



Andhra Pradesh State Skill Development Corporation



AWS CLOUD COMPUTING

LAMBDA



Configuration of AWS Lambda



AWS Lambda

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume.

With Lambda, you can run code for virtually any type of application or backend service - all with zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability.

You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

Use Cases

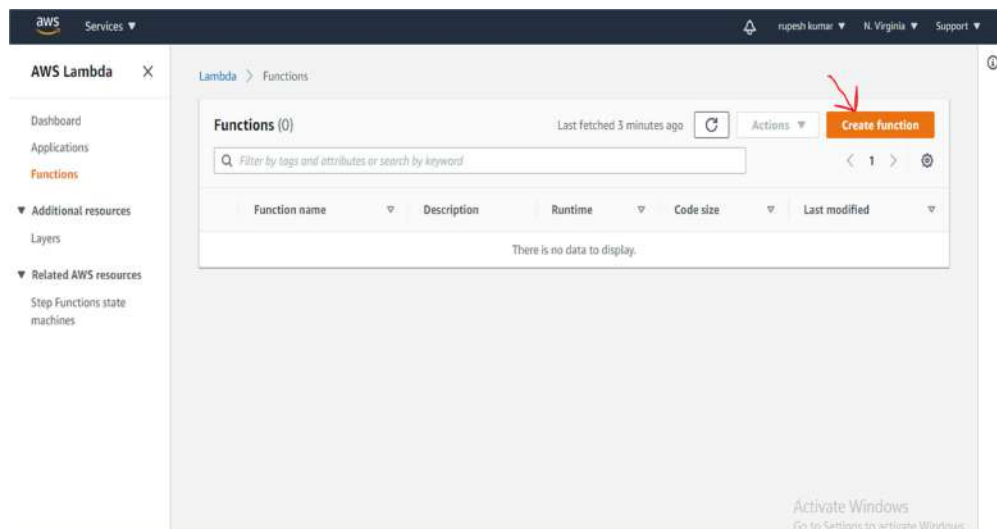
1. Data processing
2. Real-time file processing
3. Real-time stream processing
4. Machine learning
5. Web applications
6. IoT backends
7. Mobile backends

Benefits

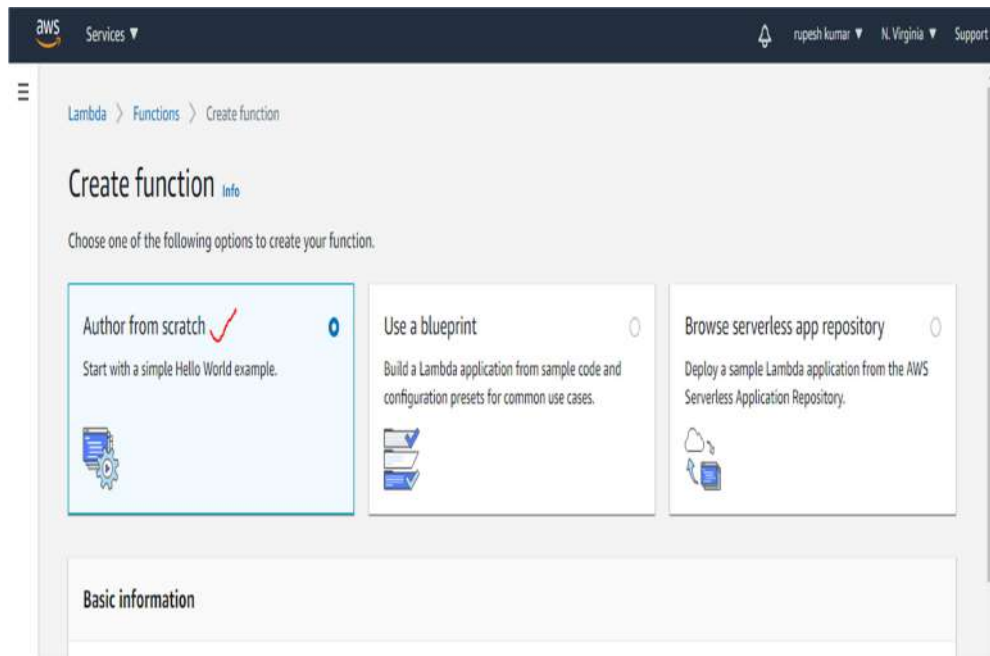
1. No servers to manage
2. Continuous scaling
3. Subsecond metering
4. Consistent performance

From the AWS management console, click on services and search for AWS lambda in the search box and click on it.

