**Serverless computing:**

1. Serverless computing allows you to build and run applications and services without thinking about servers.

2. With serverless computing, your application still runs on servers, but all the server management is done by AWS.

**AWS Lambda:**

1. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging.

1. With Lambda, all you need to do is supply your code in one of the language runtimes that Lambda supports.
2. You organize your code into Lambda functions.
3. The Lambda service runs your function only when needed and scales automatically.
4. You only pay for the compute time that you consume—there is no charge when your code is not running.

**When to use Lambda**

* **File processing:** Use Amazon Simple Storage Service (Amazon S3) to trigger Lambda data processing in real time after an upload.
* **Stream processing:** Use Lambda and Amazon Kinesis to process real-time streaming data for application activity tracking, transaction order processing, clickstream analysis, data cleansing, log filtering, indexing, social media analysis, Internet of Things (IoT) device data telemetry, and metering.
* **Web applications:** Combine Lambda with other AWS services to build powerful web applications that automatically scale up and down and run in a highly available configuration across multiple data centers.
* **IoT backends:** Build serverless backends using Lambda to handle web, mobile, IoT, and third-party API requests.
* **Mobile backends:** Build backends using Lambda and Amazon API Gateway to authenticate and process API requests. Use AWS Amplify to easily integrate with your iOS, Android, Web, and React Native frontends.

When using Lambda, you are responsible only for your code. Lambda manages the compute configuration that offers a balance of memory, CPU, network, and other resources to run your code.

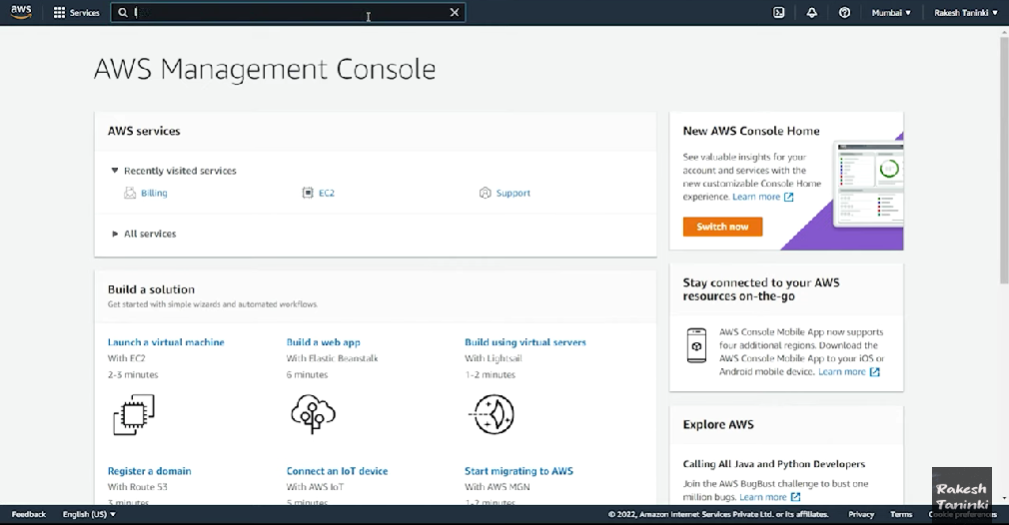
**Getting Started with Lambda:**

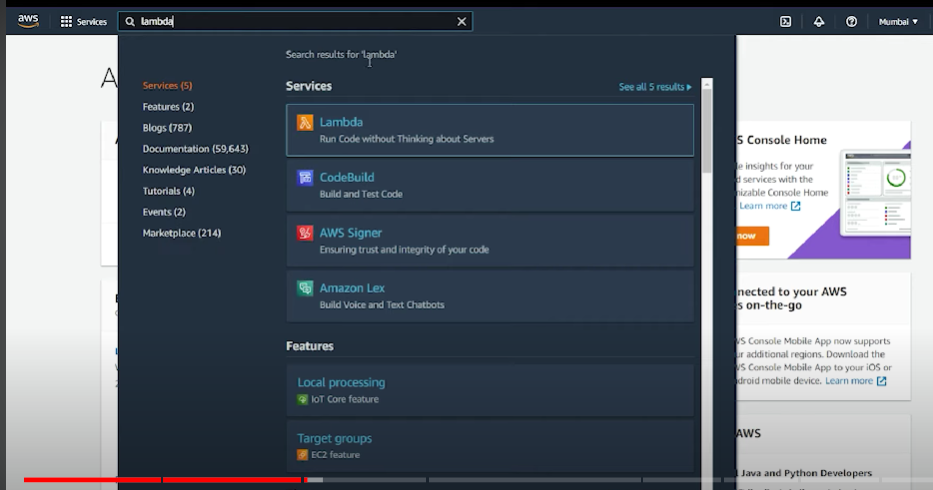
To get started with Lambda, use the Lambda console to create a function. In a few minutes, you can create and deploy a function, invoke it, and then view logs and metrics.

**Create a lambda function with in the AWS console:**

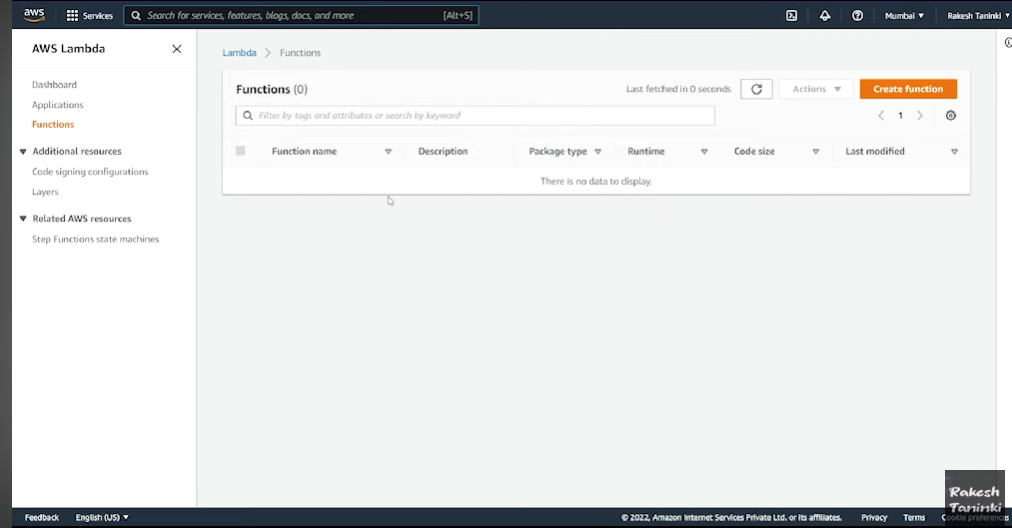
* create a Lambda function.
* If you select a blueprint, then it provides sample code to do some minimal processing.
* Most blueprints process events from specific event sources, such as Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, or a custom application.

