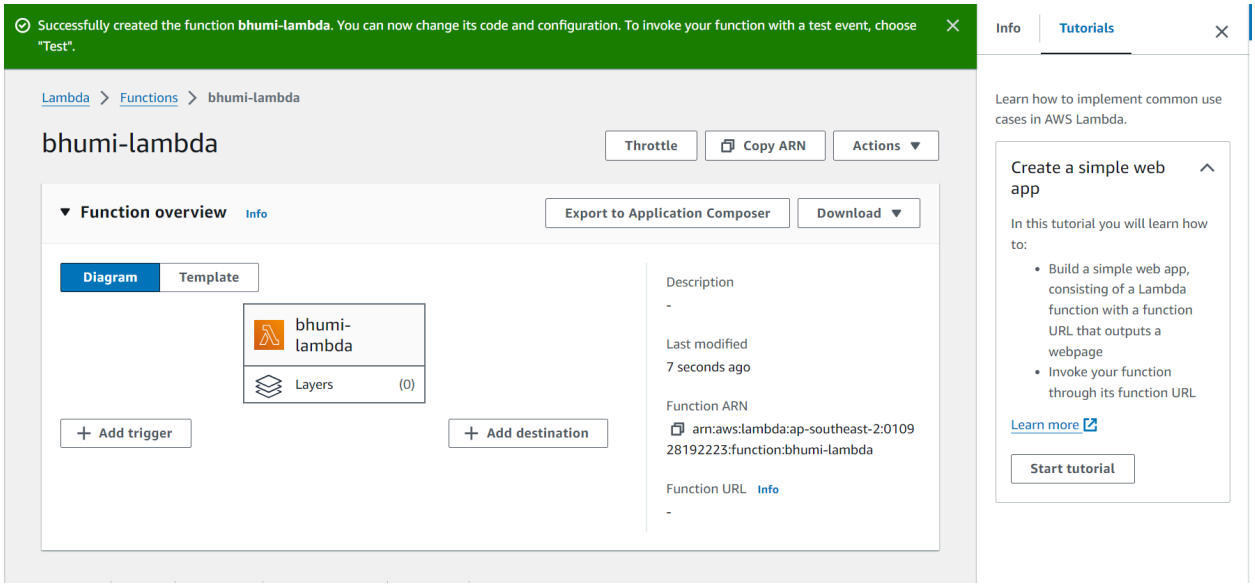


2. Now Give a name to your Lambda function.
   Select the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. So we will select Python 3.12, Architecture as x86, and Execution role to Create a new role with basic Lambda permissions

