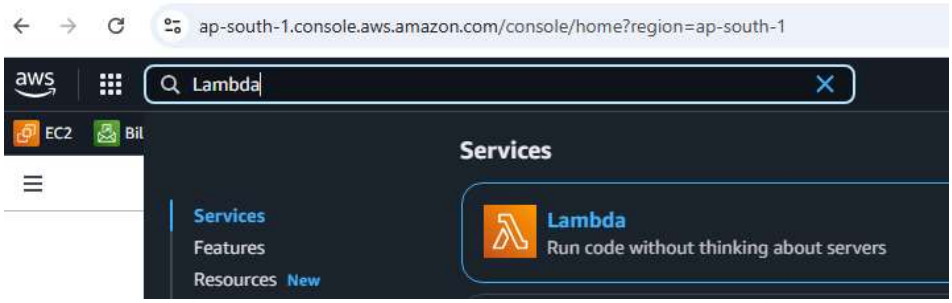
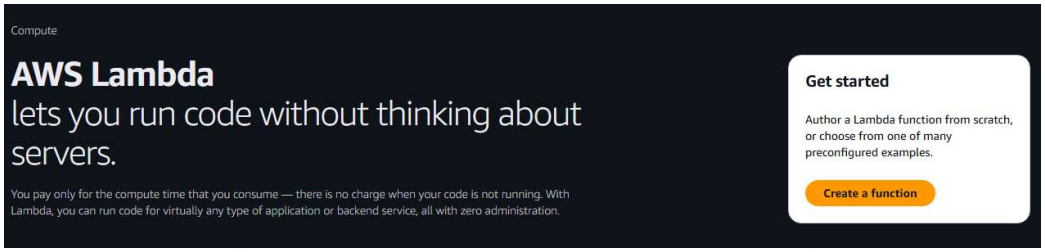
AWS Lambda:

- Run code without provisioning or managing servers, creating workload-aware cluster scaling logic, maintaining event integrations, or managing runtimes.
- Run code for virtually any type of application or backend service. Just upload your code as a ZIP file or container image, and Lambda automatically allocates compute execution power and runs your code based on the incoming request or event, for any scale of traffic.
- Write Lambda functions in your favorite language (Node.js, Python, Go, Java, and more) and use both serverless and container tools, such as AWS SAM or Docker CLI, to build, test, and deploy your functions.

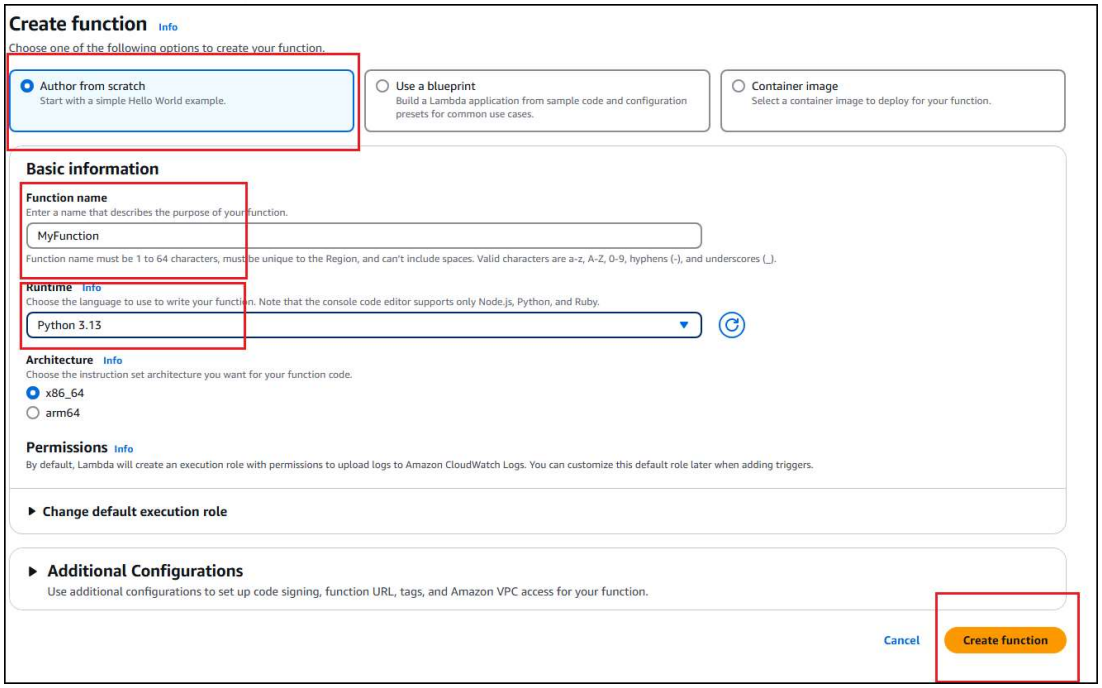
1. Login to **AWS console** and search for **Lambda**.



