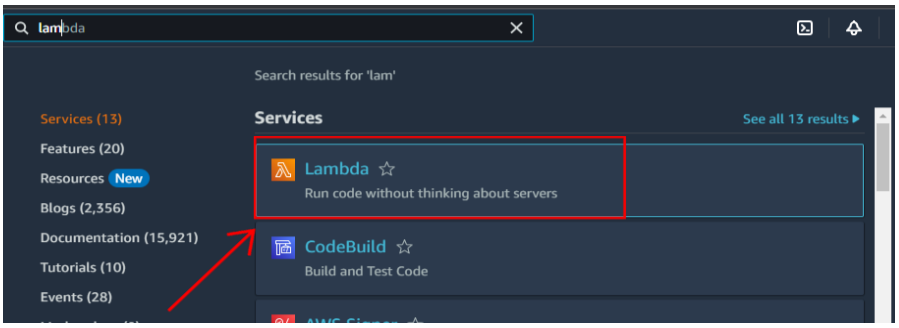
ASSIGNMENT NO-15

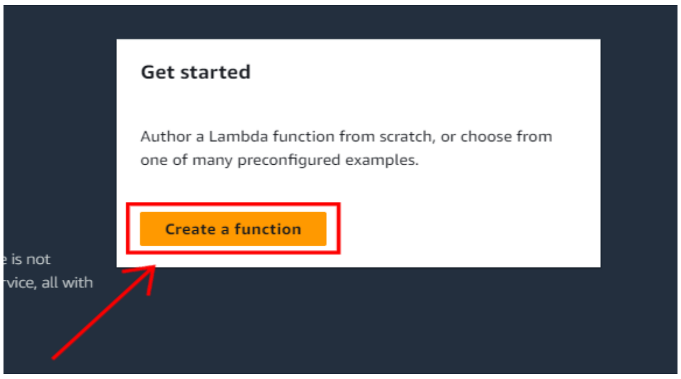
Problem Statement:- **Create server less computing service**.

Steps:-

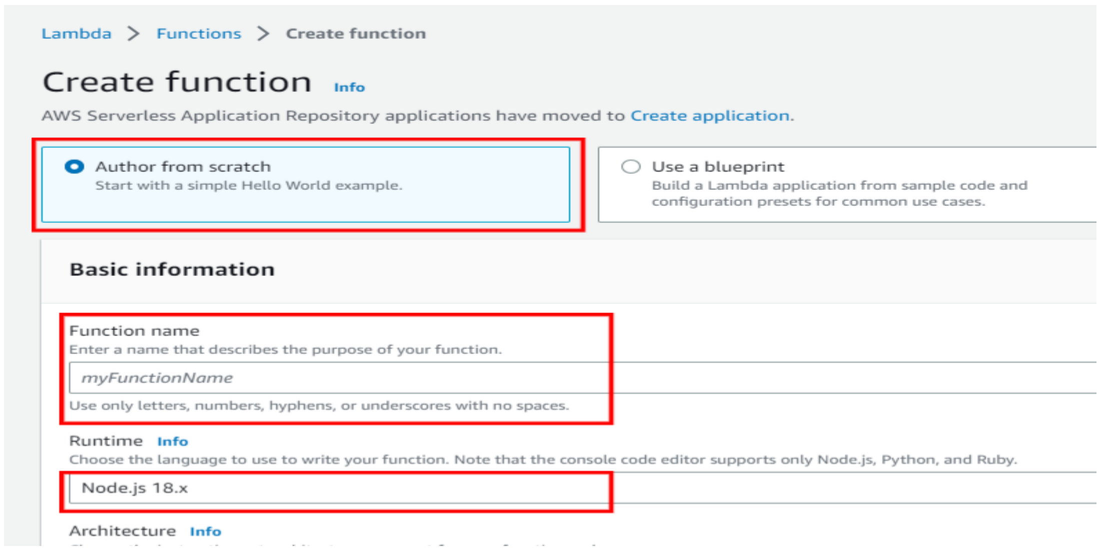
1. **Sign in.** Sign in to your **AWS account.**
2. Search **lambda** and **click on it**.



3. Click on **Create a function**.


Get started
Author a Lambda function from scratch, or choose from one of many preconfigured examples.
Create a function

4. Set the **Create function info as Author from scratch**, give the **name of the function** and set the **runtime info as Node.js.18.x**.



Lambda > Functions > Create function

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.