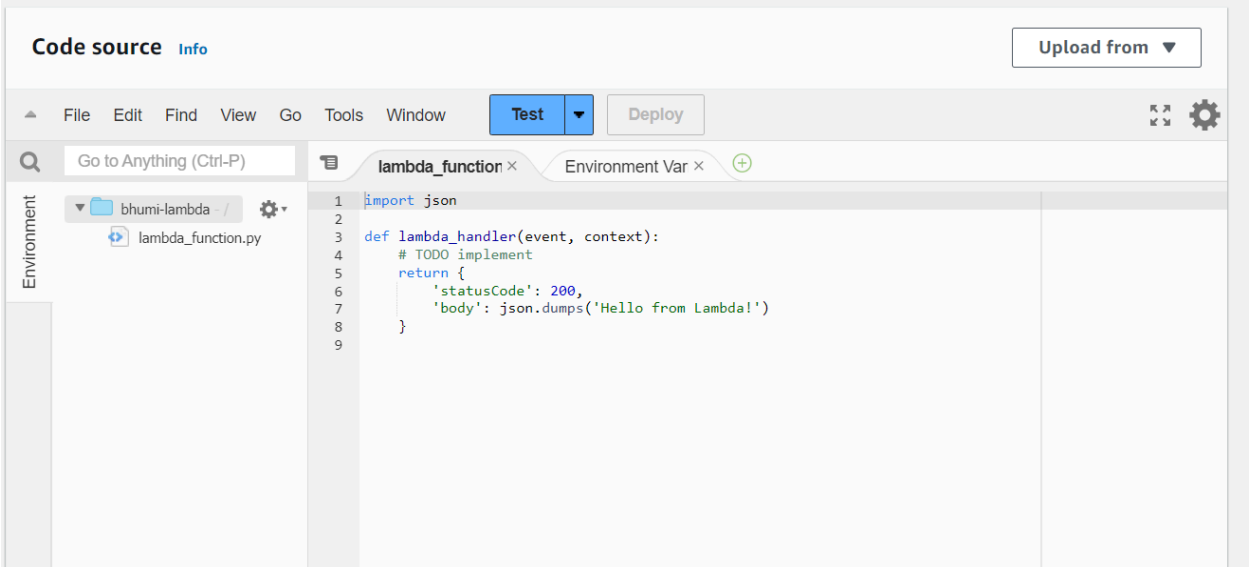




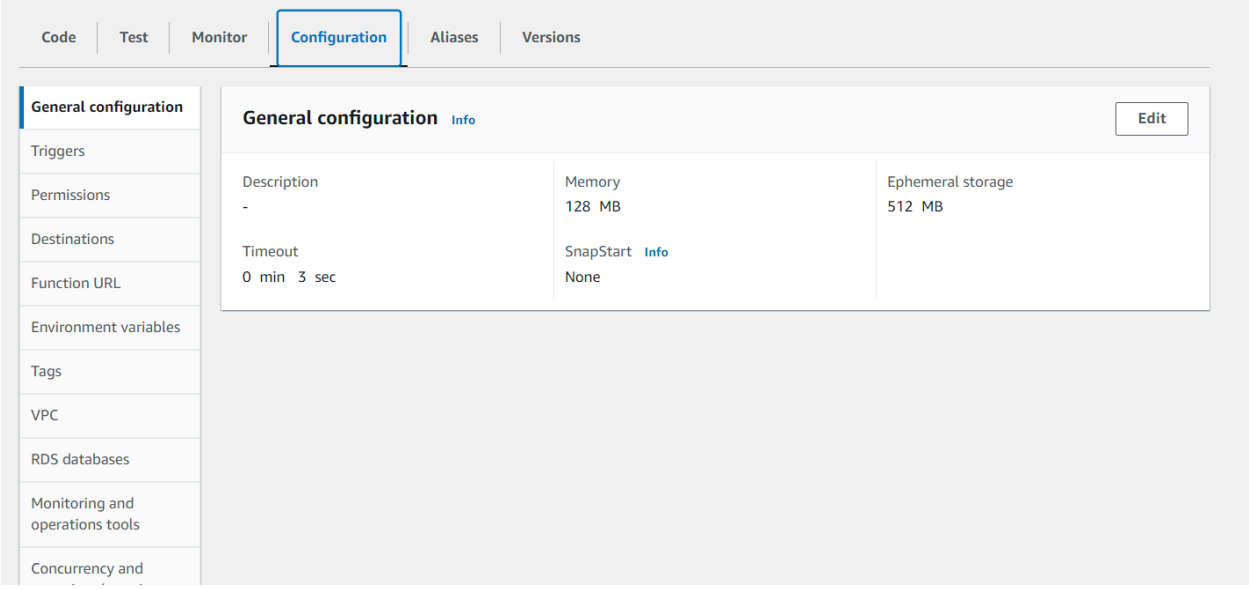
Thus, we have successfully created a lambda function named bhumi-lambda.



This is how the function is displayed.