**To create a Lambda function with the console**





Click on functions, now you can see you don’t have any functions right now

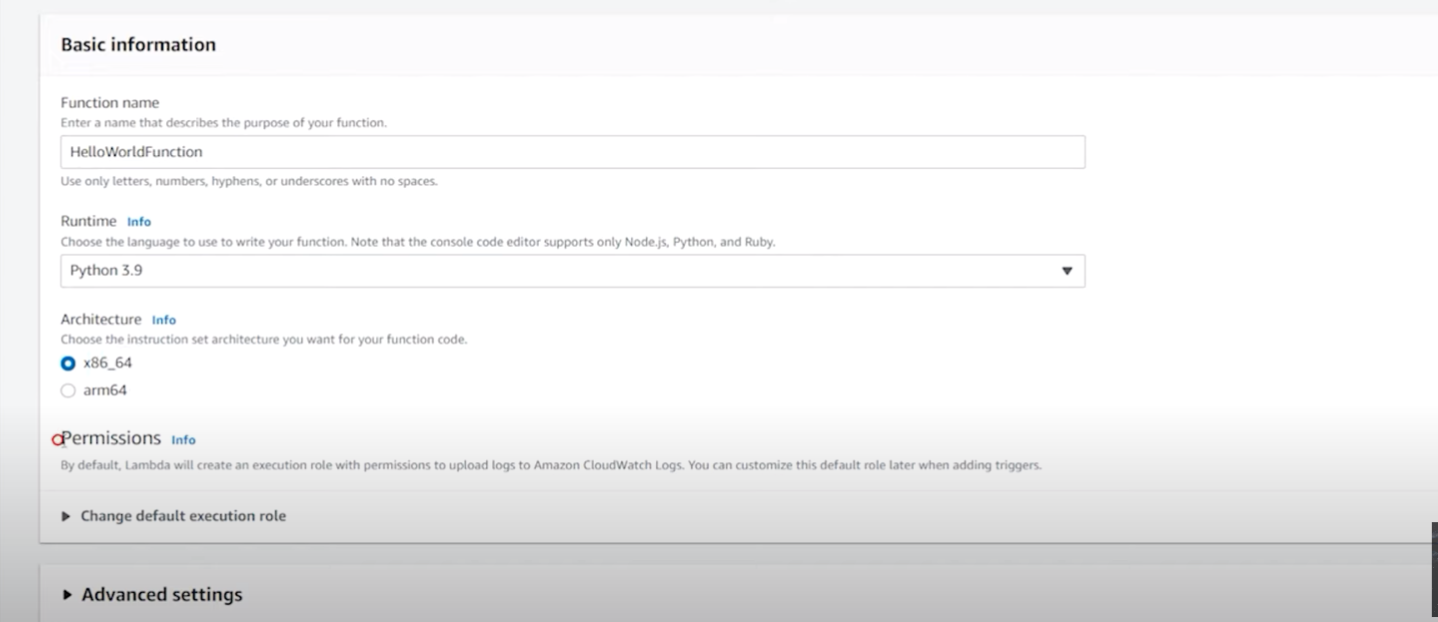


1. Now, click on create function to create a new lambda function



You will be navigated to this page. You can select any of the options to create your function.

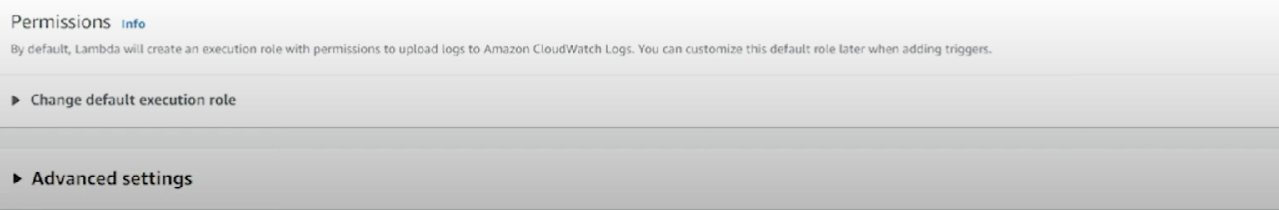
1. Provide function name for lambda
2. Select the language to write the code
3. Select the architecture

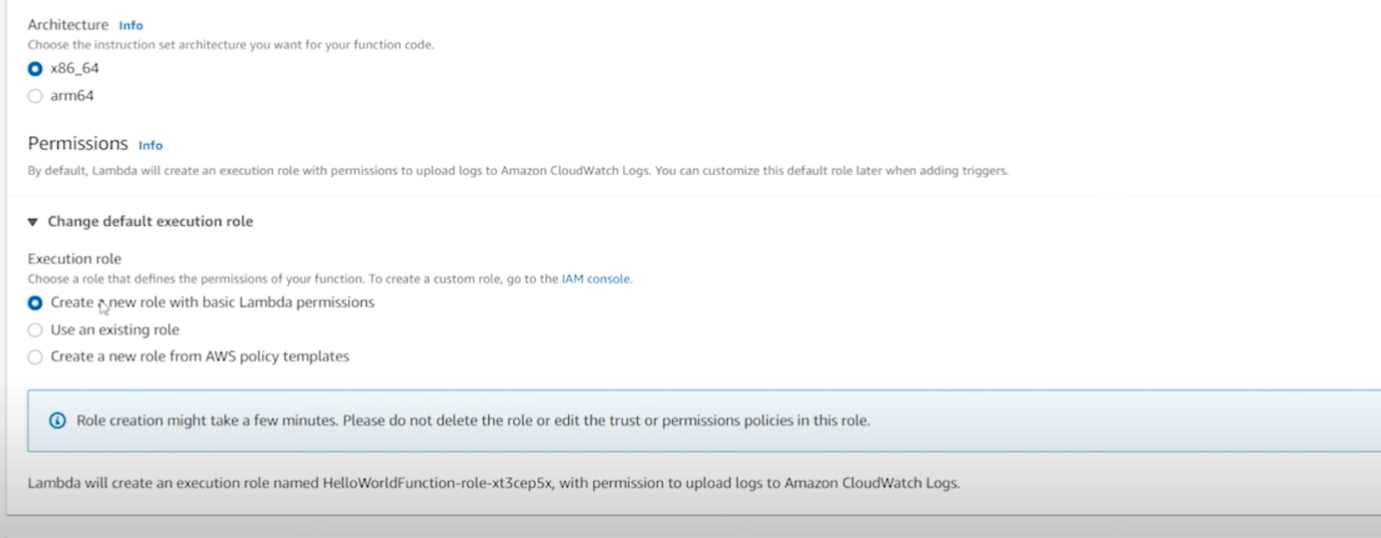


Permission, here we need to mention IAM role.

Because, when we run the lambda function, the code gets executed and some logs are generated and stored in cloud watch Services.

