For the Lambda functions we're going to create today, let's do something a little more mathematical than a simple Hello World application. Let's pass a couple of numbers into the function, and have it return the sum, product, difference, and quotient of the two numbers.

Our input will look similar to that shown below, and we'll use this example to execute a test case against each of the Lambdas we create:

```
{
  "Number1": 10,
  "Number2": 20
}
```

We'll be looking to receive the following in response:

```
{
  "Number1": 10,
  "Number2": 20,
  "Sum": 30,
  "Product": 200,
  "Difference": 10,
  "Quotient": 0.5
}
```

1. Go to AWS Management Console
2. Go to Services->Lambda
3. Click on create function
4. Select create from scratch.
   Blueprints are readymade templates which you can use to deploy functions.
   Serverless Application Repository is a library by users who put their functions on a Github page. The repository pulls functions from there. You can improvise those functions and deploy them.
5. Select a Python runtime and create a new role. We do not need any special permissions for this function.
6. Click on create a new test event
7. Once the test is created, we can trigger it with the Test button.
8. Let's add some code to the Lambda function .

```
from __future__ import division
```

```
def lambda_handler(event, context):
  number1 = event['Number1']
  number2 = event['Number2']
  sum = number1 + number2
  product = number1 * number2
  difference = abs(number1 - number2)
  quotient = number1 / number2
  return {
     "Number1": number1,
     "Number2": number2,
     "Sum": sum,
     "Product": product,
     "Difference": difference,
     "Quotient": quotient
  }
```

9. Once you've entered the code, click on Save, and then click on the Test button to try it out.

10. In the configure test events, enter the following code

```
{
  "Number1": 10,
  "Number2": 20
}
```

11. Click on Save and Test the event

12. The execution result is shown on the screen

```
{
  "Product": 200,
  "Sum": 30,
  "Number2": 20,
  "Number1": 10,
  "Difference": 10,
  "Quotient": 0.5
```

}

13. This is one way of creating a function and invoking it. You can invoke Lambda functions using code, event triggers or manually doing it the way we have done it.