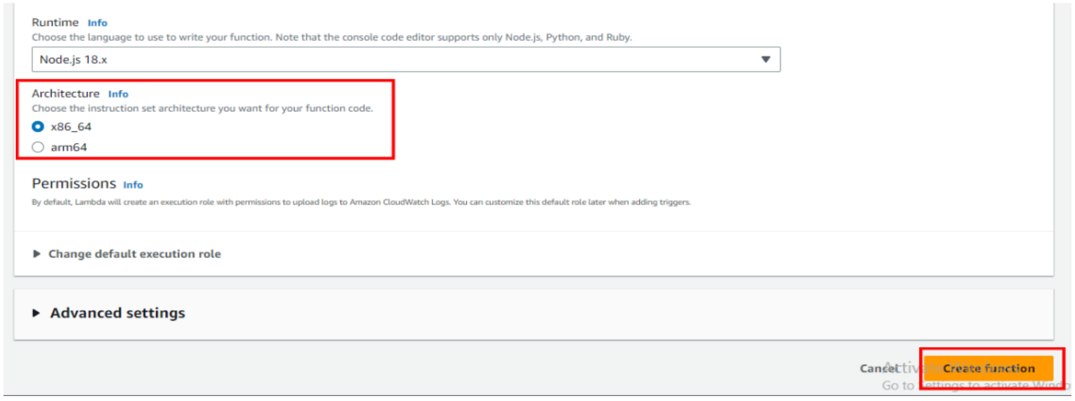
Basic information

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

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

5. Set the **Architecture info as x86_64** and click on **Create function**.

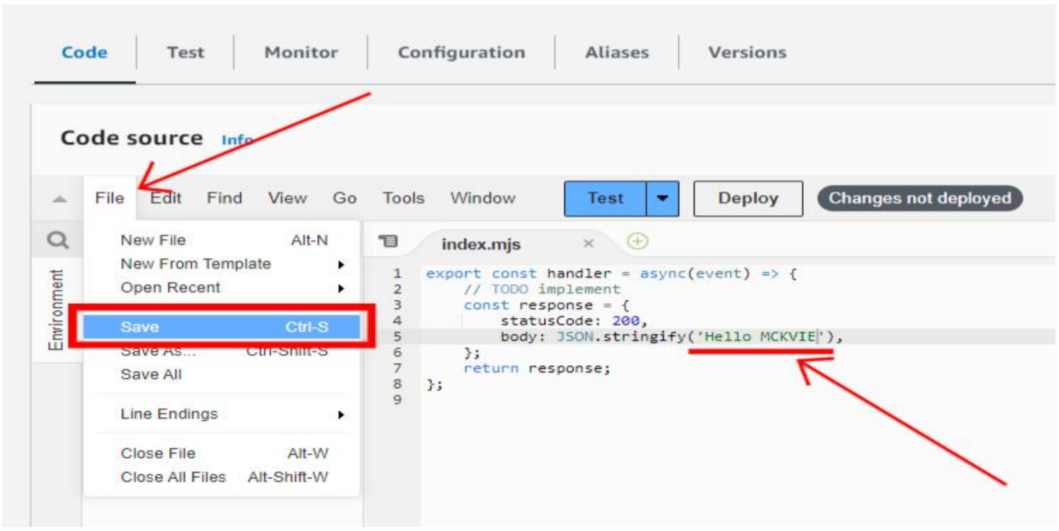


Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Node.js 18.x

Architecture Info
Choose the instruction set architecture you want for your function code.
[✓ x86_64] arm64

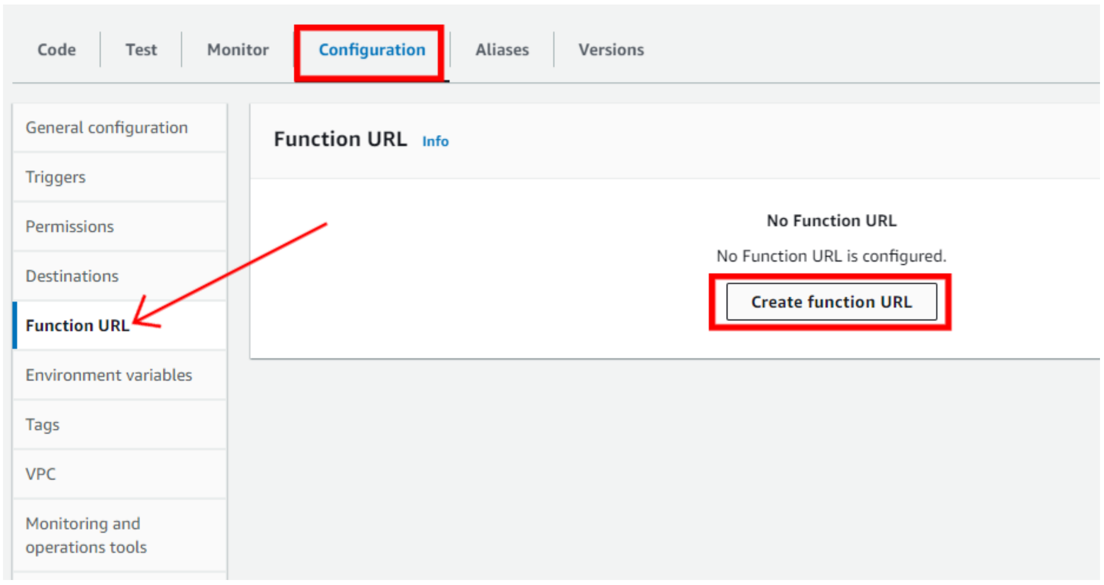
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