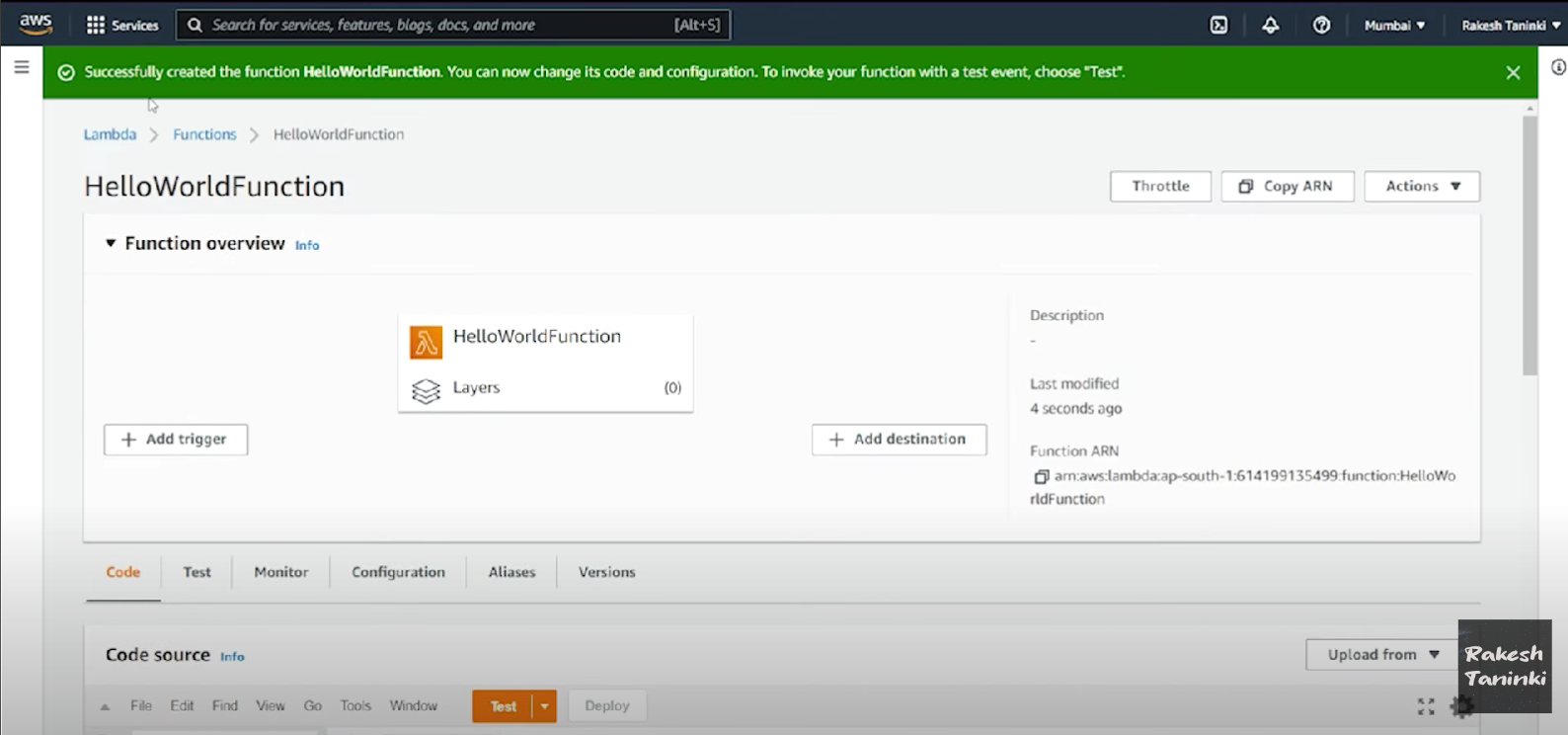
For **Execution role**, choose **Create a new role with basic Lambda permissions**. Lambda creates an [execution role](https://docs.aws.amazon.com/lambda/latest/dg/lambda-intro-execution-role.html) that grants the function permission to upload logs to Amazon CloudWatch. The Lambda function assumes the execution role when you invoke your function, and uses the execution role to create credentials for the AWS SDK and to read data from event sources.