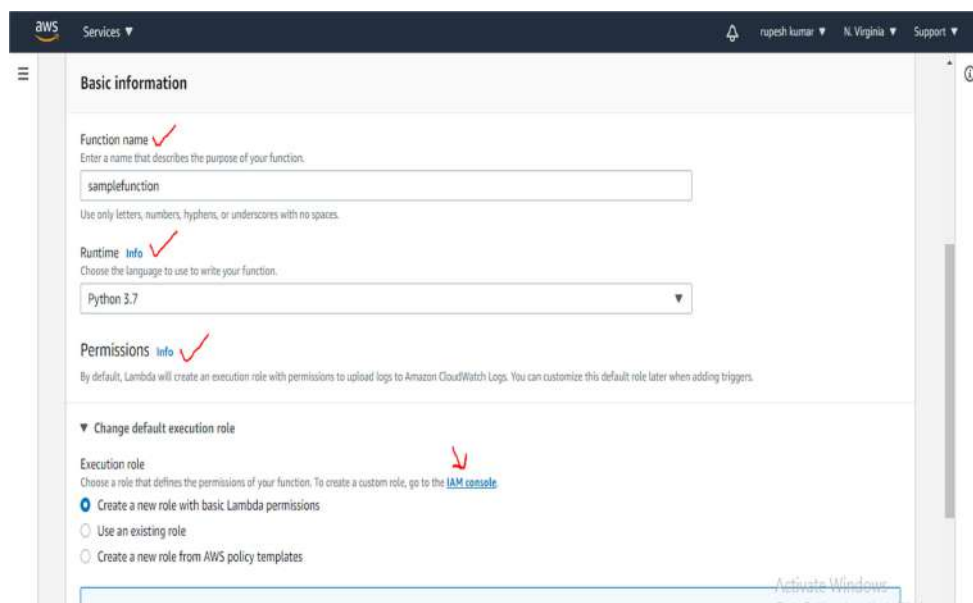
- Click on **create function**.



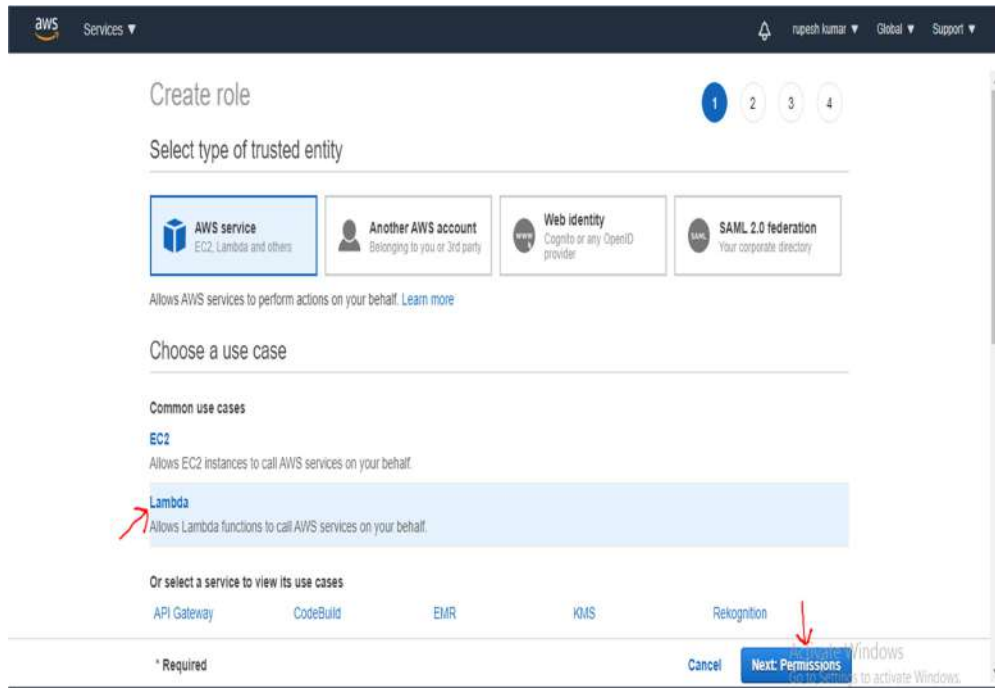
- Select **Author from scratch**



- Give a **function name** and select **runtime**.
- Under **permissions**, click on **change default execution role**.
- Select Create a new role with basic Lambda permissions.
- Then click on the **IAM Console**.