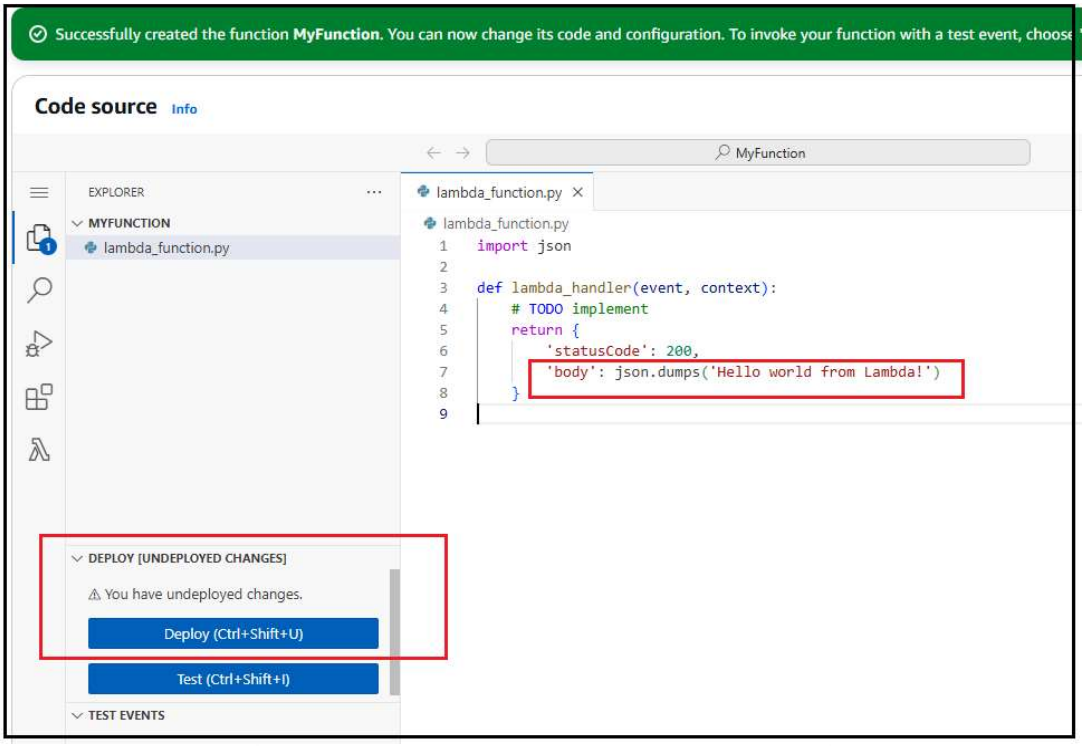
2. Click on **Create Function**.



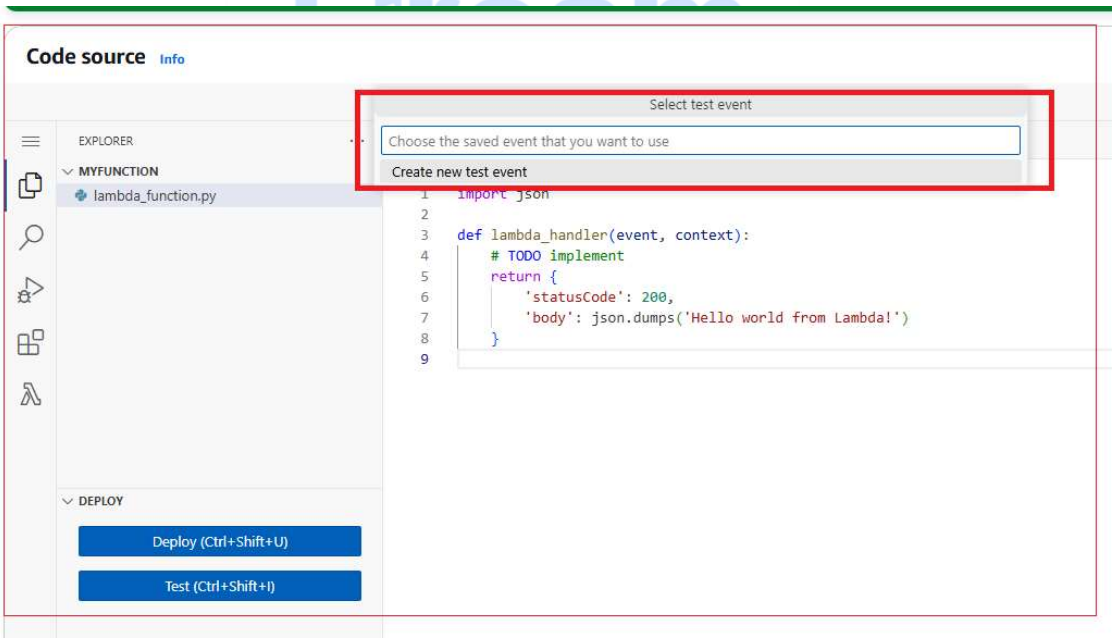
3. Give a name to the function, and Choose **Python for Runtime** and click on **Create function**.



