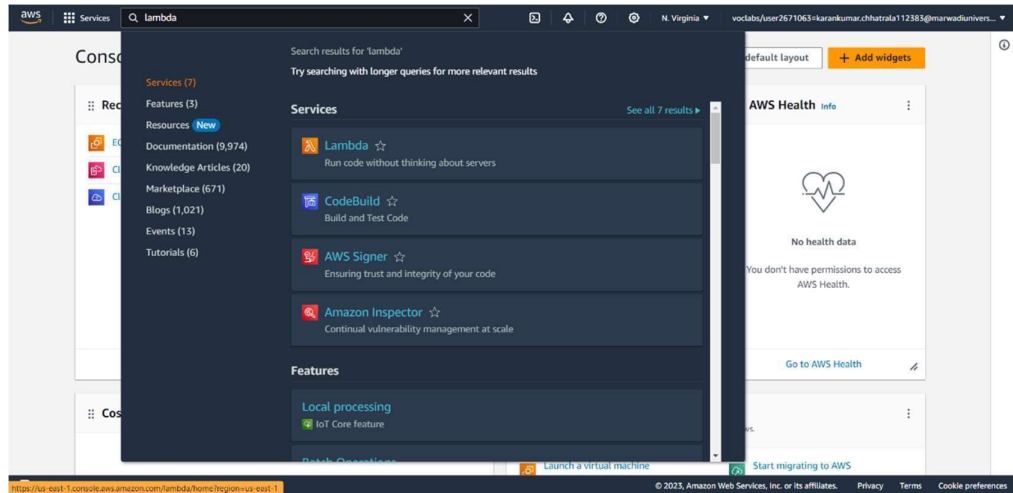


Practical 12 : Using Auto Scaling with AWS Lambda and Lifecycle Hooks

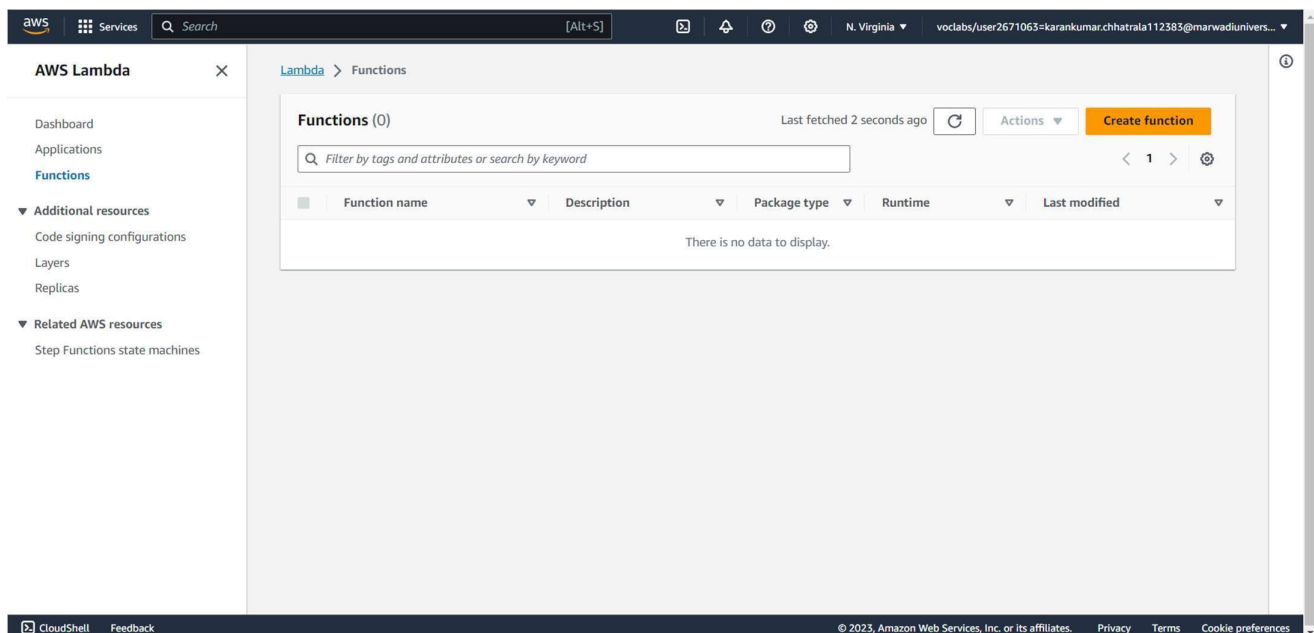
Step 01 : Search Lambda in service section