- Select **lambda** and click on **permissions**.



aws Services

Create role

Select type of trusted entity

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

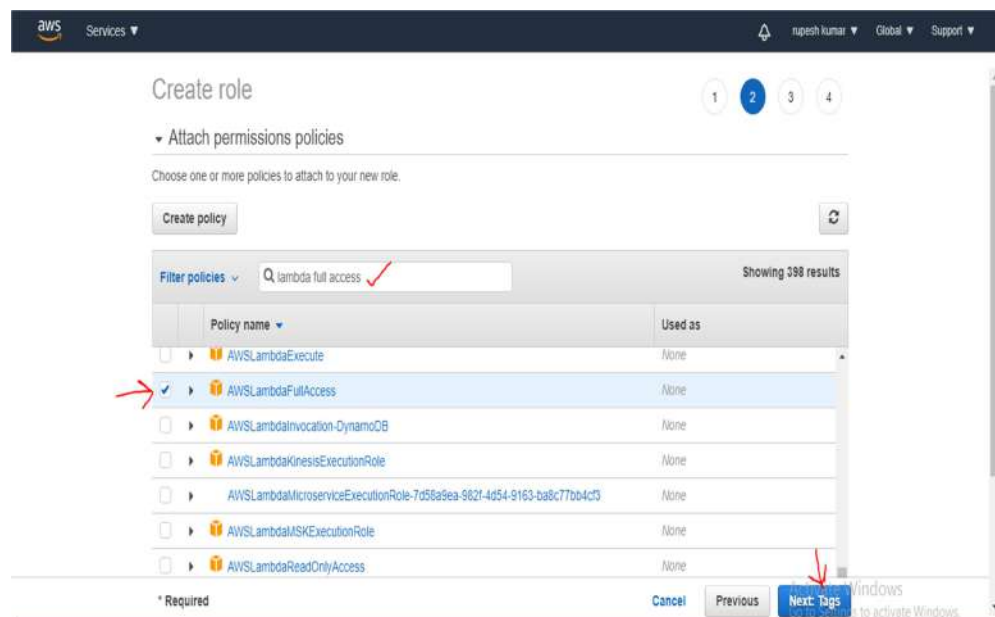
Or select a service to view its use cases

API Gateway CodeBuild EMR KMS Rekognition

* Required

Cancel Next: Permissions

- Search for **AWS Lambda Full Access**, then select AWS Lambda full access.
- Click on **tags**.



aws Services

Create role

Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

Filter policies

Showing 398 results

Policy name	Used as
<input type="checkbox"/> AWSLambdaExecute	None
<input checked="" type="checkbox"/> AWSLambdaFullAccess	None
<input type="checkbox"/> AWSLambdaInvocation-DynamoDB	None
<input type="checkbox"/> AWSLambdaKinesisExecutionRole	None
<input type="checkbox"/> AWSLambdaMicroserviceExecutionRole-7d58a9ea-982f-4d54-9163-ba8c77b04cf3	None
<input type="checkbox"/> AWSLambdaMSKExecutionRole	None
<input type="checkbox"/> AWSLambdaReadOnlyAccess	None

* Required

Cancel Previous Next: Tags

- If you want to add tags then you can add here, then click on review.



aws Services

rupesh kumar Global Support

Create role

1 2 3 4

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Add new key"/>	<input type="text"/>	

You can add 50 more tags.

Cancel Previous **Next: Review**

- Give a **role name** and click on **create role**.

aws Services

rupesh kumar Global Support

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name*
Use alphanumeric and "+, @, _" characters. Maximum 64 characters.

Role description
Maximum 1000 characters. Use alphanumeric and "+, @, _" characters.