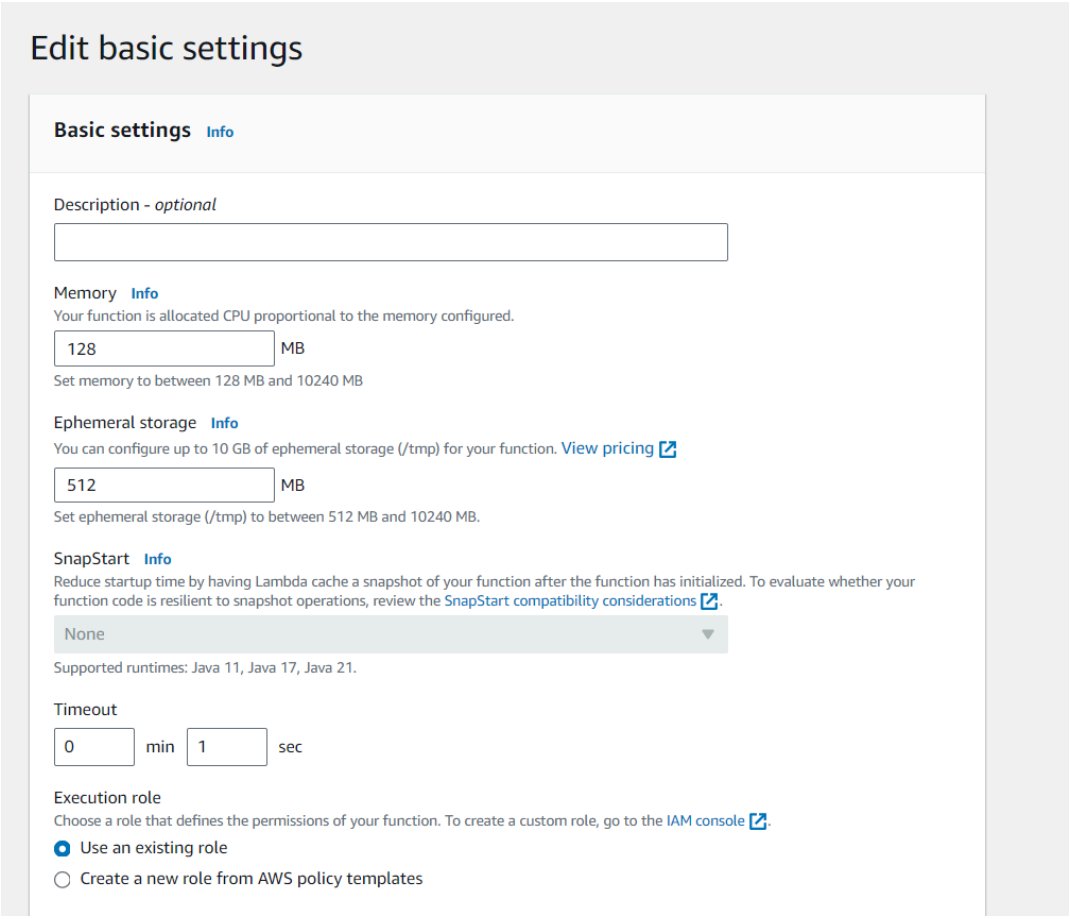


Go to the Configuration tab to see the general configuration of our function.

