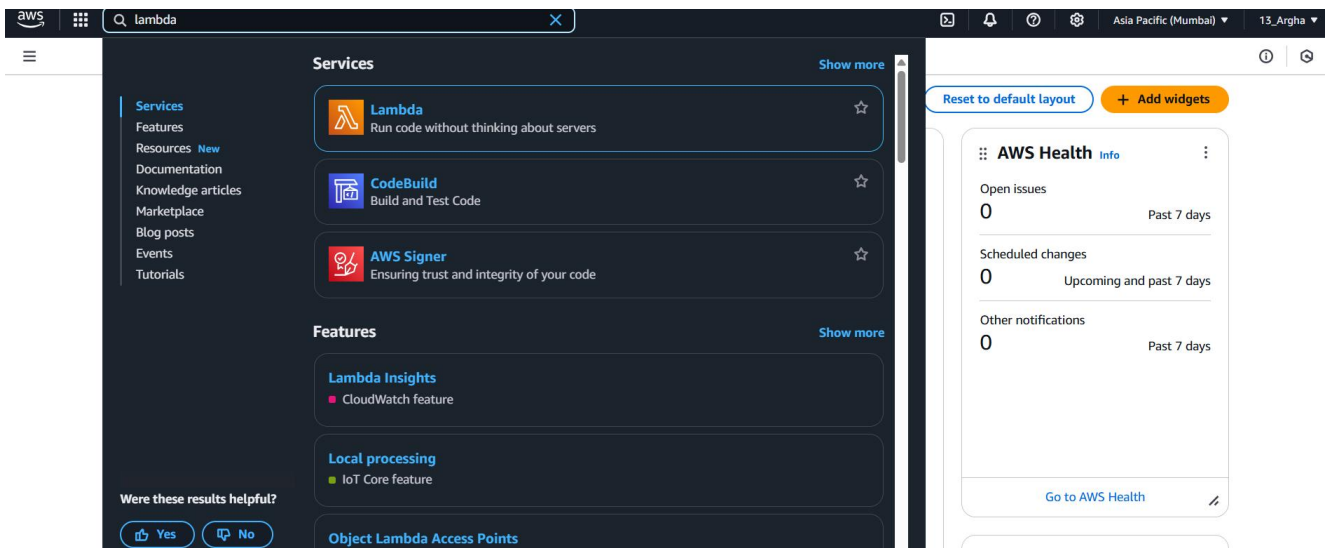


Assignment 14

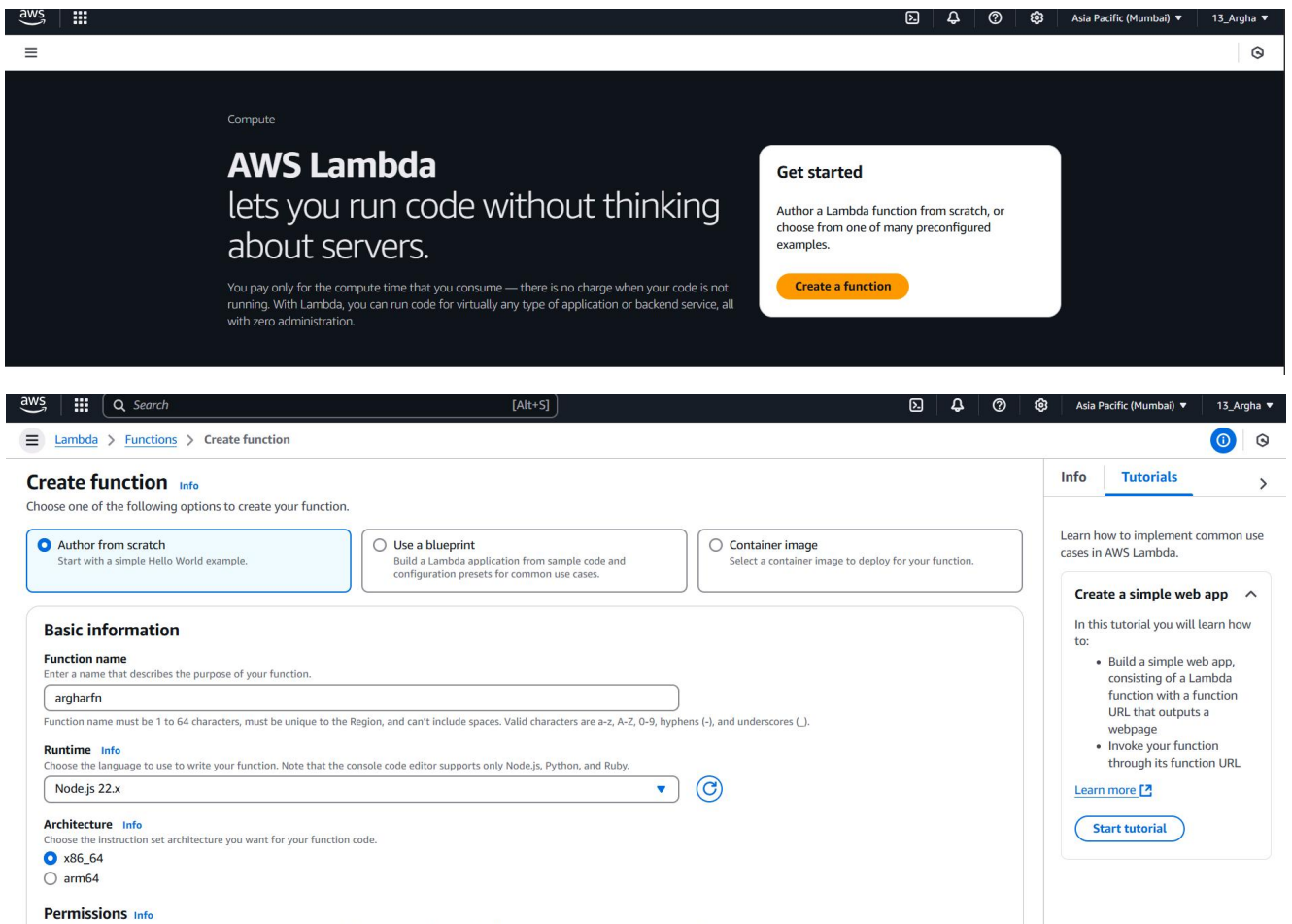
Title → Create an Elastic IP for an Instance.