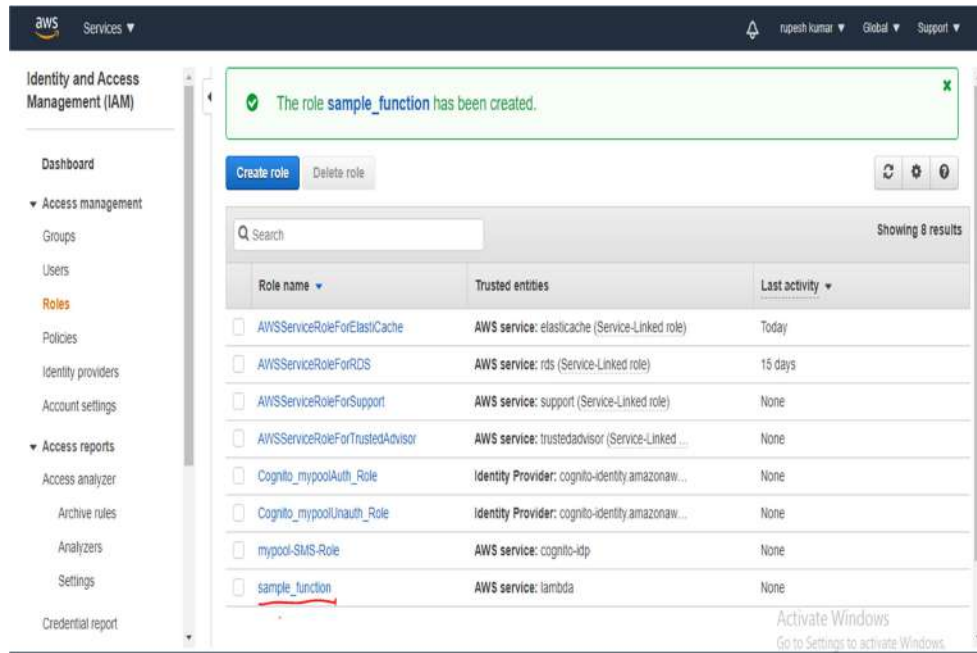
Trusted entities AWS service: lambda.amazonaws.com

Policies AWSLambdaFullAccess

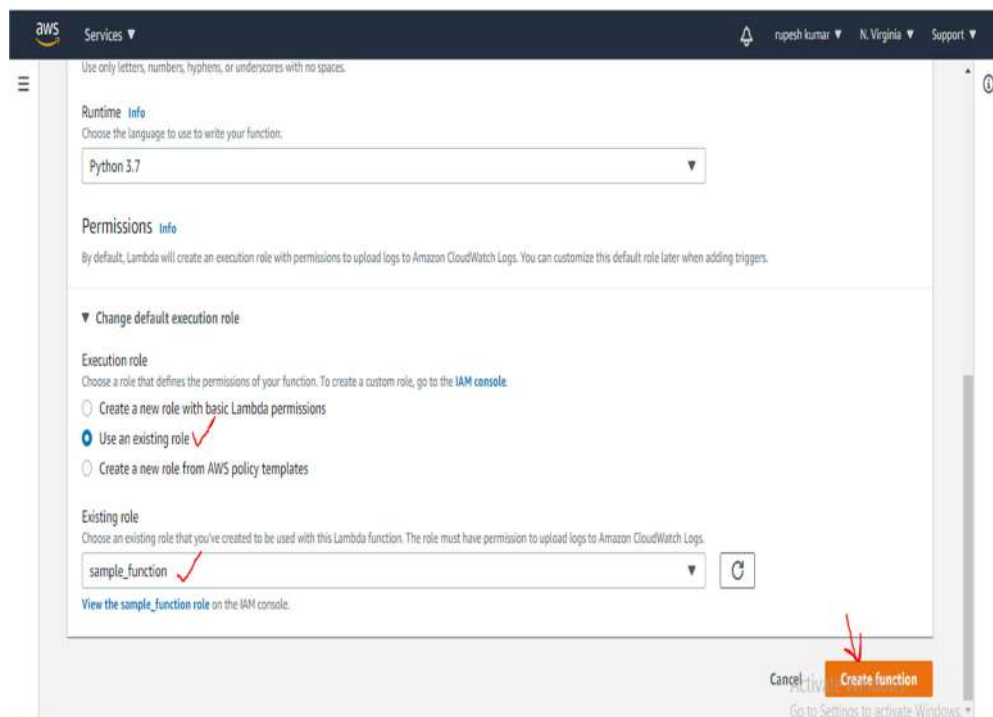
Permissions boundary Permissions boundary is not set