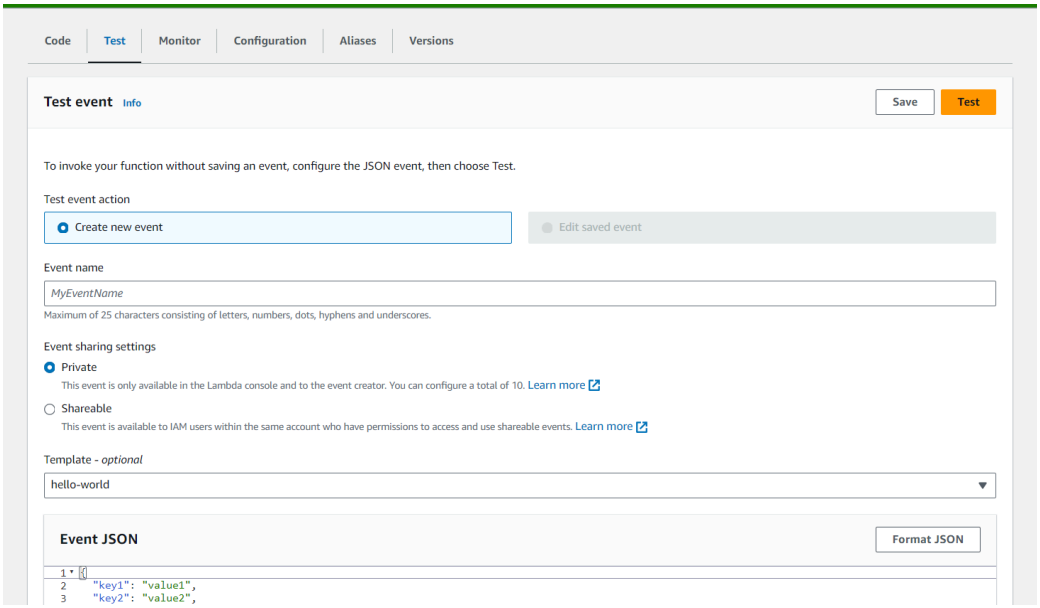


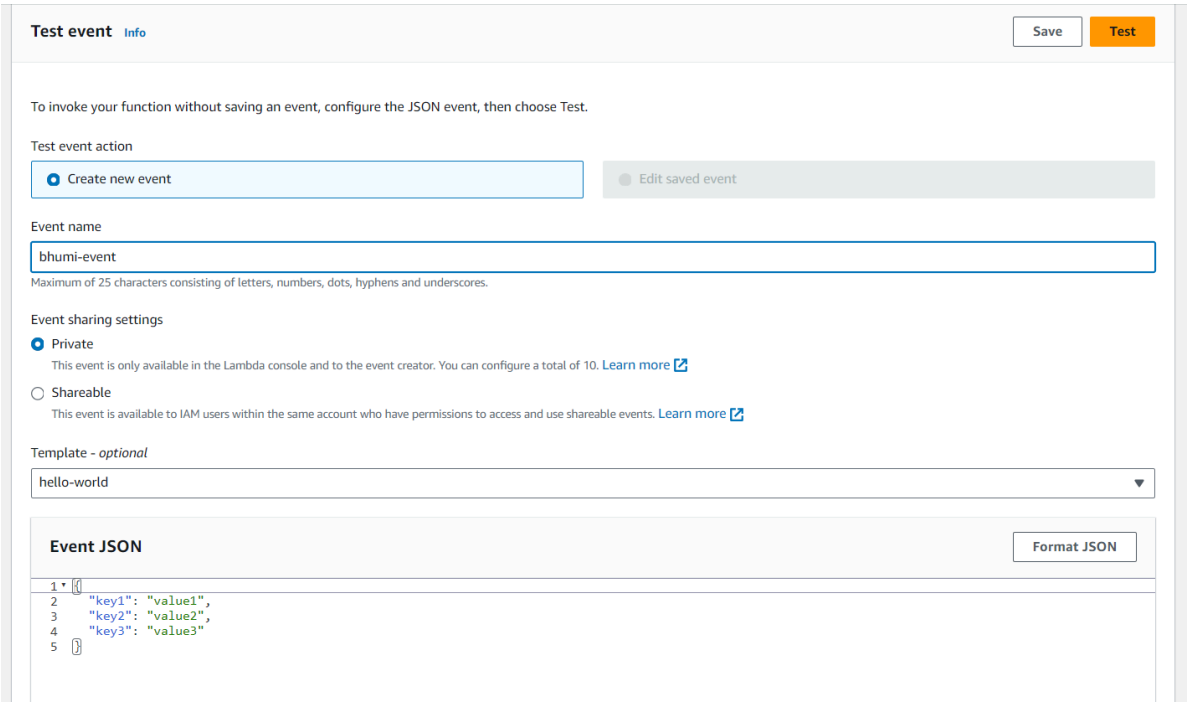
We want to edit the timeout time and the rest can be kept the default.
Here, we can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.

To edit, we go to the Edit button seen in the above screenshot.

