

## Assignment – 15

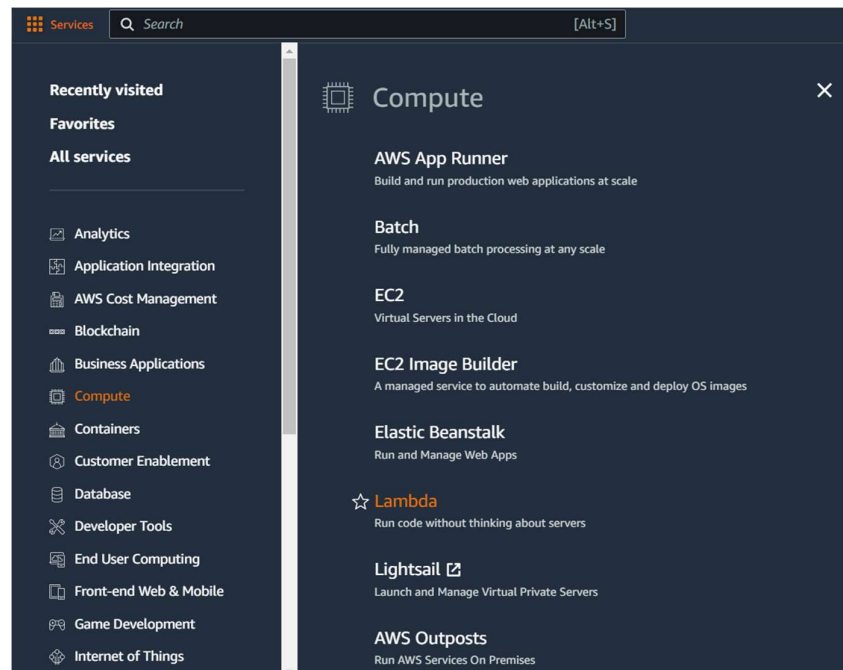
### Problem Statement: -

Create serverless computing services.

### Steps for create serverless computing services: -

1. Open the **Amazon Web Services** (<https://aws.amazon.com/console/>) home page.
2. Log in your **AWS Management Console** account.
3. Go **Services** and click on **Compute**.
4. Next go to **Lambda**.
5. Now click on **Create a function**.
6. Fill Create function info.
7. Now go to **Code** section and you can modify **index.mjs** file.
8. Click on **Test**. Configure test event and **Save**. After that again click on **Test**.
9. Now click on **Deploy**.
10. After that go to **Configuration** and click on **Create function URL**.
11. In Configure Function URL, choose **Auth type** and **Save**.
12. Now click on **Function URL** and you can see result.

### Some snapshots of above process: -



Go to Lambda

#### Get started

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

Create a function

Click on Create a function

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.

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

### Basic information

**Function name**  
Enter a name that describes the purpose of your function.

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](#)  
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.

### Create Lambda function

**Code source** [Info](#)

Upload from ▾

File Edit Find View Go Tools Window **Test** Deploy Changes not deployed

Go to Anything (Ctrl-P)

Environment

- functionAB ▾
  - index.mjs

index.mjs

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

Execution results x

### Click on Test

### 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**

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**

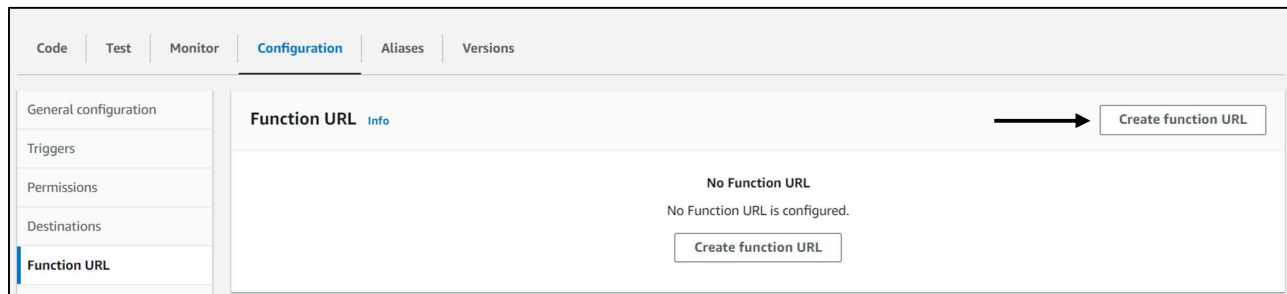
hello-world

**Event JSON** [Format JSON](#)

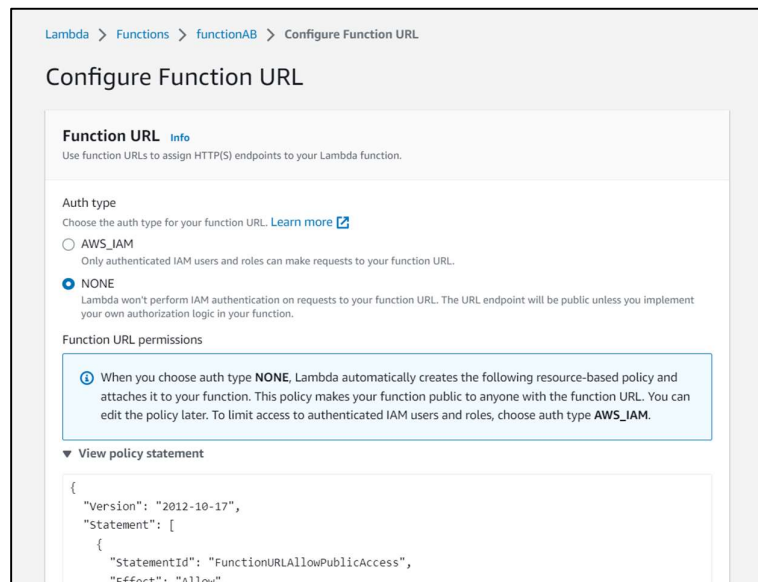
```
1 {  
2   "event": "  
3   "  
4   "  
5
```

Cancel Save

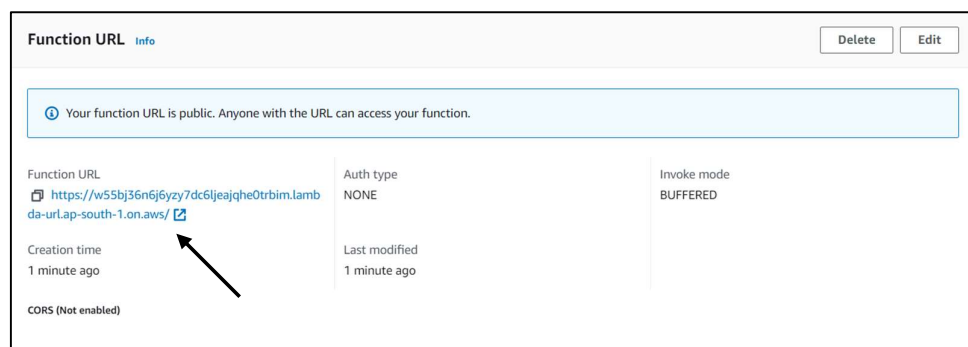
### Configure test event



Click on Create function URL



Configure Function URL



Click on Function URL



Deploy the project