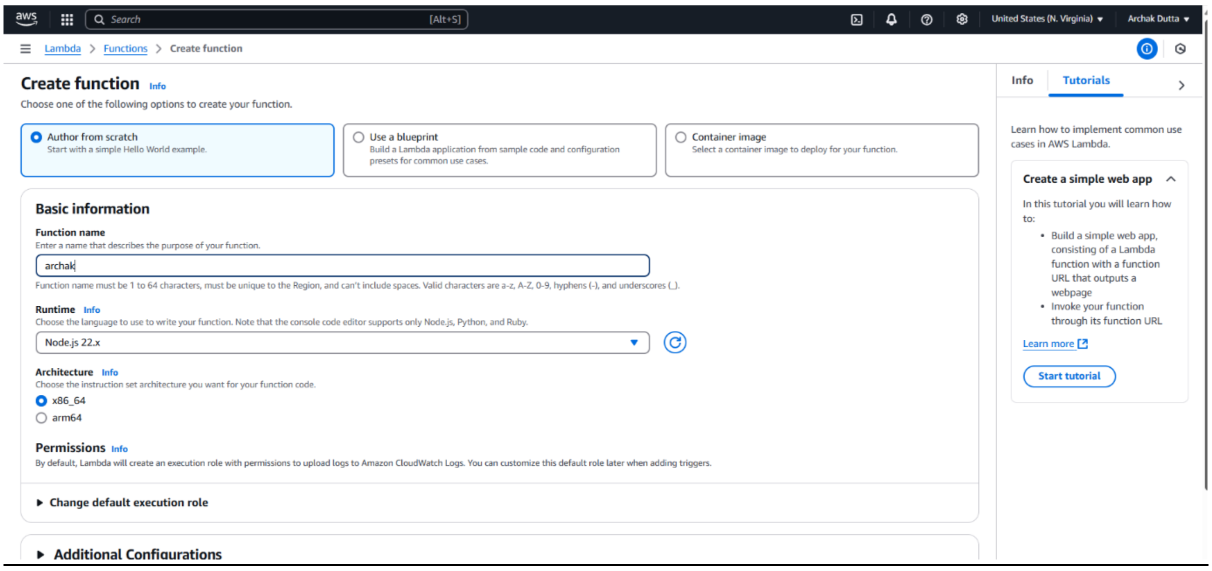
***Assignment No*: 15**

***Problem Statement*: Create a Serverless computing service.**

Solution: To create a Serverless computing service, the steps are-

1.Search lambda on AWS console and select it.Then click on **create function** and select **Author from scratch.** Provide the **function name**, **Runtime** as Node.js.22.x, **Architecture** as x86_64.



Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

- (●) 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

Function name: archak

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime: Node.js 22.x

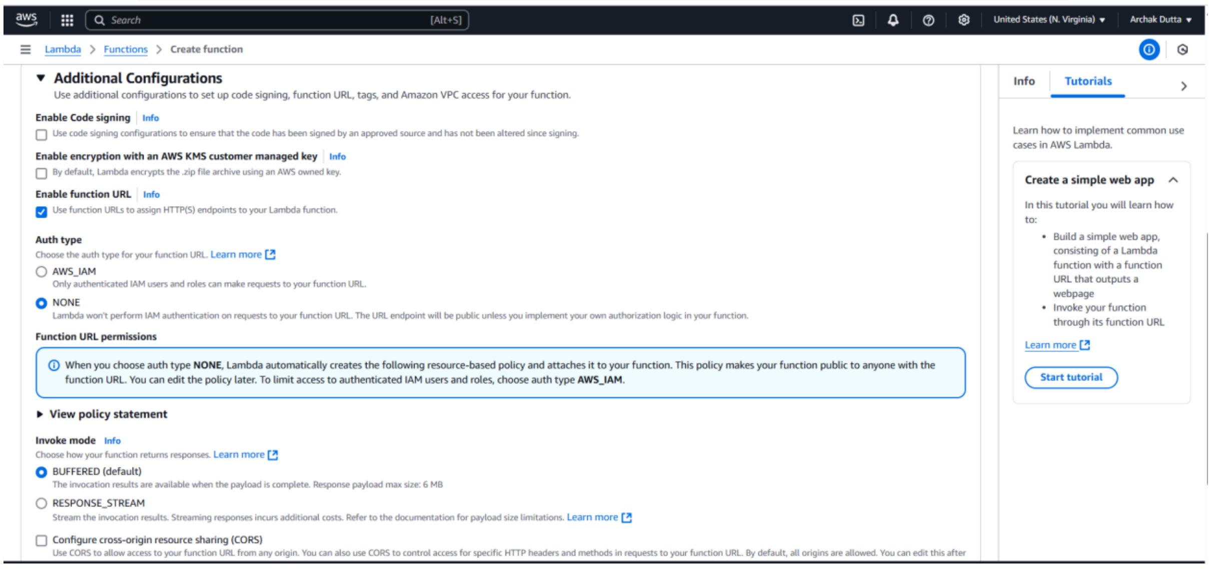
Architecture: (●) x86_64 ( ) arm64

Permissions: 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

▸ Additional Configurations

2. Under **Additional Configurations**, select **Enable function URL**, **Auth type** as NONE, **Invoke mode** as BUFFERED(default).Then click on **create function**.