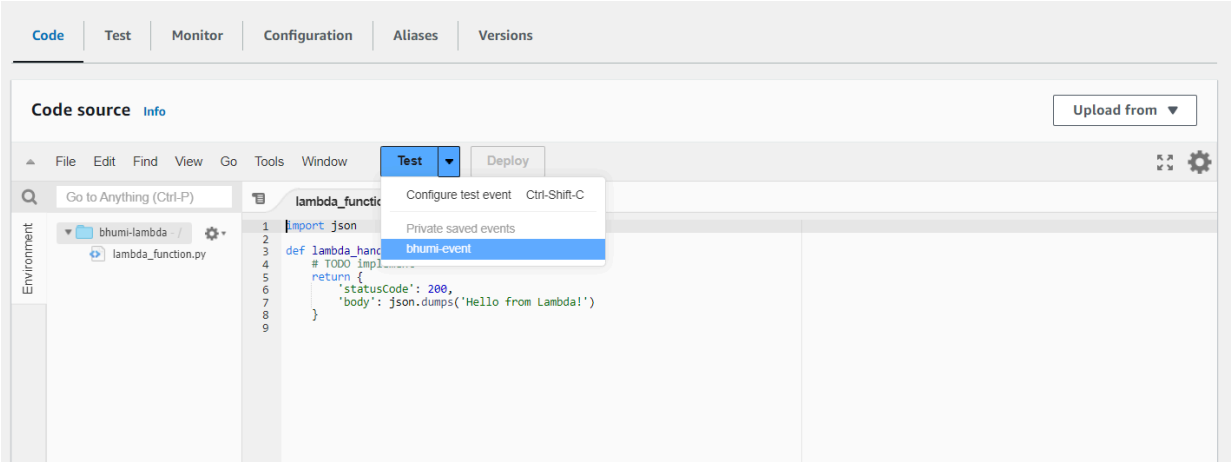


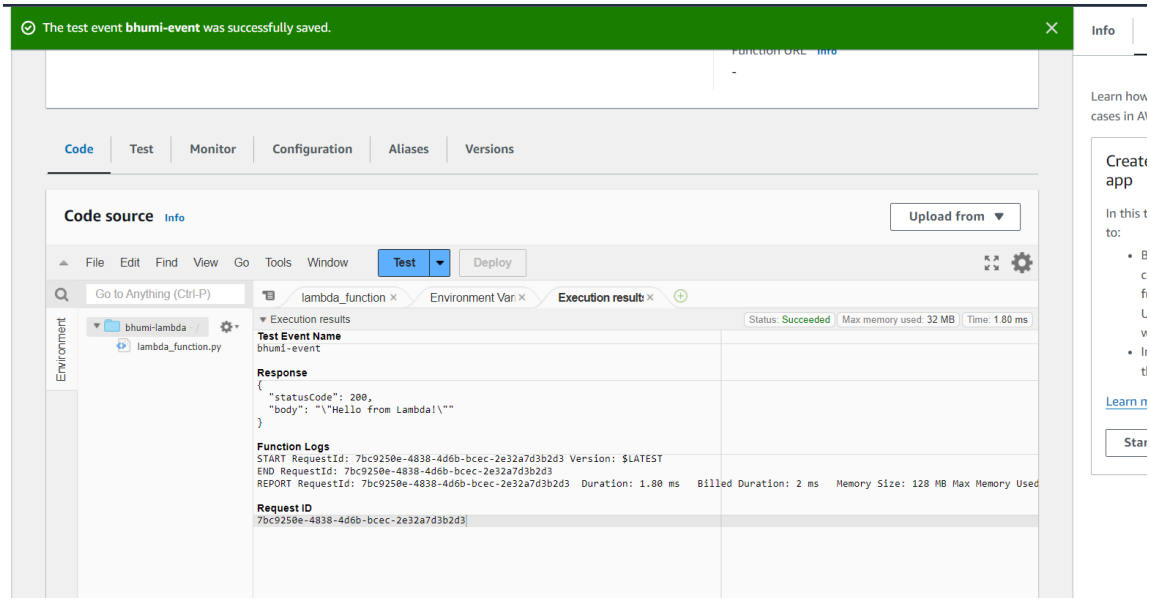
3. Now Click on the Test tab then select *Create a new event,* give a name to the event and select Event Sharing to private and select hello-world template.