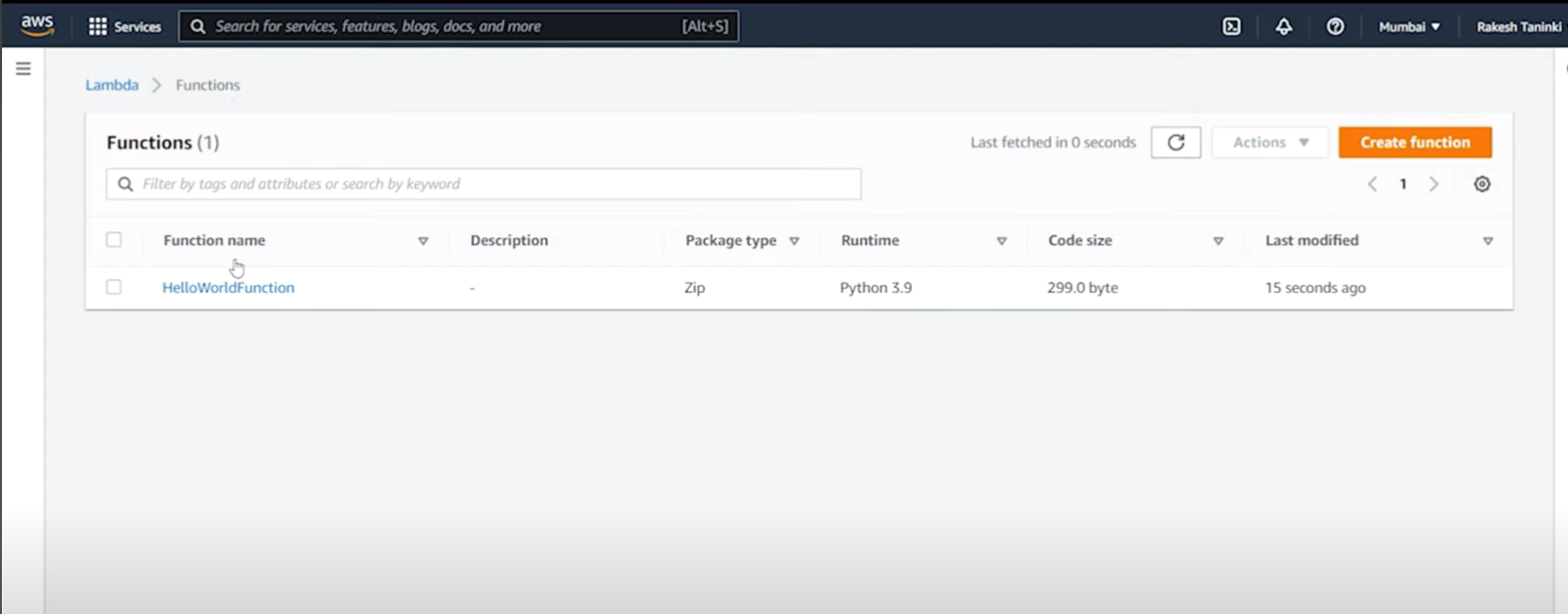


Now click on create function

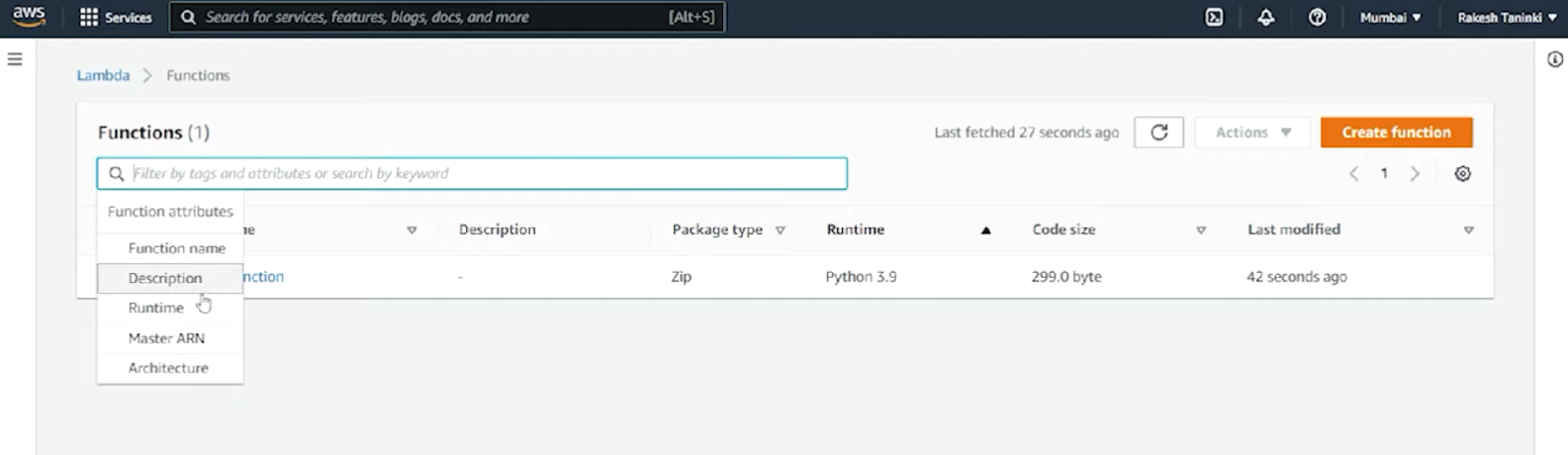


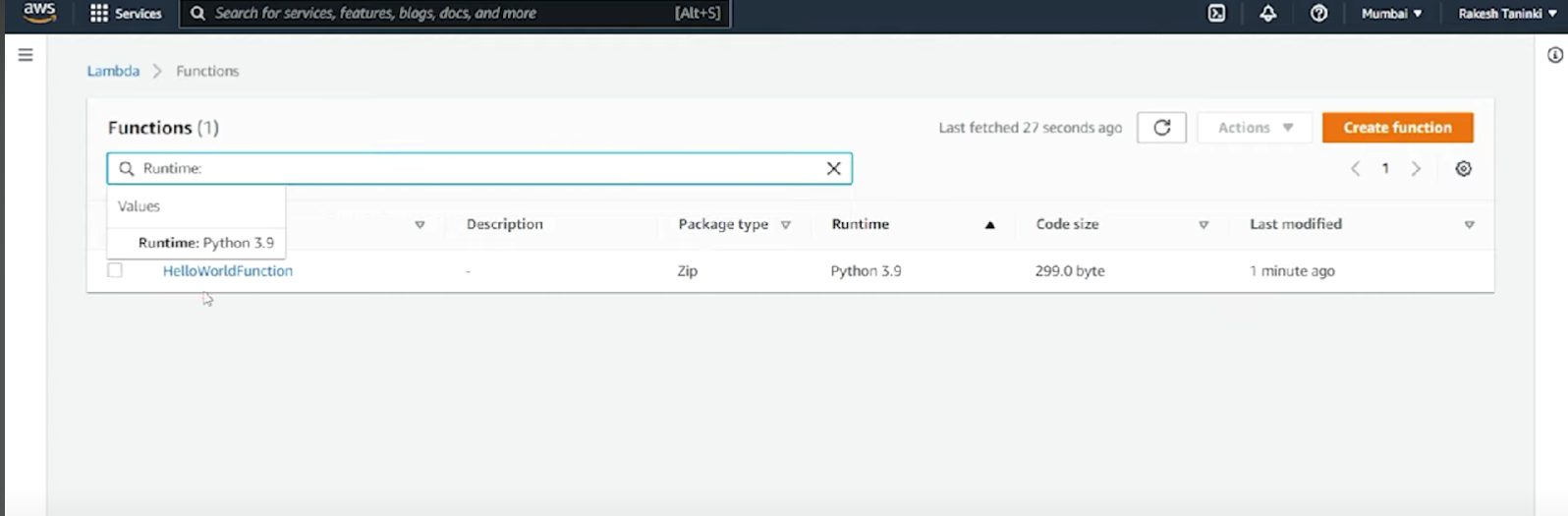


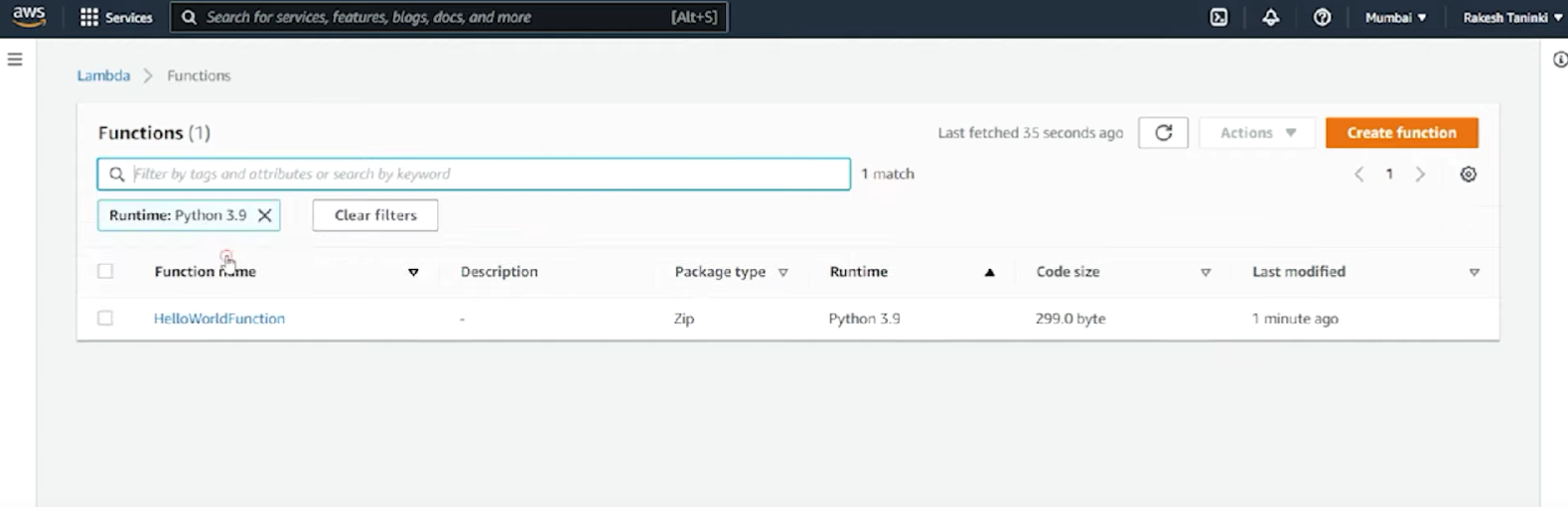
Now if u go back to functions , then u can see the HelloWorldFunction.



Suppose you want to search functions which are written in certain language, then u can search it by using Runtime.









U can see the function overview, and navigate to the tab section.