Snapshot :



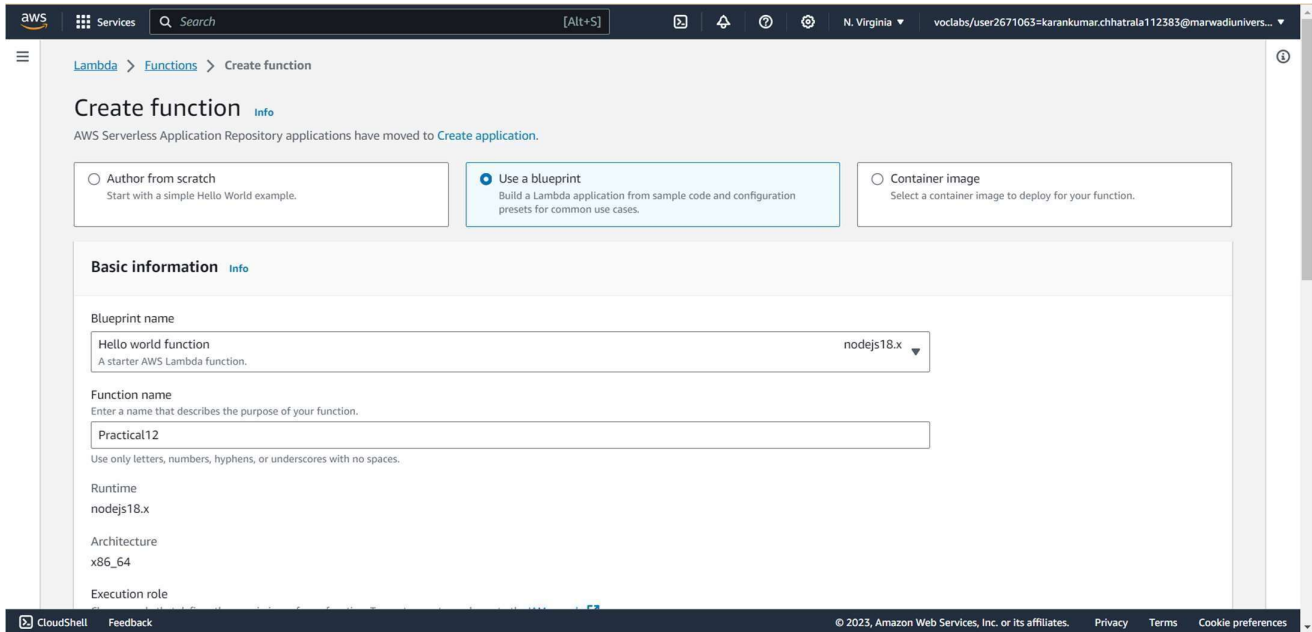
Step 02 : Click on create function

Snapshot :



Step 03 : Create a function using Lambda service

Snapshot :



The screenshot shows the AWS Lambda 'Create function' page. The 'Use a blueprint' option is selected. The 'Blueprint name' is 'Hello world function' and the 'Runtime' is 'nodejs18.x'. The 'Function name' is 'Practical12'. The 'Architecture' is 'x86_64' and the 'Execution role' is 'AWSLambdaBasicExecutionRole'.

Create function [Info](#)

AWS Serverless Application Repository applications have moved to [Create application](#).

☐ Author from scratch
Start with a simple Hello World example.

☒ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

☐ Container image
Select a container image to deploy for your function.

Basic information [Info](#)