6. Now your function is created successfully, now **scroll down in the code source**, **edit the code as per your requirement** and **click on file** and **click on save** of save the edited code.

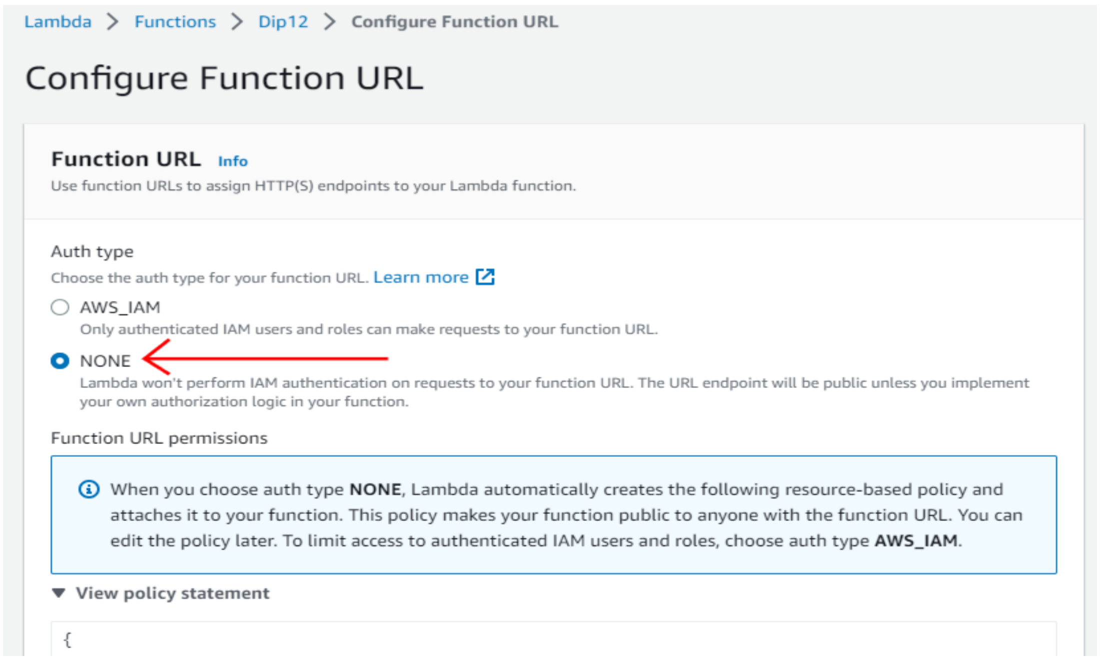


```
export const handler = async(event) => {
    // TODO implement
    const response = {
        statusCode: 200,
        body: JSON.stringify('Hello MCKVIE'),
    };
    return response;
};
```

7. Now go to **configuration**, then click on **Function URL** and after that click on **Create Function URL**.



8. Set the **Auth type as NONE**.



Lambda > Functions > Dip12 > Configure Function URL

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.