* Required

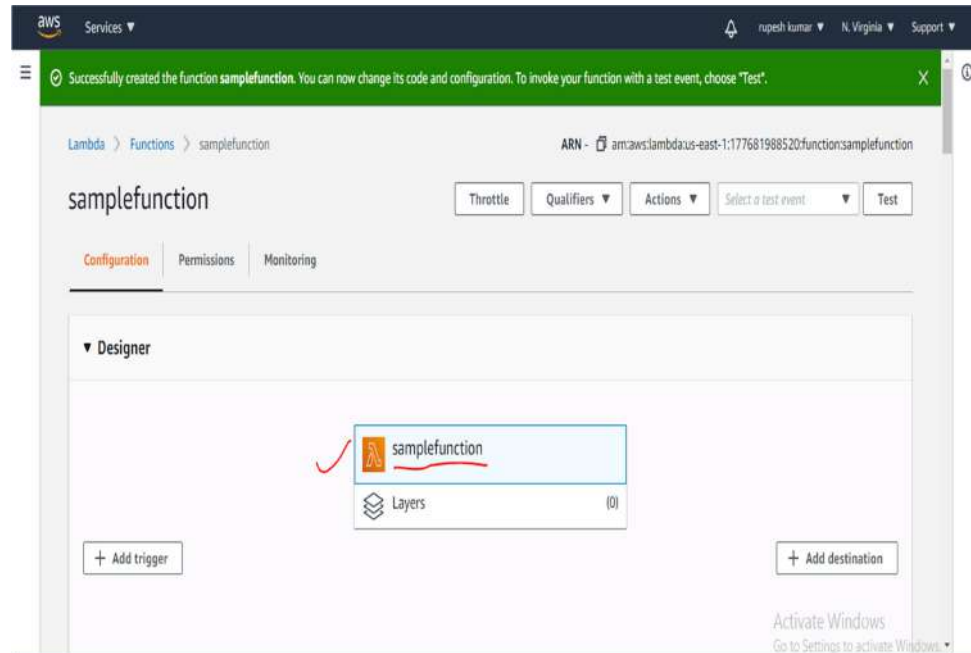
Cancel Previous **Create role**



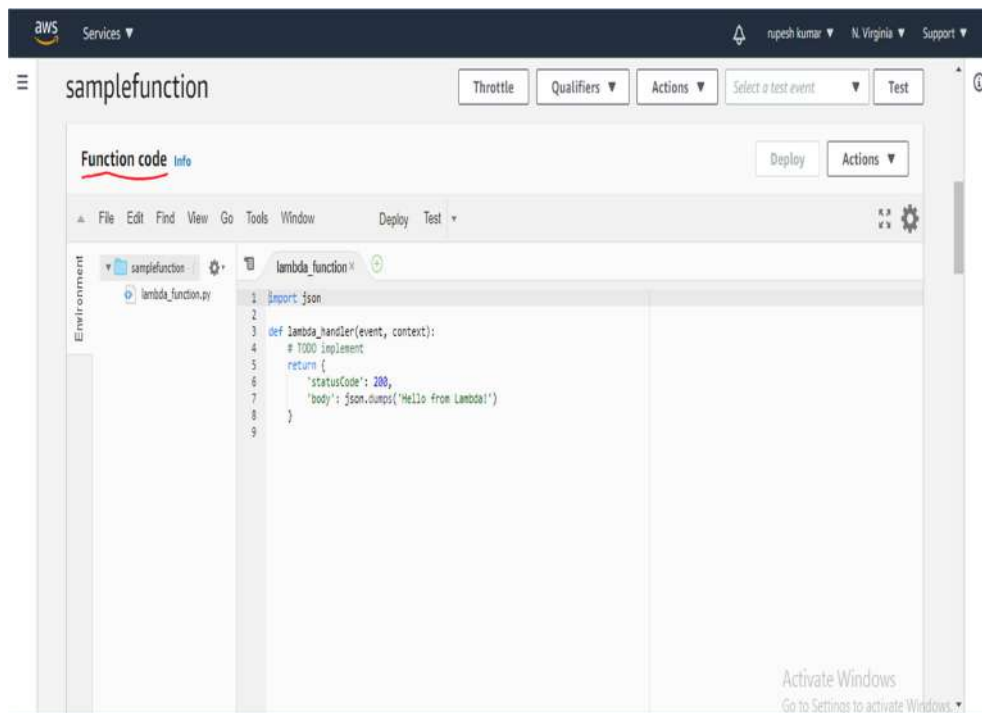
- Come back to the lambda **console** and select **Use an existing role** then select your **created role**.
- Click on **create function**.



