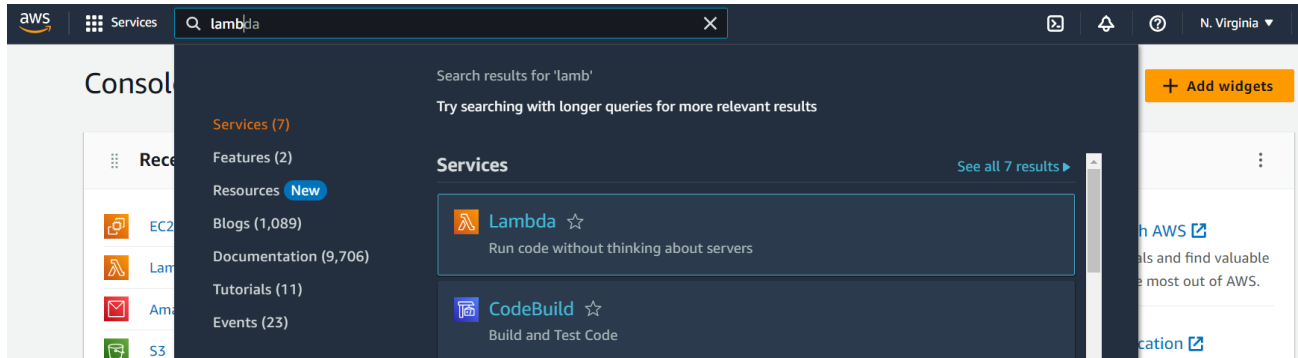


Assignment 15

Problem Statement: Create a serverless computing service.

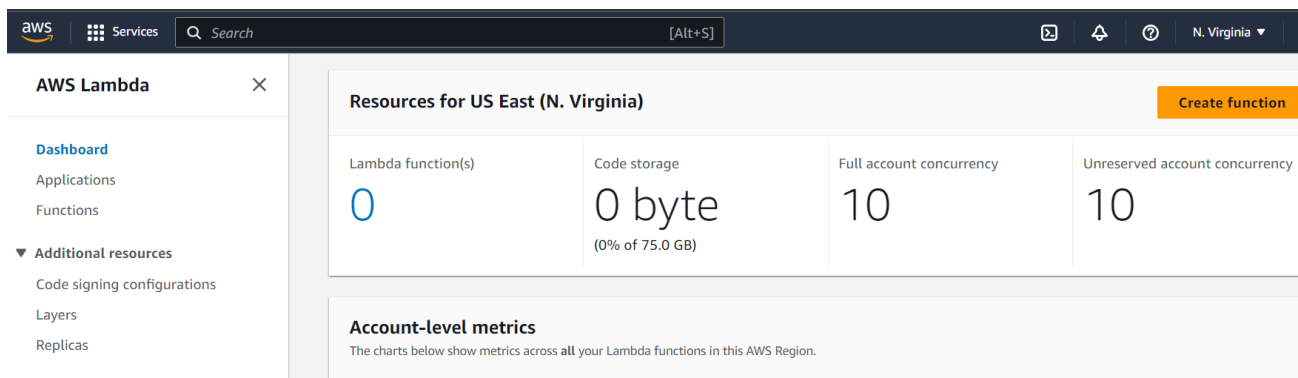
Procedure:

1. Sign-in to your AWS console.
2. Search for Lambda.

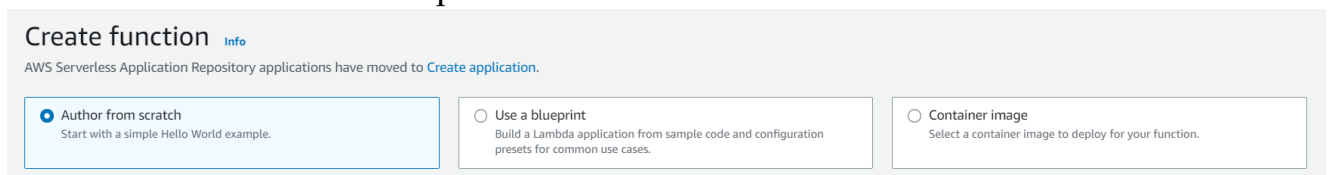


Click on the first result named Lambda.