Additional Configurations

Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

Enable Code signing: [ ] Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

Enable encryption with an AWS KMS customer managed key: [ ] By default, Lambda encrypts the .zip file archive using an AWS owned key.

Enable function URL: [✓] Use function URLs to assign HTTP(S) endpoints to your Lambda function.

Auth type:
- ( ) 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.

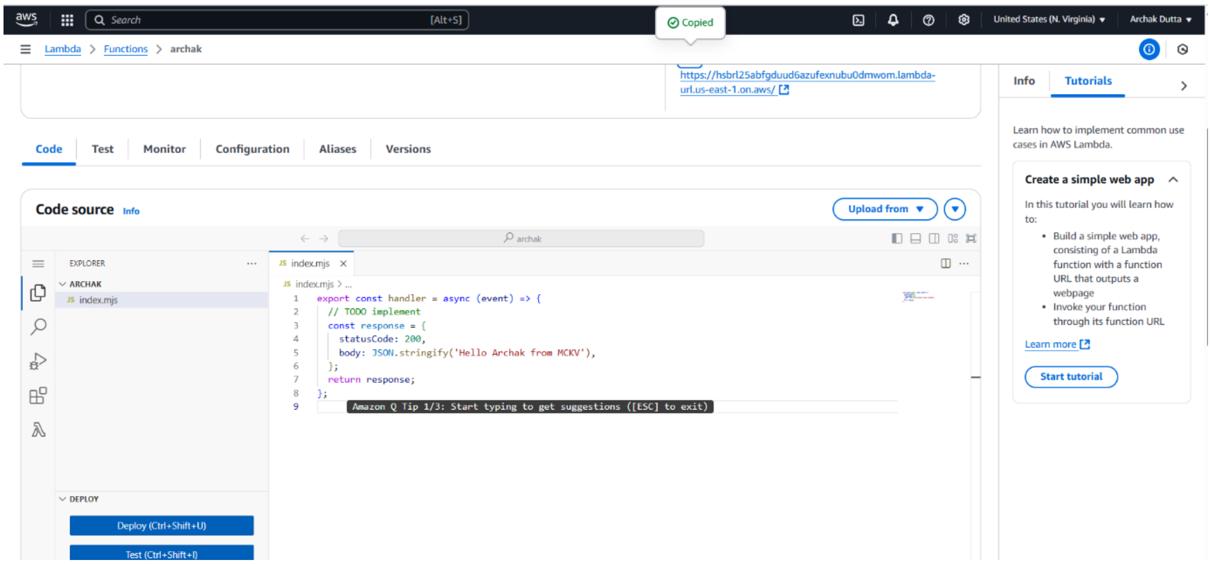
Function URL permissions: When you choose auth type **NONE**, Lambda automatically creates the following resource-based policy and attaches it to your function. This policy makes your function public to anyone with the function URL. You can edit the policy later. To limit access to authenticated IAM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

▸ View policy statement

Invoke mode:
- (●) BUFFERED (default) — The invocation results are available when the payload is complete. Response payload max size: 6 MB
- ( ) RESPONSE_STREAM — Stream the invocation results. Streaming responses incurs additional costs. Refer to the documentation for payload size limitations.

[ ] Configure cross-origin resource sharing (CORS)

3.Click on the created function and go the code part of it.There we need to modify some code and save it.



https://hsbrl25abfgduud6azufexnubu0dmwom.lambda-url.us-east-1.on.aws/

```
export const handler = async (event) => {
  // TODO implement
  const response = {
    statusCode: 200,
    body: JSON.stringify('Hello Archak from MCKV'),
  };
  return response;
};
```

Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)

Learn how to implement common use cases in AWS Lambda.

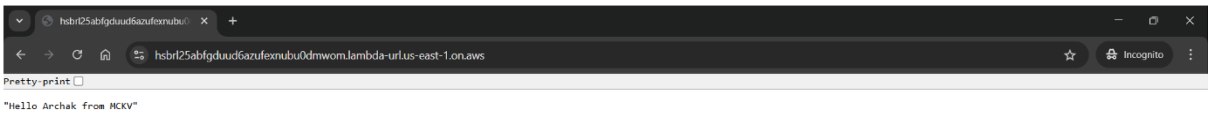
Create a simple web app

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

4.Now using the function URL we can see our modified code’s output on a incognito browser.


hsbrl25abfgduud6azufexnubu0
hsbrl25abfgduud6azufexnubu0dmwom.lambda-url.us-east-1.on.aws
Incognito
Pretty-print

```
"Hello Archak from MCKV"
```