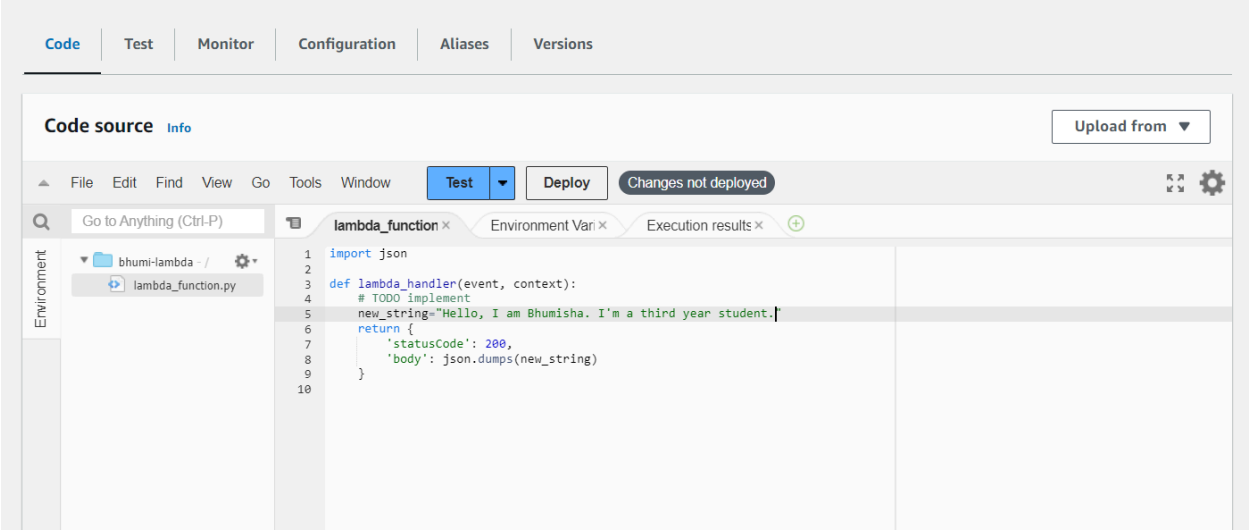


4. Now in the Code section, select the created event from the dropdown of test then click on test . You will see the below output.