- Click on **function name**.



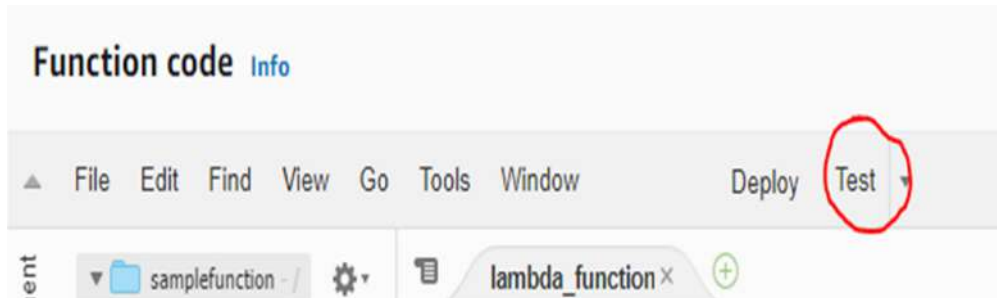
- Here you can see the **function code**, if you want to change the code as per your requirement then you can change.
- The code format is **json**.



Simple arithmetic operation:

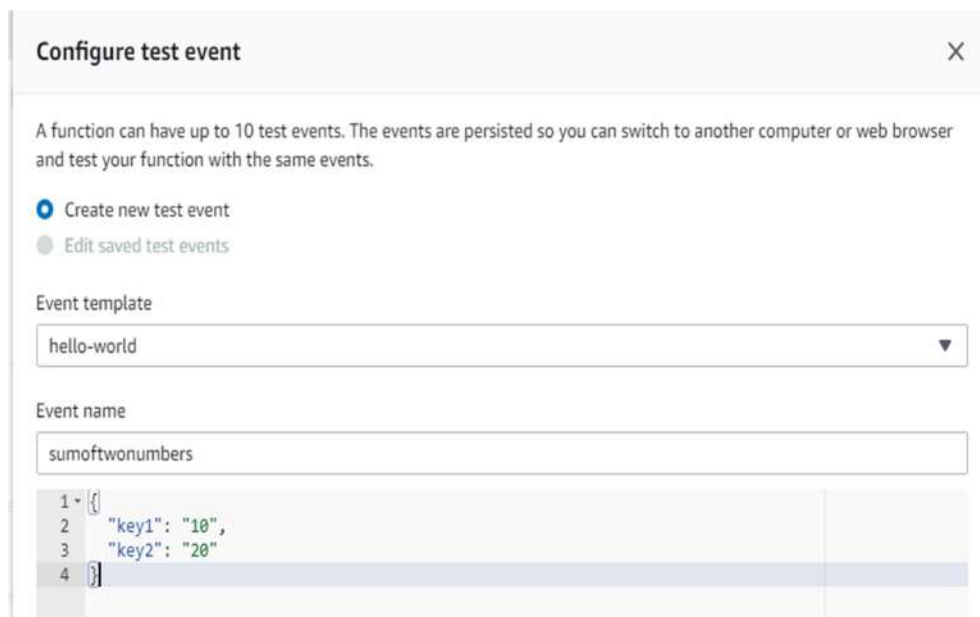
Sum of two numbers

- Assign events, for that click on **test**.



- Give an **Event template** as per your requirement and give the **event name**.
- Assign some values to keys.

```
{  
  "key1": "10",  
  "key2": "20"  
}
```