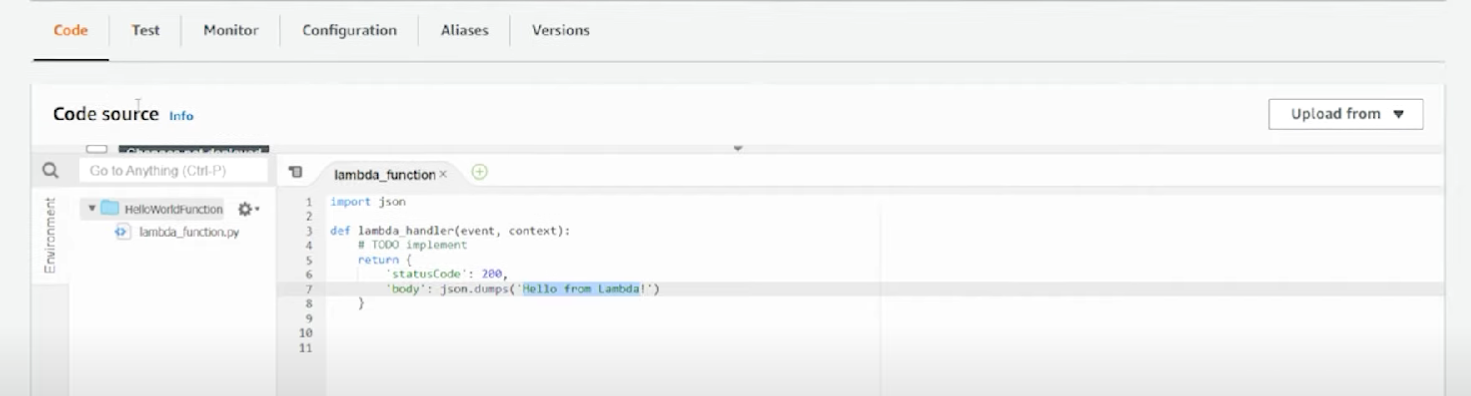
So, here we have 6 tabs

1. Code, Test, Monitor, Configuration, Aliases, Versions

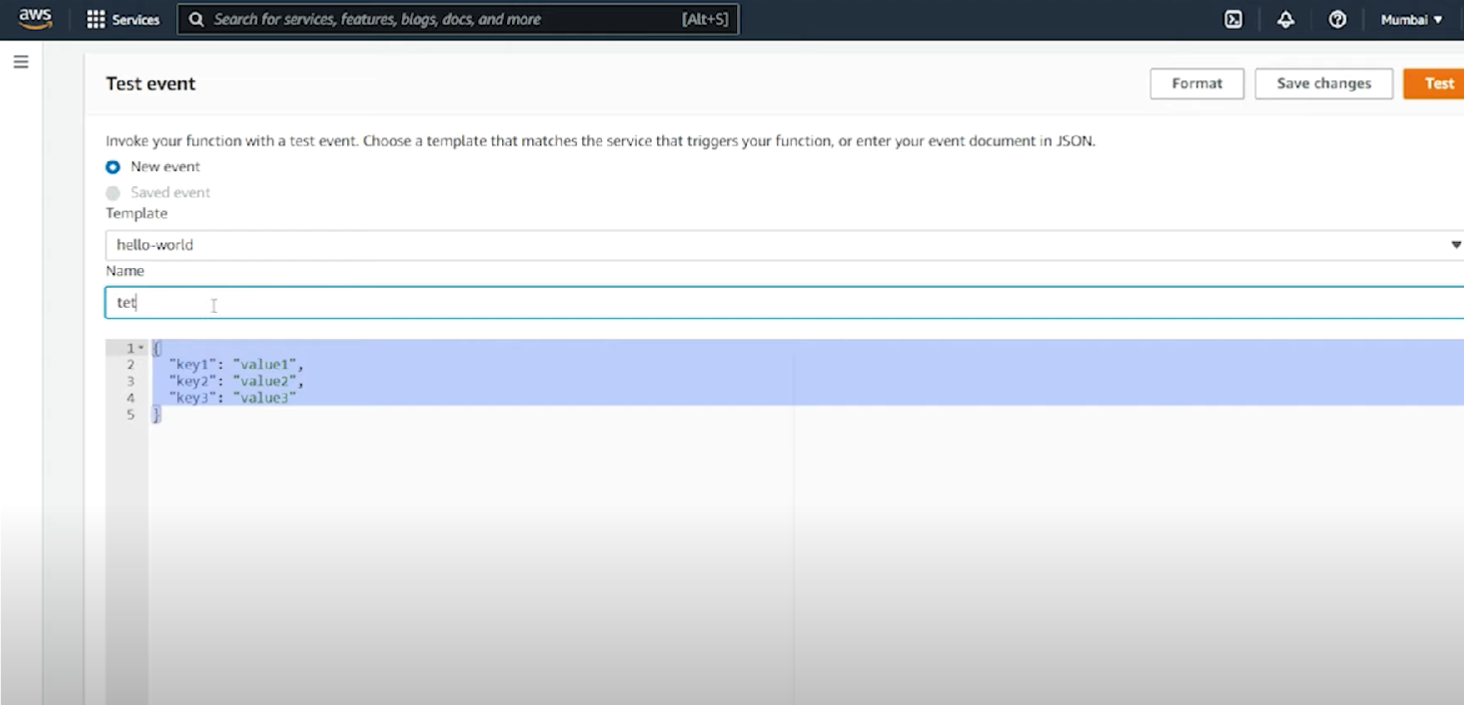
Code:

Here you can write code, or u can convert the existing code into deployment package and upload it.

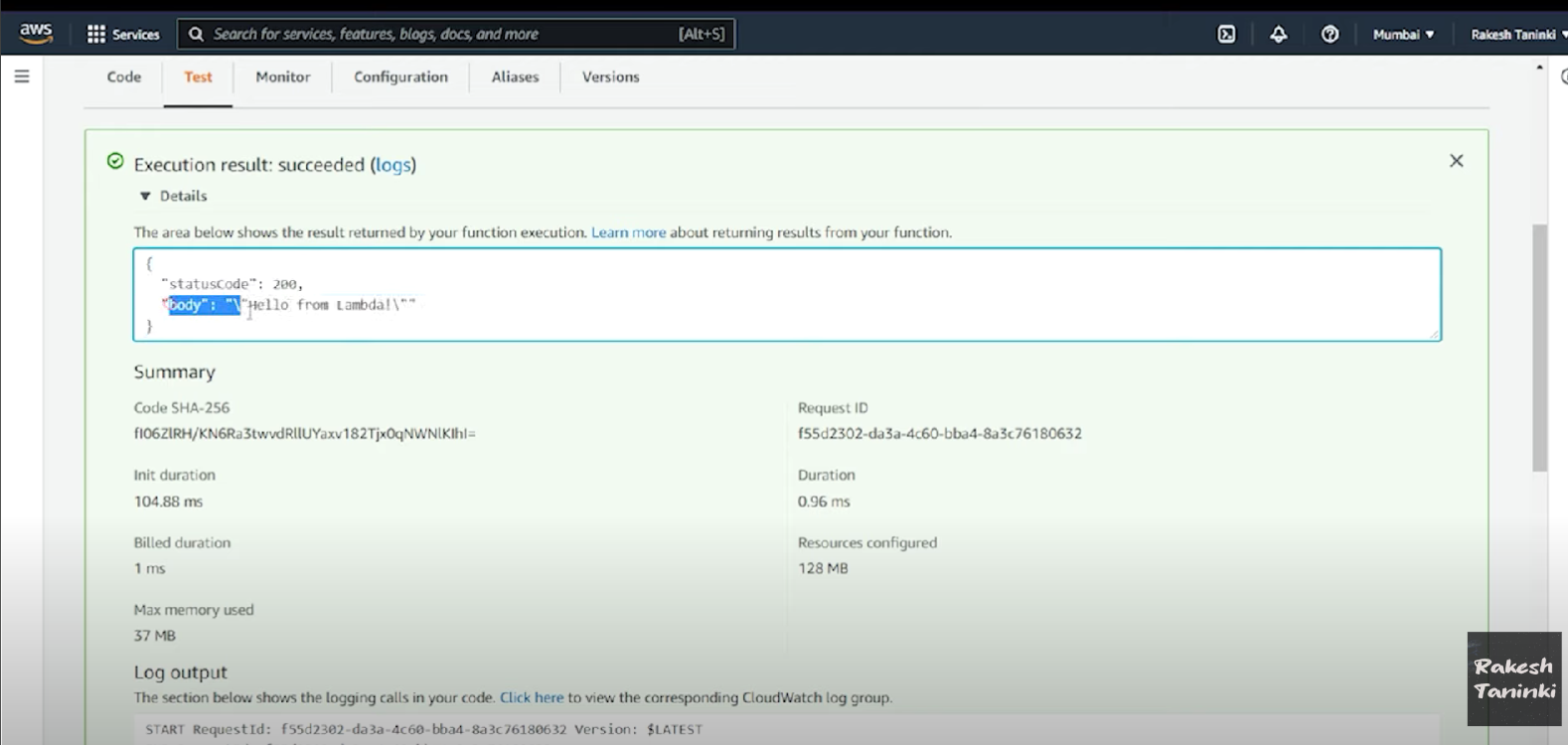
Or you can also do copy paste.



* To execute this function. We have to move to Test area.
* Lambda functions are event base.
* If some event occurs then only our lambda will execute.
* So, for every lambda we need to pass an event.



And click on Test you can see the execution result and logs.