TO create an Elastic IP for an Instance we have to follow some steps

Step 1 → Login to the account first. Then at the top left corner of AWS Management Console search **Lambda** and click on that .



Step 2 → Click on Create a Function . And give the Function name and Runtime as node.js 22.x and then Create Function.



Step 3 → After successfully created the function Click on code modify the code and deploy the code.

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

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

Upload from

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

General configuration Info

Memory: 128 MB

Ephemeral storage: 512 MB

Function URL Info

No Function URL

No Function URL is configured.

Create function URL

Info | **Tutorials**

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#)

[Start tutorial](#)

Step 4 → click on Configuration then click on function URL and Create a Function URL.

Configuration | Aliases | Versions

General configuration Info

Memory: 128 MB

Ephemeral storage: 512 MB

Function URL Info

No Function URL

No Function URL is configured.

Create function URL

Info | **Tutorials**

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#)

[Start tutorial](#)

Step 5 → click on Configure Function URL and select the Auth type None and save it.

The screenshot shows the 'Configure Function URL' page in the AWS Lambda console. The 'Auth type' is set to 'NONE'. A blue box contains a note: 'When you choose auth type NONE, Lambda automatically creates the following resource-based policy and attaches it to your function. This policy makes your function public to anyone with the function URL. You can edit the policy later. To limit access to authenticated IAM users and roles, choose auth type AWS_IAM.' Below this, the 'View policy statement' section shows a JSON policy. On the right, there is a 'Tutorials' sidebar with a 'Create a simple web app' tutorial.

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.

Function URL permissions

When you choose auth type **NONE**, Lambda automatically creates the following resource-based policy and attaches it to your function. This policy makes your function public to anyone with the function URL. You can edit the policy later. To limit access to authenticated IAM users and roles, choose auth type **AWS_IAM**.

View policy statement

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "StatementId": "FunctionURLAllowPublicAccess",  
6       "Effect": "Allow",  
7       "Principal": "*",  
8       "Action": "lambda:InvokeFunctionUrl",  
9       "Resource": "arn:aws:lambda:ap-south-1:194722440244:function:argharfn",  
10      "Condition": {  
11        "StringEquals": {  
12          "lambda:FunctionUrlAuthType": "NONE"  
13        }  
14      }  
15    ]  
16  }  
17 }
```

Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#)

[Start tutorial](#)

Step 6 → Copy the URL and open in a new browser/tab/incognito tab.

The top screenshot shows the 'Function overview' page for 'argharfn'. It includes buttons for 'Throttle', 'Copy ARN', and 'Actions'. The 'Function overview' section shows the function name, layers, and a '+ Add trigger' button. The 'Description' section shows the function ARN and a 'Copy' button. The 'Function URL' section shows the URL and a 'Copy' button. The bottom screenshot shows a browser window with the URL 'https://3b3d7emoz3zuhk2pbtiv6bnmd0kovpm.lambda-url.ap-south-1.on.aws'.

argharfn

[Throttle](#) [Copy ARN](#) [Actions](#)

Function overview [Info](#)

[Diagram](#) [Template](#)

[+ Add trigger](#)

Layers (0)

[+ Add destination](#)

Description

Last modified 3 minutes ago

Function ARN

[Copy](#) arn:aws:lambda:ap-south-1:194722440244:function:argharfn

[Copy](#) [View URL](#) [Info](#)

[https://3b3d7emoz3zuhk2pbtiv6bnmd0kovpm.lambda-url.ap-south-1.on.aws/](#)

Browser: 3b3d7emoz3zuhk2pbtiv6bnmd0kovpm.lambda-url.ap-south-1.on.aws

Pretty-print ☐

"Hello from Argha!"