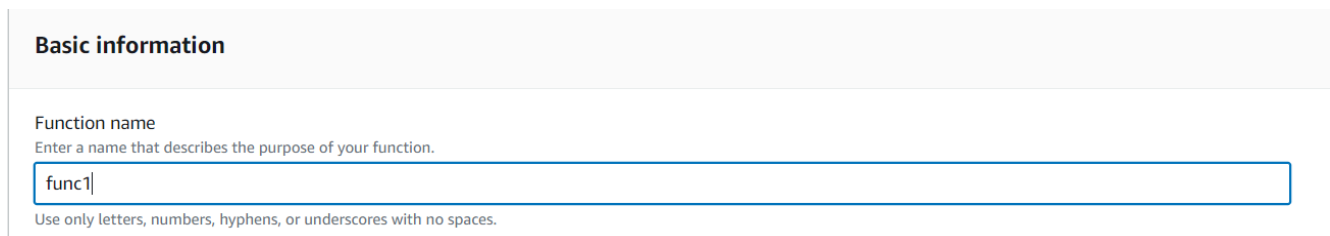
3. Now click on the Create Function button on the top right corner.



4. Select Author from scratch option.



5. Give the name of the function.



6. Choose Node.js as the Runtime. (No need to change architecture.)

Runtime [Info](#)

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Node.js 18.x

Architecture [Info](#)

Choose the instruction set architecture you want for your function code.

☒ x86_64

☐ arm64

Permissions [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

► Change default execution role

7. Now click on the create function button.

Create function

8. Now scroll-down to the code section of your newly created function.


Successfully created the function **func1**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".


Lambda > Functions > func1

func1

Throttle Copy ARN Actions

▼ Function overview [Info](#)


 func1

 Layers (0)

+ Add trigger + Add destination

Description
-

Last modified
8 seconds ago

Function ARN
 arn:aws:lambda:ap-south-1:728364961341:function:func1

Function URL [Info](#)
-

Code Test Monitor Configuration Aliases Versions

9. Change the string in the code to be displayed.

Code source [Info](#) Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

func1 / index.mjs

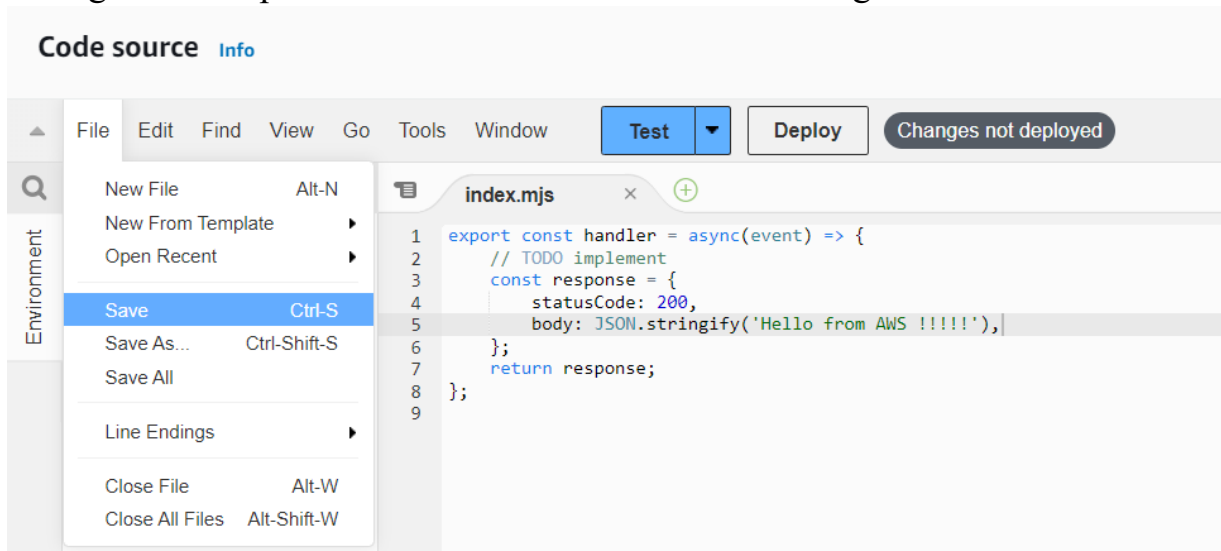
```
1 export const handler = async(event) => {
2   // TODO implement
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify('Hello from Lambda!'),
6   };
7   return response;
8 };
9
```