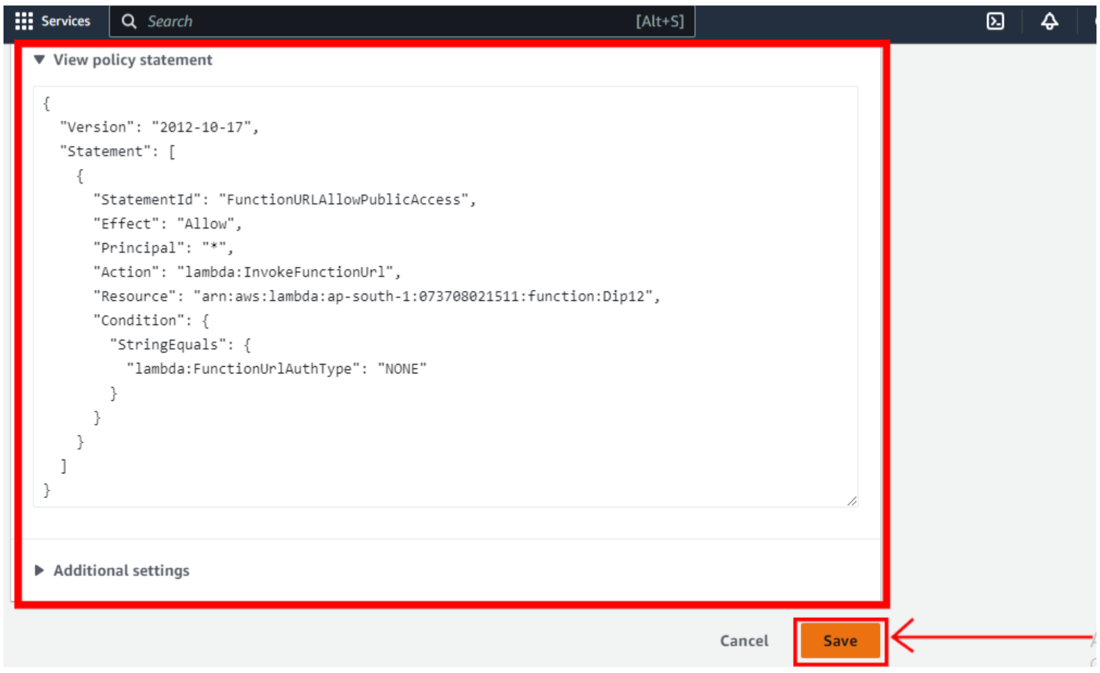
● NONE
Lambda won't perform IAM authentication on requests to your function URL. The URL endpoint will be public unless you implement your own authorization logic in your function.

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

9. Now click **save**.



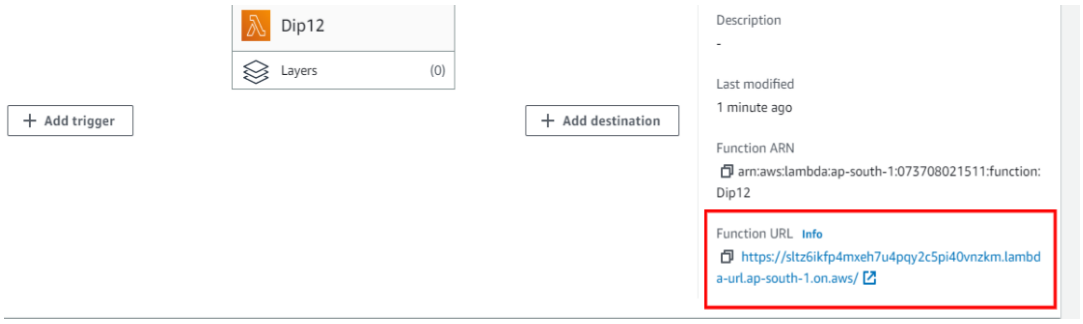
▼ View policy statement

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "StatementId": "FunctionURLAllowPublicAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "lambda:InvokeFunctionUrl",
      "Resource": "arn:aws:lambda:ap-south-1:073708021511:function:Dip12",
      "Condition": {
        "StringEquals": {
          "lambda:FunctionUrlAuthType": "NONE"
        }
      }
    }
  ]
}
```

▶ Additional settings

Cancel Save

10. Now **copy the Function URL** and open it on a new tab.



Dip12

Layers (0)

+ Add trigger

+ Add destination

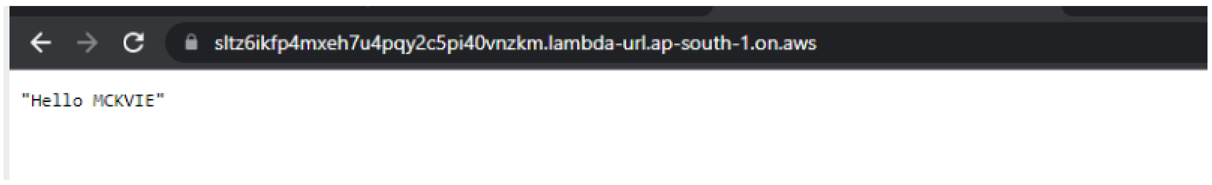
Description
-

Last modified
1 minute ago

Function ARN
arn:aws:lambda:ap-south-1:073708021511:function:Dip12

Function URL Info
https://sltz6ikfp4mxeh7u4pqy2c5pi40vnzkm.lambda-url.ap-south-1.on.aws/

11. The Function URL is working.


sltz6ikfp4mxeh7u4pqy2c5pi40vnzkm.lambda-url.ap-south-1.on.aws

```
"Hello MCKVIE"
```