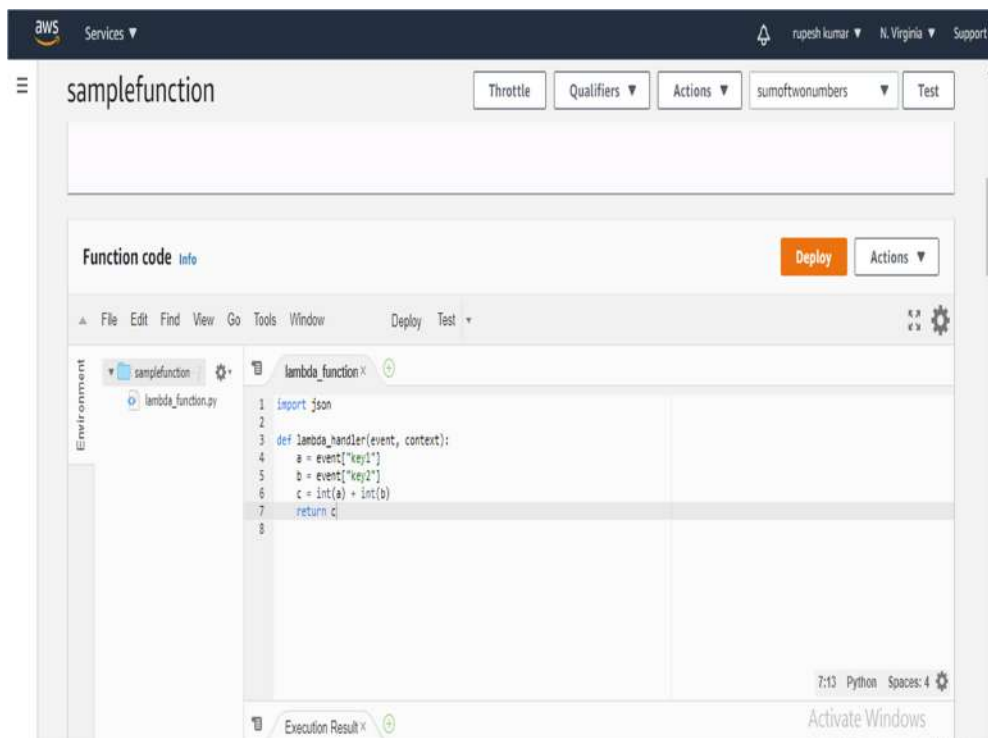


- Scroll down and click on **create**.





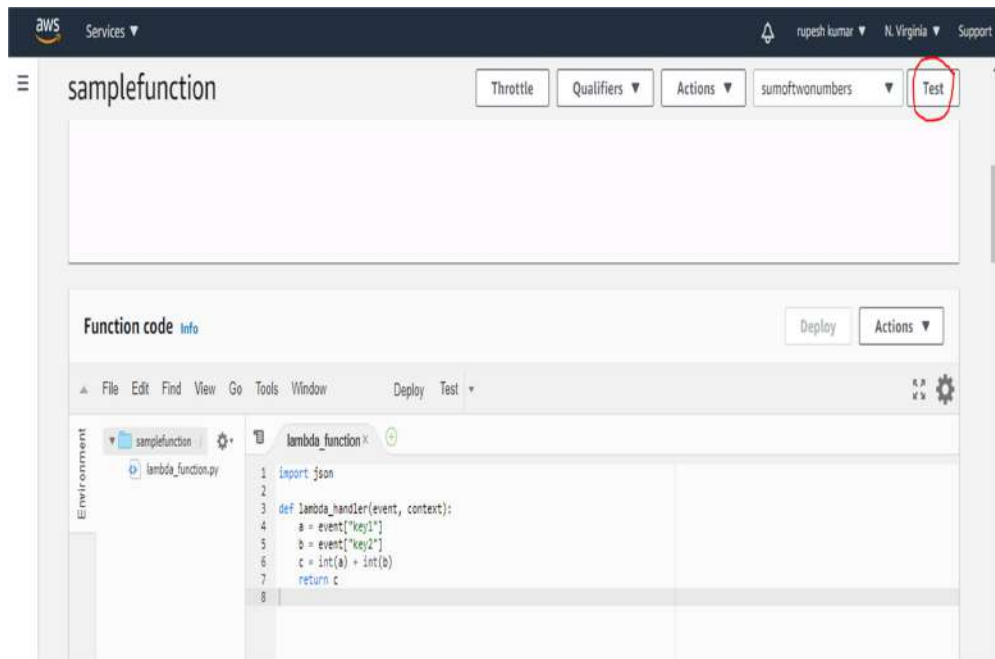
- Now you can write your code in lambda function, as shown below.
- Click on **deploy**.



```
import json
```

```
def lambda_handler(event, context):
    a = event["key1"]
    b = event["key2"]
    c = int(a) + int(b)
    return c
```

- Click on **test**.



- **Response** is shown in the below picture.
- **Response:30**