We changed it to this...

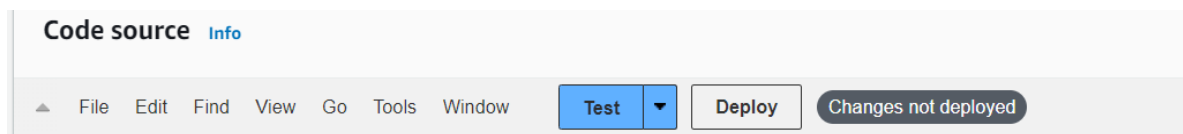
index.mjs

```
1 export const handler = async(event) => {
2   // TODO implement
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify('Hello from AWS !!!!!'),
6   };
7   return response;
8 };
9
```

10. Now go to File option and click on save to save the changes.



11. Now click on the Test button.



12. Select Create New Event. Then give a name. Then click on save.

Configure test event ✕

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event

☐ Edit saved event

Event name

e1

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

☒ Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

☐ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

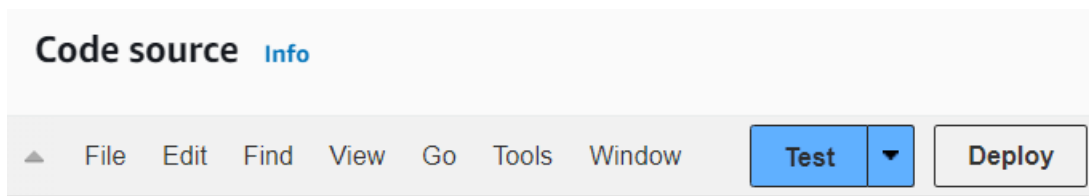
Template - optional

hello-world

Event JSON Format JSON

1 2 3

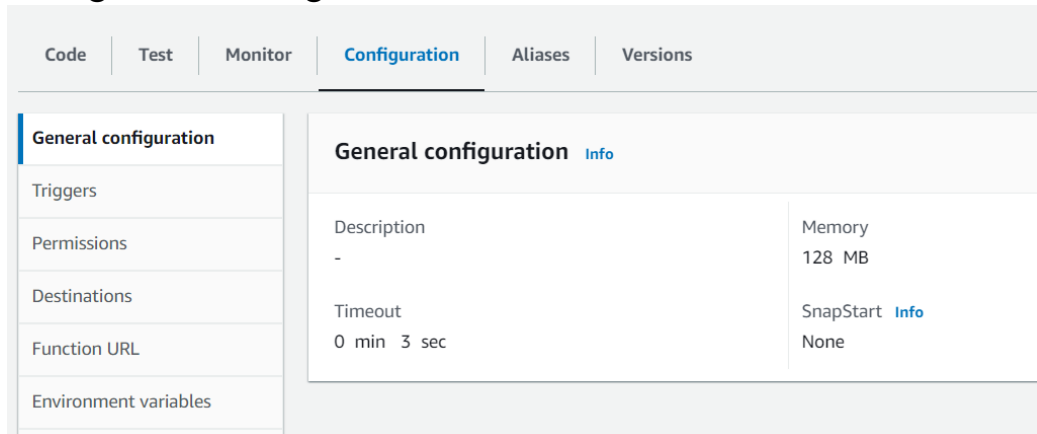
Cancel Save



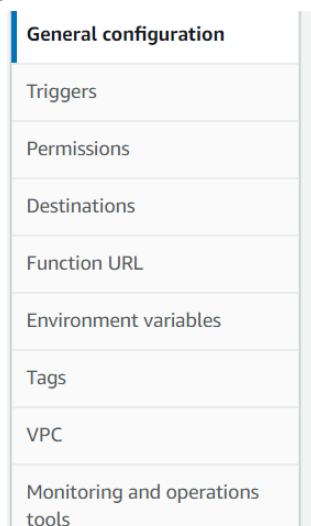
14. After successful deployment a message will pop as mentioned below and the deploy button will be locked out, indicating that our Function has been successfully deployed.



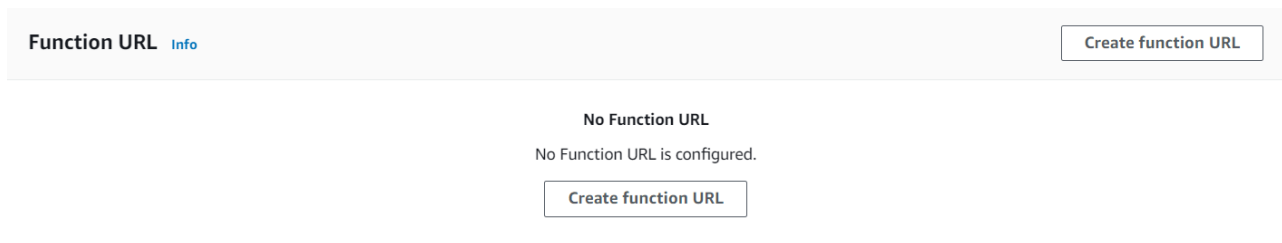
15. Now go to the configuration tab.



16. Click on the Function URL option in the left side Bar.



17. Click on Create function URL.



18. Choose NONE and click on the save button.

Configure Function URL

Function URL [Info](#)
Use function URLs to assign HTTP(S) endpoints to your Lambda function.

Auth type
Choose the auth type for your function URL. [Learn more](#)

☐ AWS_IAM
Only authenticated IAM users and roles can make requests to your function URL.

