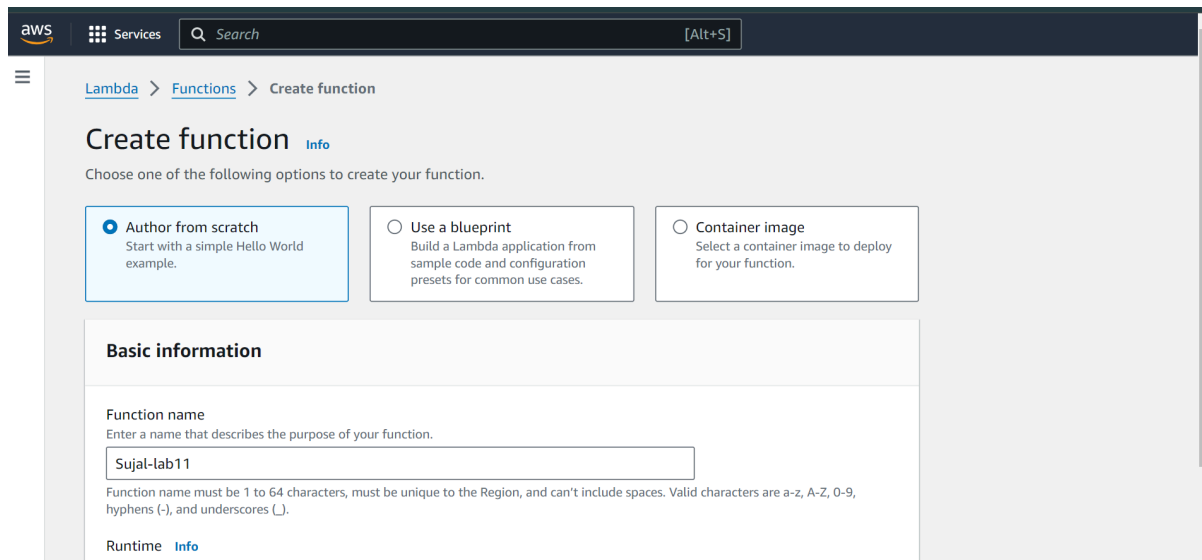


Name:Sujal.S.Tekwani  
Roll No:63  
Class:D15B

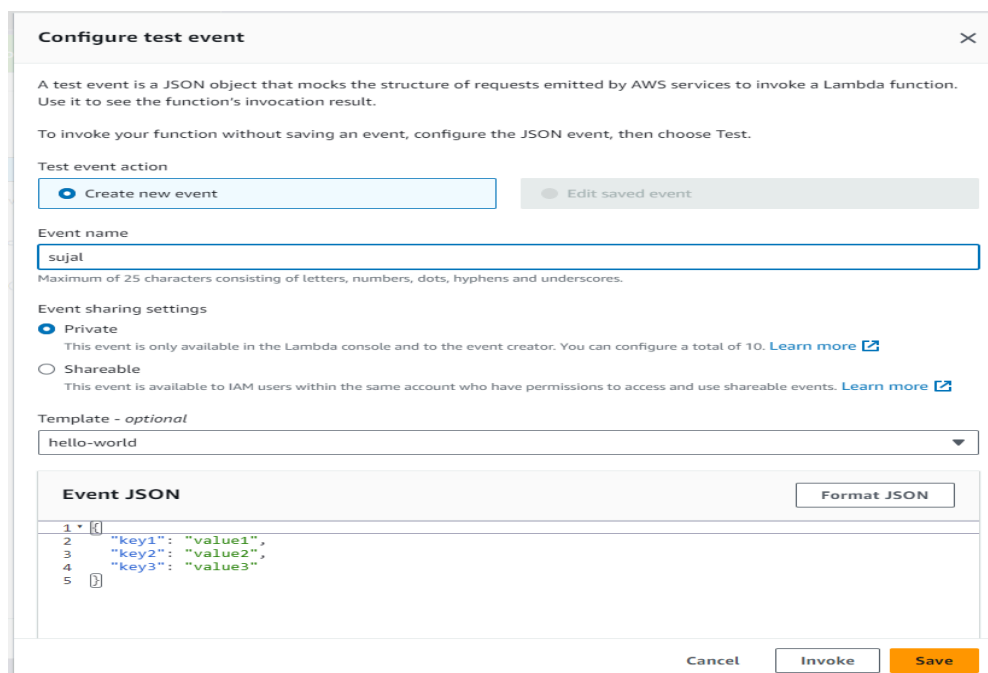
## Adv.DevOps Lab 11

### Function Creation:-



The screenshot shows the AWS Lambda 'Create function' page. The breadcrumb navigation is 'Lambda > Functions > Create function'. The title is 'Create function' with an 'Info' link. Below the title, it says 'Choose one of the following options to create your function.' There are three radio button options: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Author from scratch' option has a description: 'Start with a simple Hello World example.' Below the options is a section titled 'Basic information'. It contains a 'Function name' field with the value 'Sujal-lab11' and a description: 'Enter a name that describes the purpose of your function.' Below the field, it says '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 (\_).' There is also a 'Runtime' section with an 'Info' link.

### Configuration Changing:-

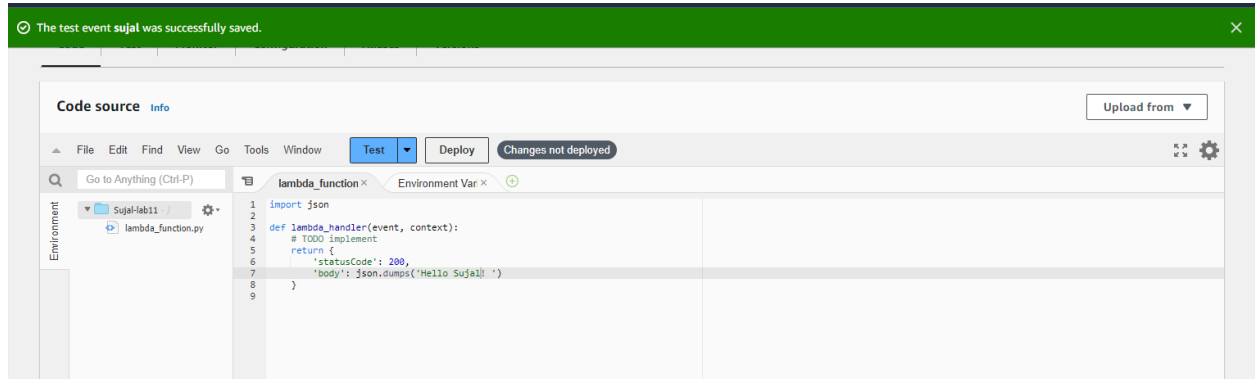


The screenshot shows the 'Configure test event' dialog box. It has a title bar with a close button. The main text says: 'A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.' Below this, it says: 'To invoke your function without saving an event, configure the JSON event, then choose Test.' There are two radio button options for 'Test event action': 'Create new event' (selected) and 'Edit saved event'. Below this is an 'Event name' field with the value 'sujal' and a description: 'Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.' Below the field is the 'Event sharing settings' section with two radio button options: 'Private' (selected) and 'Shareable'. Below this is a 'Template - optional' dropdown menu with the value 'hello-world'. Below the dropdown is the 'Event JSON' section with a 'Format JSON' button. The JSON content is: 

```
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3"
}
```

 At the bottom of the dialog are three buttons: 'Cancel', 'Invoke', and 'Save'.

## Code Changing:-



## Deploying:-

**Basic settings** [Info](#)

Description - *optional*

**Memory** [Info](#)

Your function is allocated CPU proportional to the memory configured.

128

MB

Set memory to between 128 MB and 10240 MB

**Ephemeral storage** [Info](#)

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)

512

MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

**SnapStart** [Info](#)

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#).

None

Supported runtimes: Java 11, Java 17, Java 21.

**Timeout**

0

 min 

1

 sec

**Execution role**

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☒ Use an existing role

☐ Create a new role from AWS policy templates

**Existing role**

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

## Final Testing:-