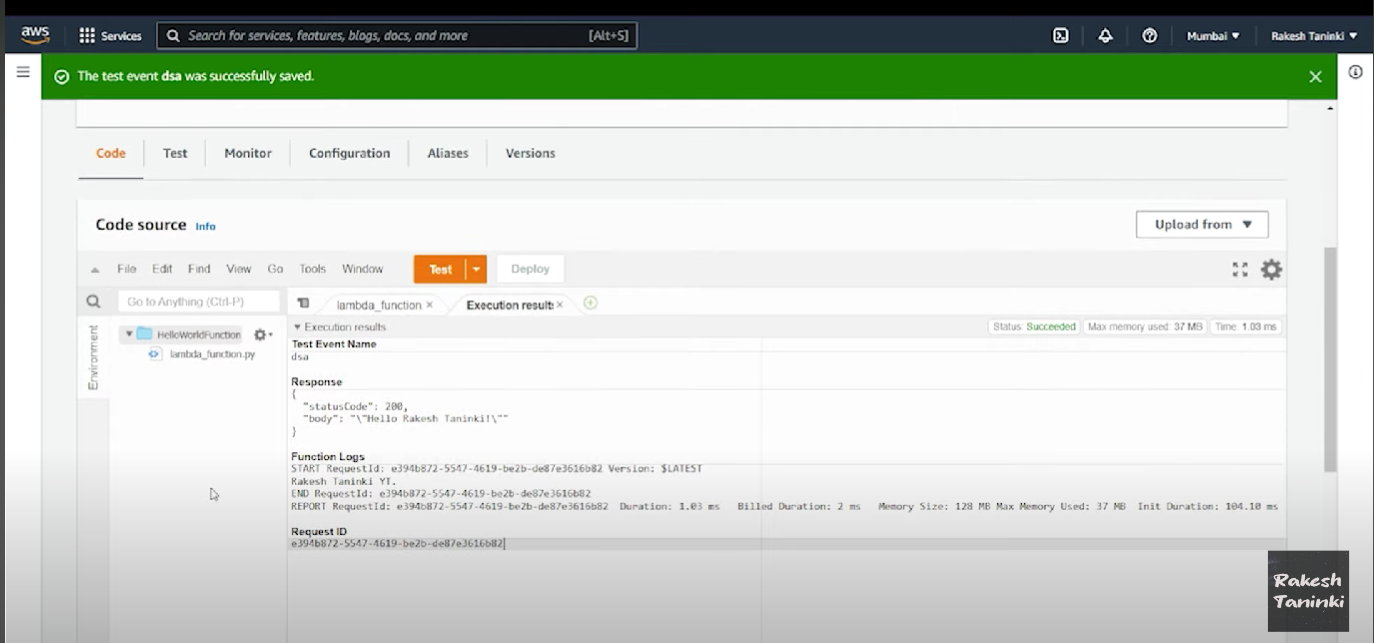




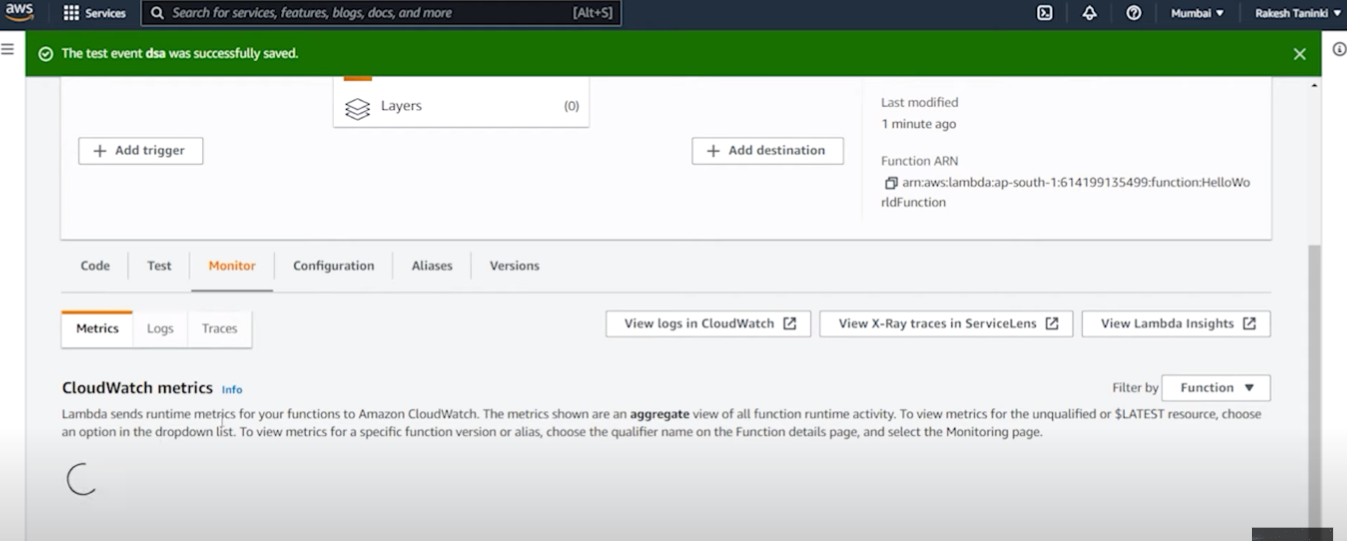
If you change/ modify code in lambda function, then you need to click on deploy. Then only our code gets modified.

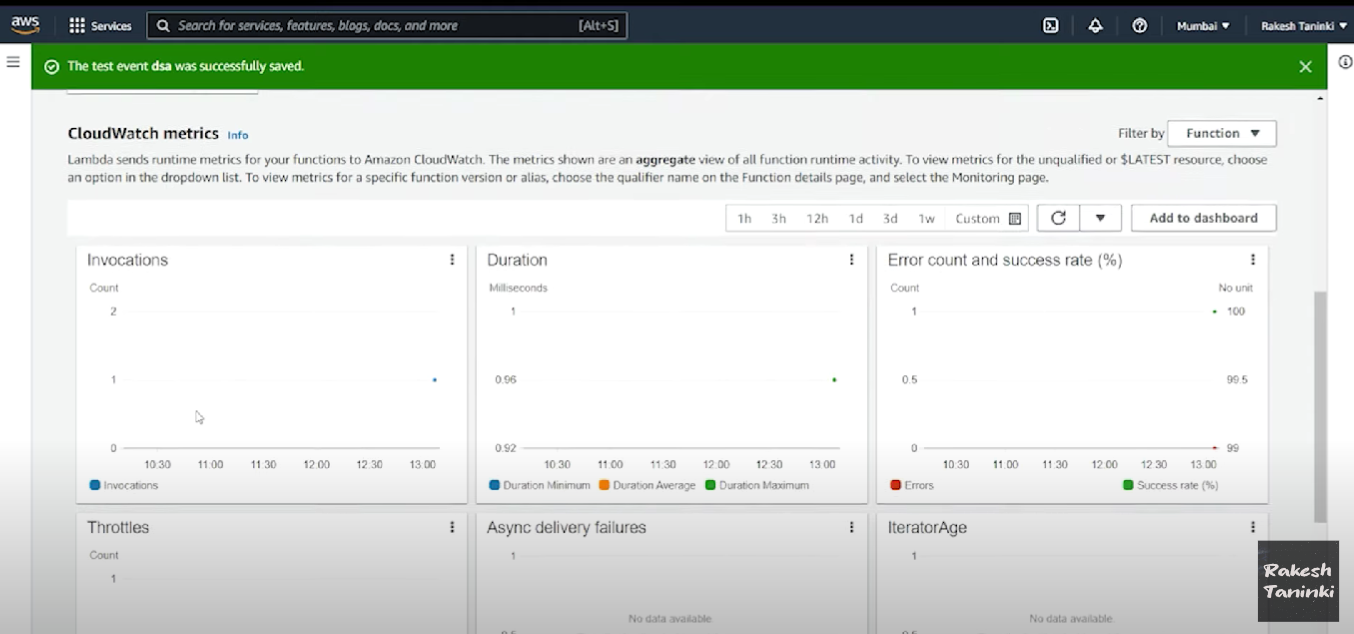
To execute this lambda function, we can click on **Test button**.