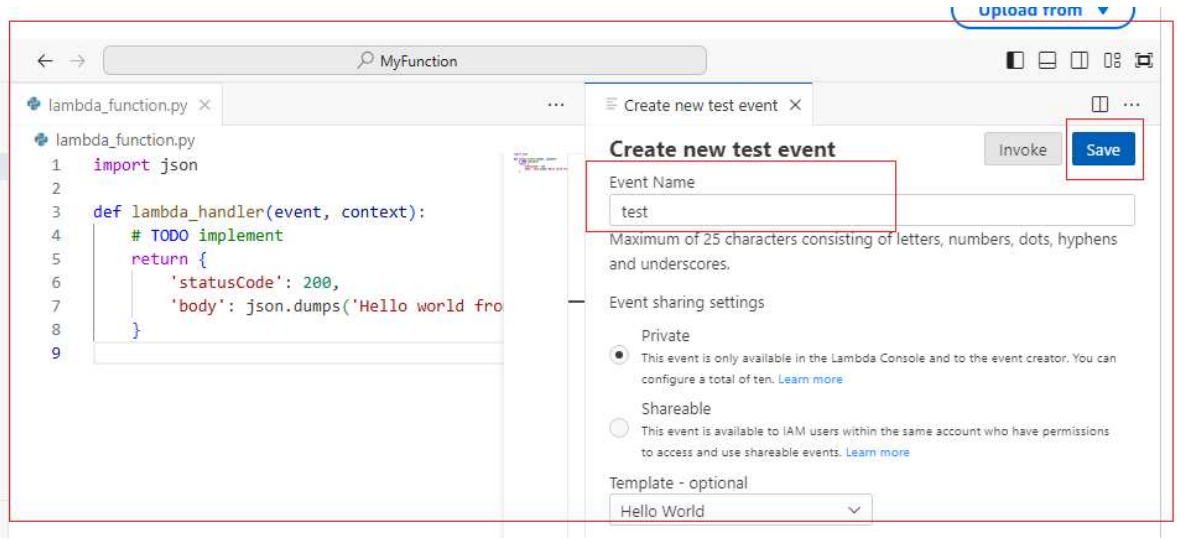
4. Change some content on the body section and Click on **Deploy**.



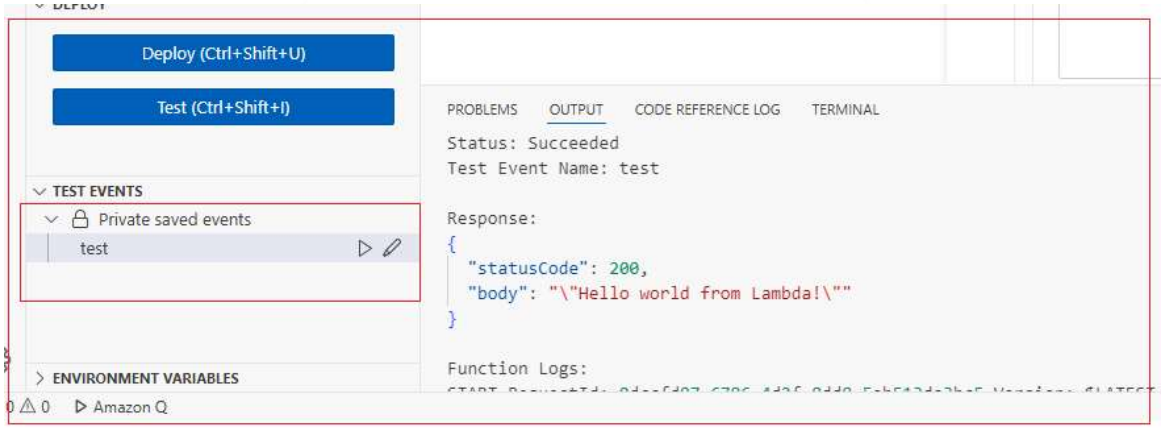
5. Click on the Test button, click on new test event.