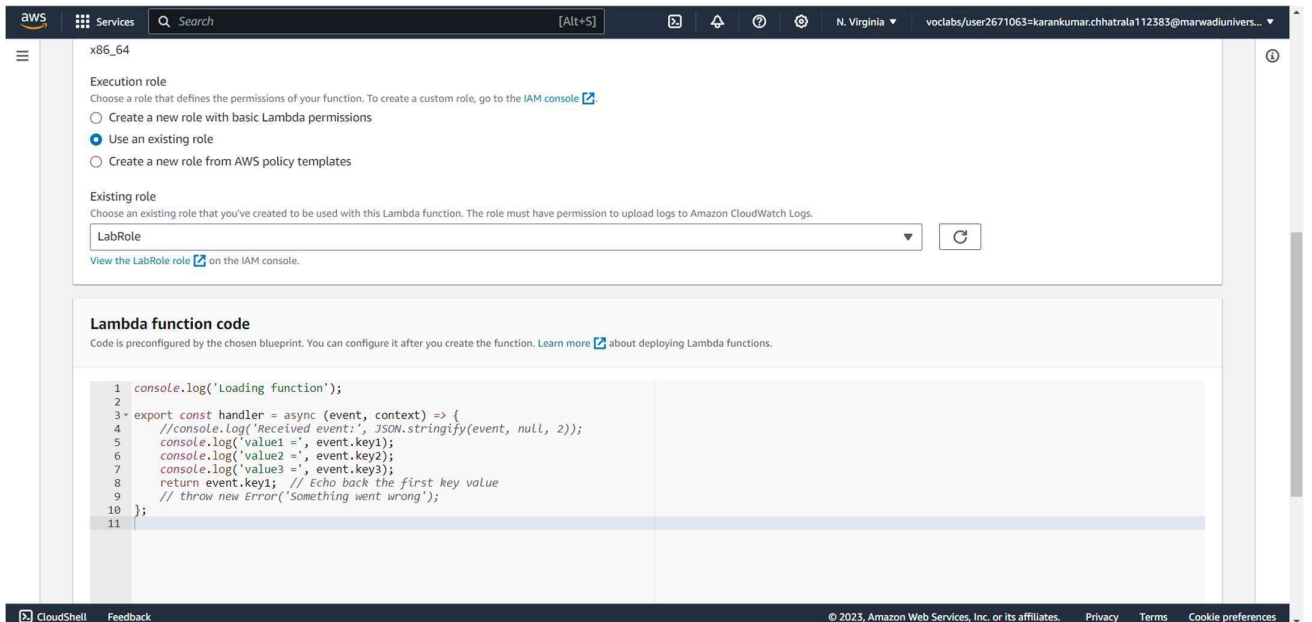
Blueprint name
Hello world function
A starter AWS Lambda function.

Function name
Enter a name that describes the purpose of your function.
Practical12
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime
nodejs18.x

Architecture
x86_64

Execution role
AWSLambdaBasicExecutionRole



The screenshot shows the AWS Lambda 'Execution role' page. The 'Use an existing role' option is selected. The 'Existing role' is 'LabRole'.

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
LabRole
[View the LabRole role](#) on the IAM console.