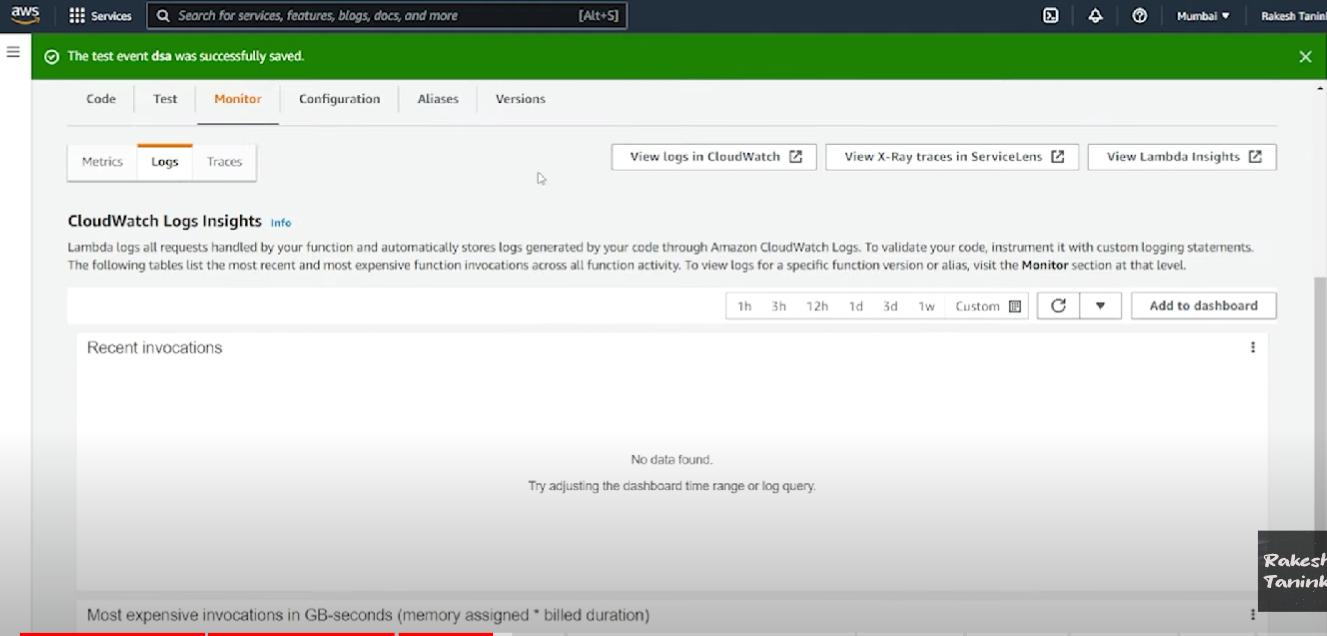
If you click on Test button, then it will asks for event name and event and click on Test.