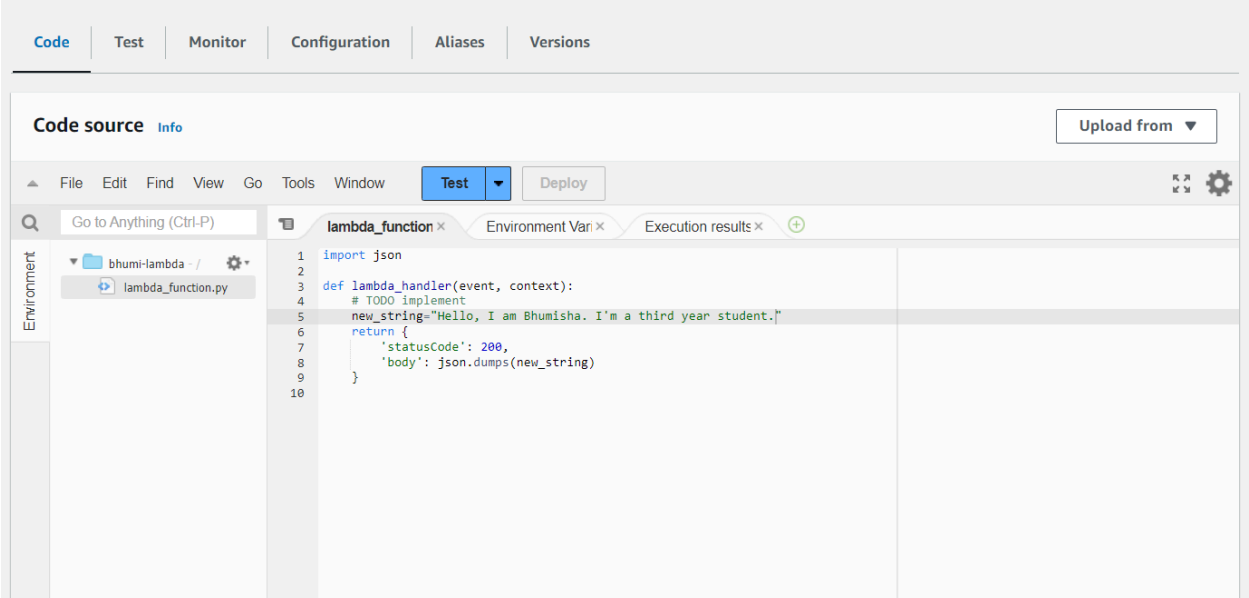




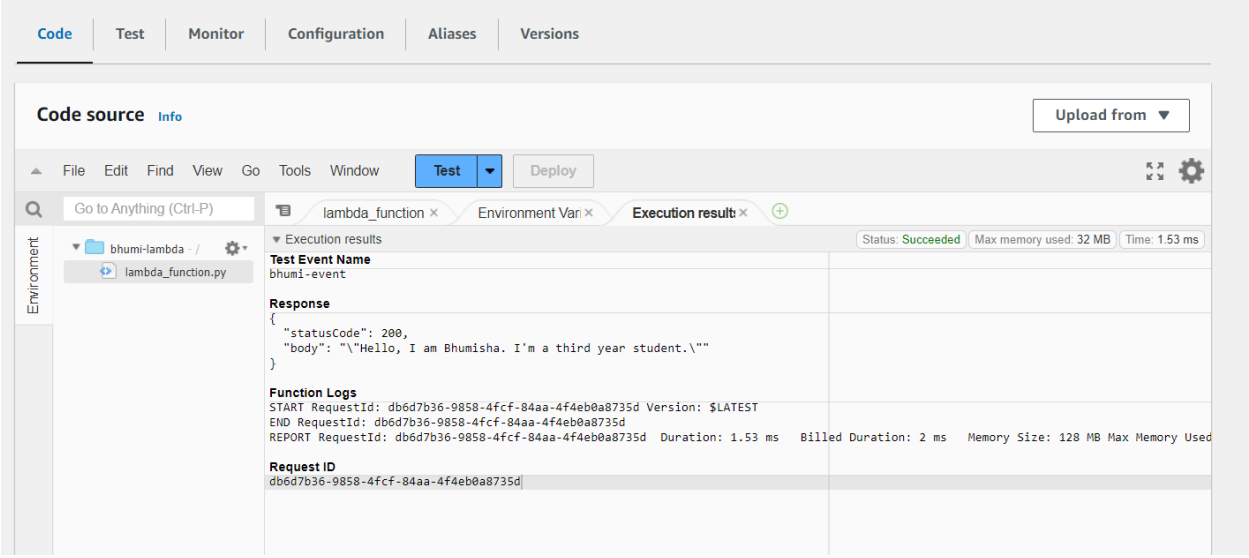
5. You can edit your lambda function code. I have changed the code to display the new string.



Now ctrl+s to save and click on deploy to deploy the changes.



6. Now click on the test and observe the output. We can see the status code 200 and your string output and function logs on successful deployment.



Conclusion: In this experiment, we successfully created an AWS Lambda function and walked through its essential steps. After setting up the function with Python, we configured the basic settings, including adjusting the timeout to 1 second. We then created a test event, deployed the function, and validated the output. Additionally, we modified the Lambda function's code and redeployed it to observe the changes in real-time.

This practical experience demonstrated the simplicity and flexibility of AWS Lambda in creating serverless applications, allowing you to focus on code while AWS manages the infrastructure and scaling