☒ NONE
Lambda won't perform IAM authentication on requests to your function URL. The URL endpoint will be public unless you implement your own authorization logic in your function.

ⓘ A policy exists that grants public access to your function URL. If you choose auth type NONE, anyone with the URL can access your function.


► Additional settings


Cancel Save

19. Now copy the newly created Function URL and paste it in a different browser.

ⓘ Your changes have been saved.

▼ Function overview [Info](#)

 func1


 Layers (0)



[+ Add trigger](#)

[+ Add destination](#)

Description
-

Last modified
6 minutes ago

Function ARN
 `arn:aws:lambda:ap-south-1:728364961341:function:func1`

Function URL [Info](#)
 <https://ojtk6scr7wkr5hqrh2iavadyhe0thqki.lambda-url.ap-south-1.on.aws/> 

https://e2plgkbsn7pinv2obpnlm... x +

<https://e2plgkbsn7pinv2obpnlmpwvcn0nhke.lambda-url.ap-south-1.on.aws>

"Hello from AWS !!!!"

We have successfully Created a Serverless Computing service.


To delete the Lambda Function, follow these steps:

1. Click on the Actions button on the top right side.


ⓘ Your changes have been saved.


Lambda > Functions > func1

func1

Throttle  Copy ARN Actions ▼

▼ Function overview [Info](#)

 func1


 Layers (0)



[+ Add trigger](#)

[+ Add destination](#)

Description
-

Last modified
13 minutes ago

Function ARN
 `arn:aws:lambda:ap-south-1:728364961341:function:func1`

Function URL [Info](#)
 <https://e2plgkbsn7pinv2obpnlmpwvcn0nhke.lambda-url.ap-south-1.on.aws/> 

2. Select the Delete function option and then click on delete button in the pop-up.