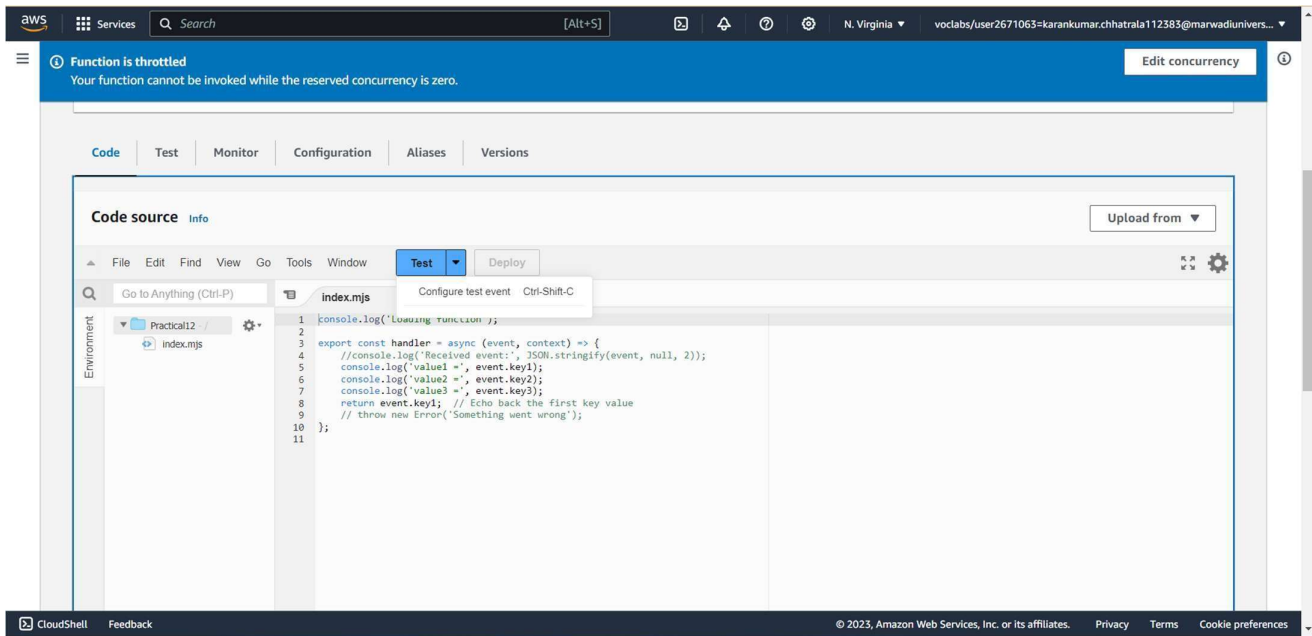
Lambda function code

Code is preconfigured by the chosen blueprint. You can configure it after you create the function. [Learn more](#) about deploying Lambda functions.

```
1 console.log('Loading function');
2
3 export const handler = async (event, context) => {
4   //console.log('Received event:', JSON.stringify(event, null, 2));
5   console.log('value1 =', event.key1);
6   console.log('value2 =', event.key2);
7   console.log('value3 =', event.key3);
8   return event.key1; // Echo back the first key value
9   // throw new Error('Something went wrong');
10 };
11
```

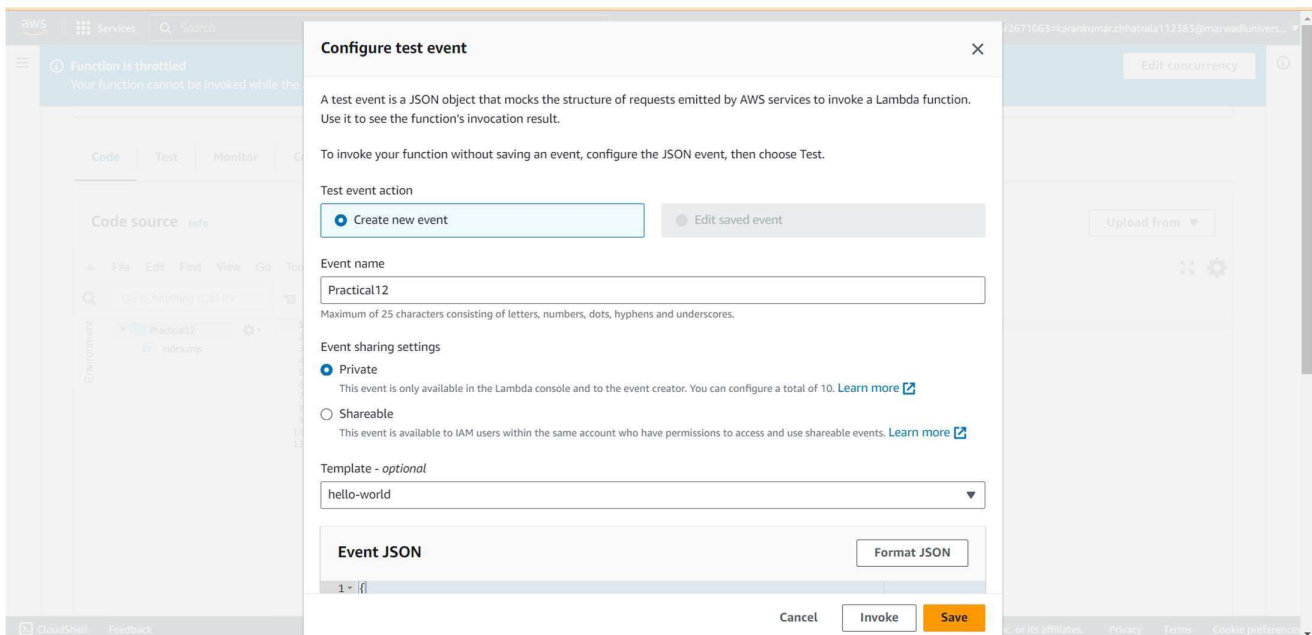
Step 04 : Test >> Configure Test Event

Snapshot :



Step 05 : Create a Configure Test Event

Snapshot :



Step 06 : Click on Test to check config.

Snapshot :