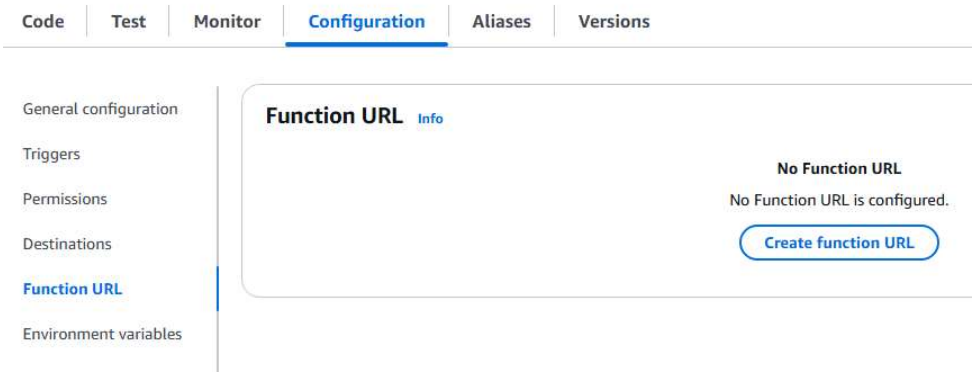
6. Give the test event a Name and click on Save.



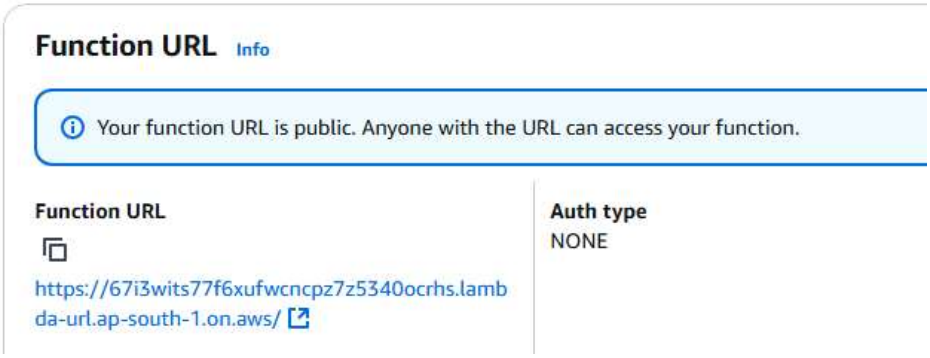
7. Click the test button again and you should get the response.



8. Click Configuration tab and click on Function URL.



9. Click on Create function URL, select Auth type as NONE and save.
10. Now you can invoke this Lambda function using the function URL.

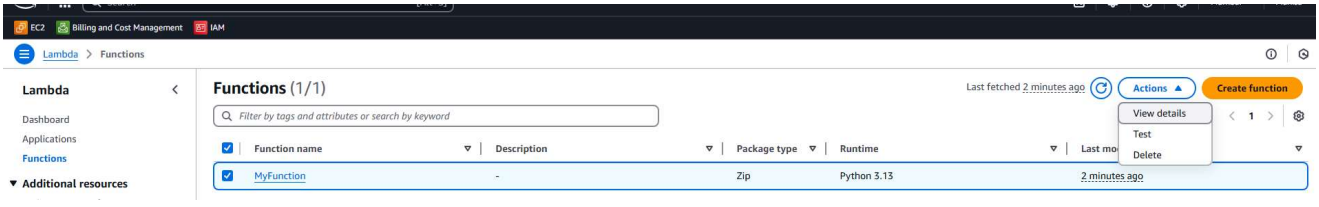


Example:

```
def lambda_handler(event, context):
    return {
        'statusCode': 200,
        'body': '<h1>Welcome</h1>',
        'headers' : {
            'Content-type': 'text/html'
        }
    }
```

It should show the output as:

Delete a Lambda Function:



Cloudwatch:

Amazon CloudWatch is a service that monitors applications, responds to performance changes, optimizes resource use, and provides insights into operational health. By collecting data across AWS resources, CloudWatch gives visibility into system-wide performance and allows users to set alarms, automatically react to changes, and gain a unified view of operational health.