If you navigate to Monitor tab, then you can able to watch the logs

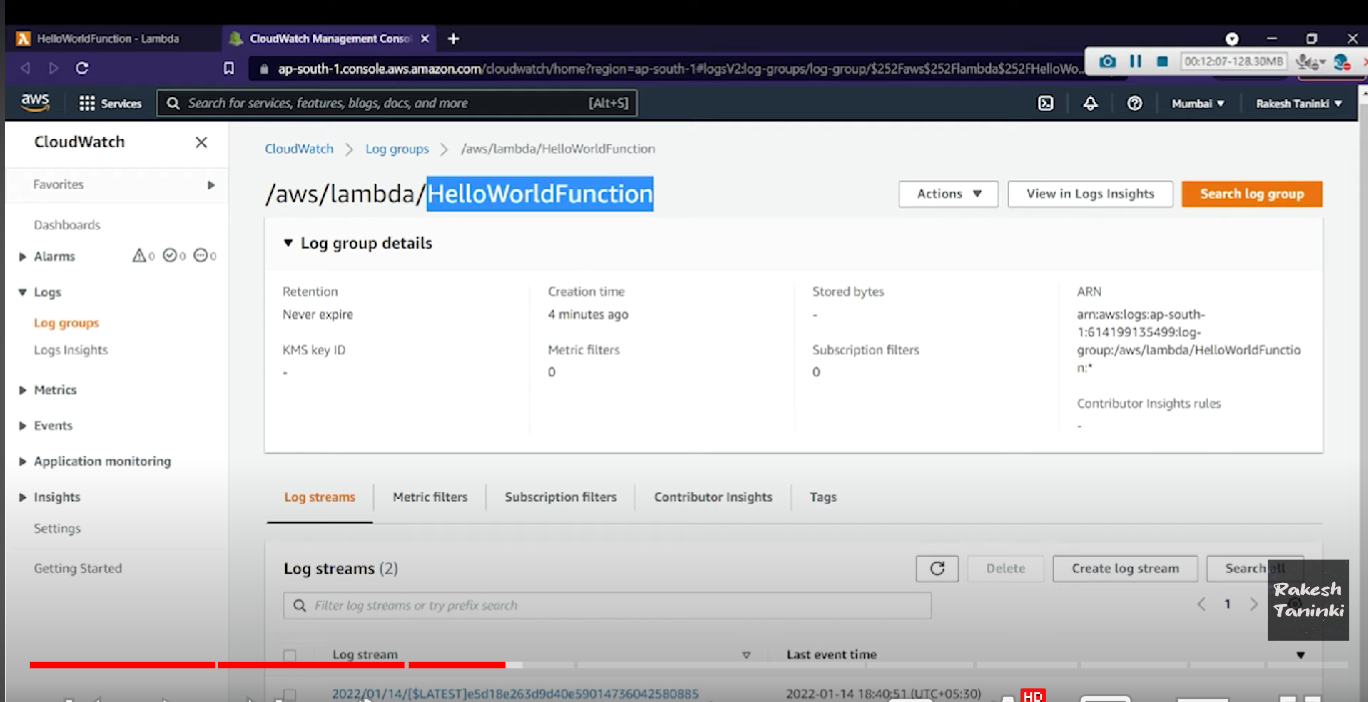




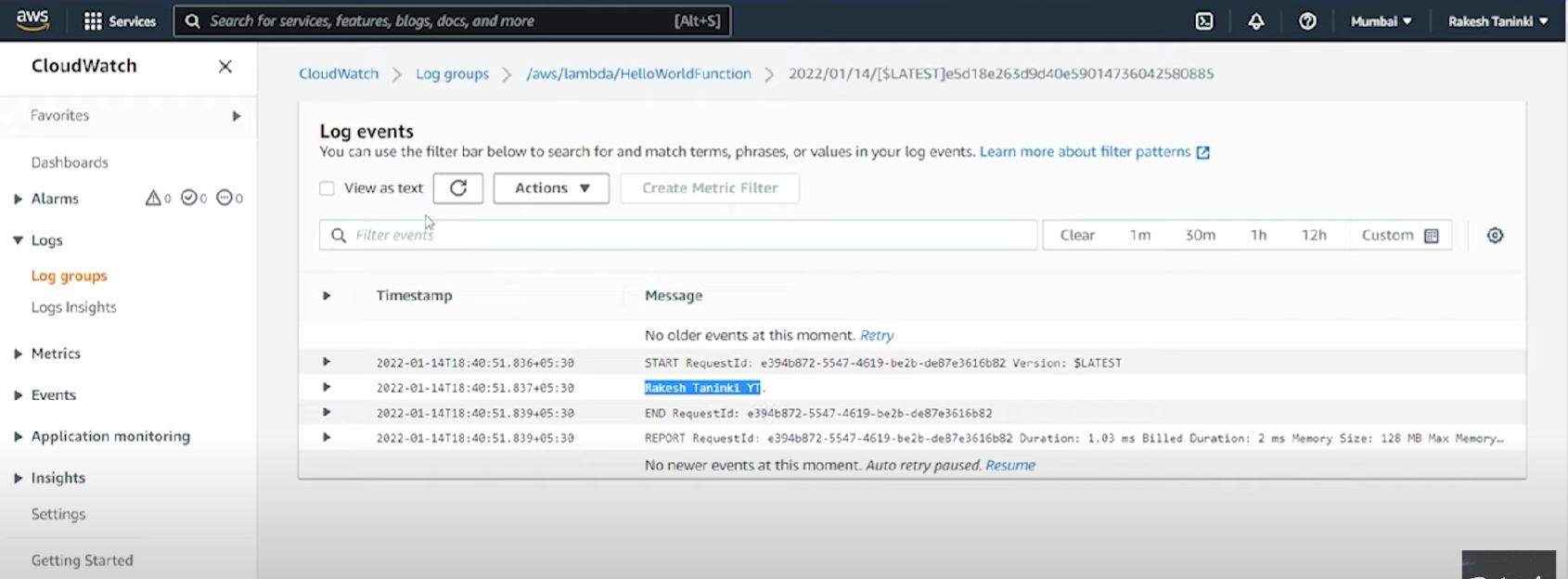


To See the logs, you can click on view logs in cloudWatch.

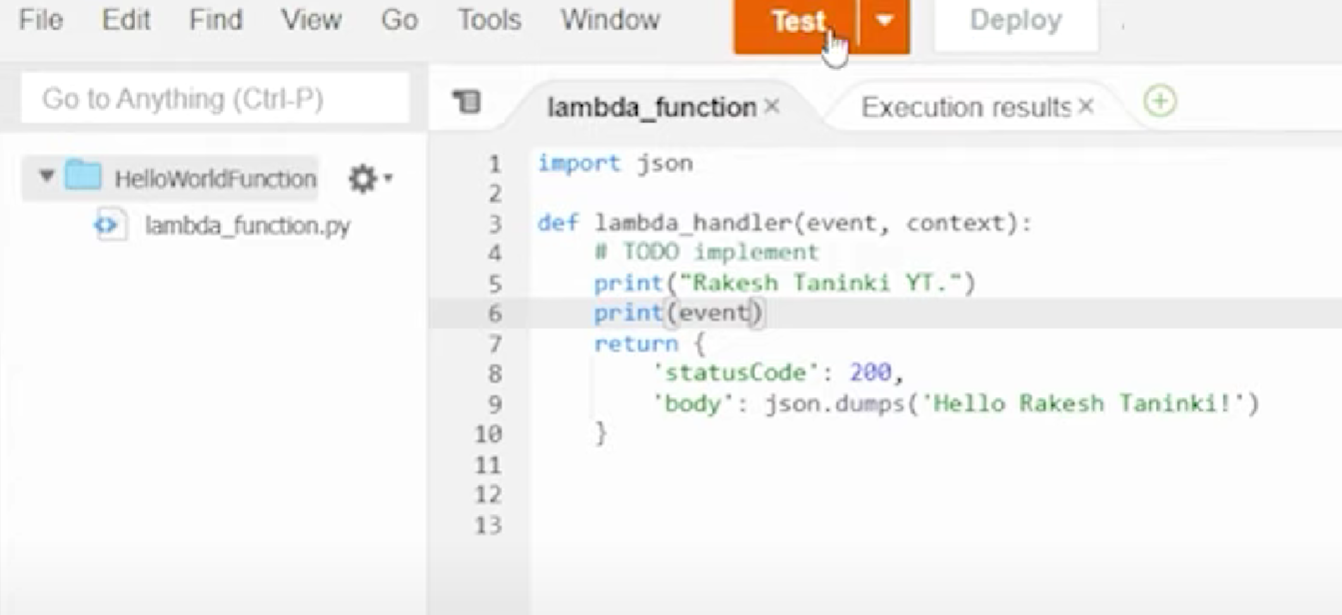
Now, you can navigate to cloud watch console.



Now, you can see the particular lambda function, and respective logs.



Here you can see the logs.



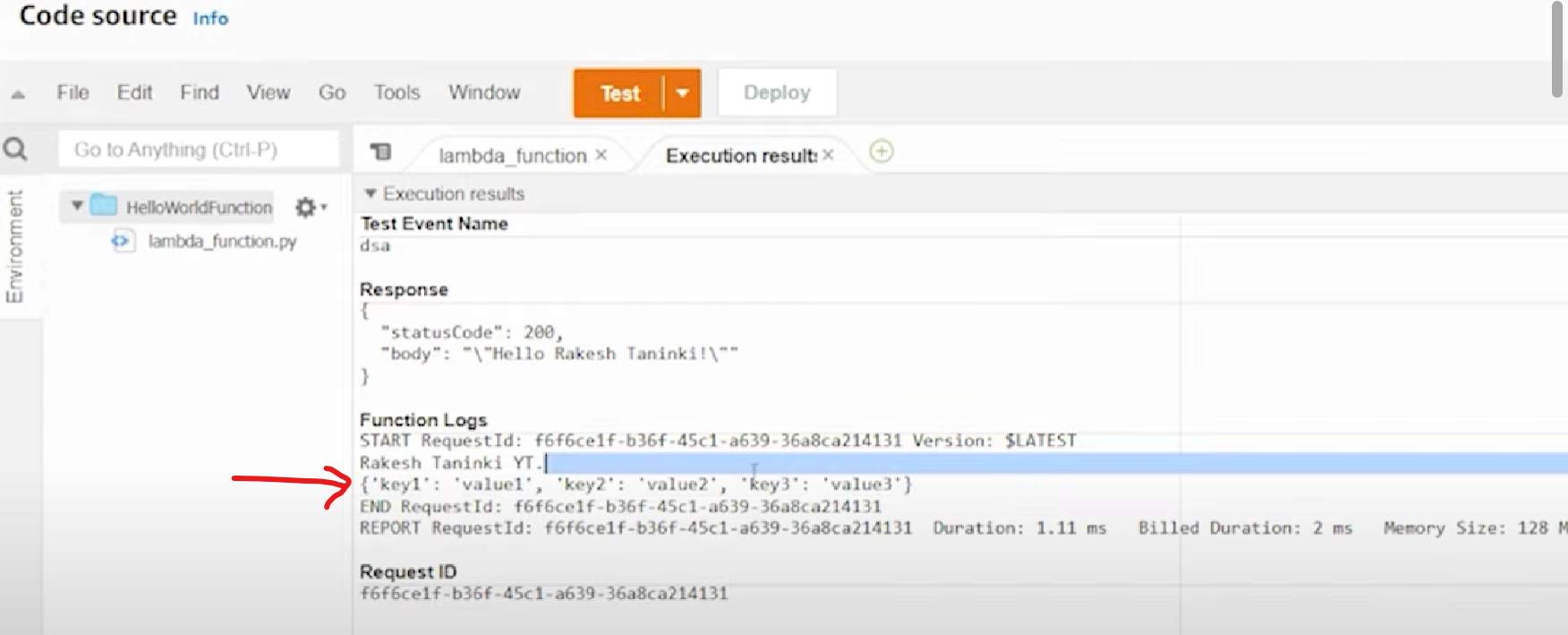
So, for every lambda function, we have two parameters -> **event and context**

Here the event which we are passing is stored in **event** parameter

In the above lambda function, we wrote ,

print(event) which prints the event which we passed.

Now see the execution result,



Here , we have passed request payload as {‘key1’ : ‘value1’, ‘key2’ : ‘value2’} as an event.

And we are printing this event in the lambda function.

So, we got event in the response.