

😊 AWS Lambda calling S3

1. In this lab you are going to use AWS Lambda to call S3 buckets.
2. First you need to create an IAM role for Lambda. Add the below permissions in your role for this lab.

Permissions policy summary		
Policy name 🔗	Type	Attached as
AmazonS3ReadOnlyAccess	AWS managed	Permissions policy
AWSLambdaBasicExecutionRole	AWS managed	Permissions policy

3. Now you need to create a function in Lambda with Python as your runtime environment.

☒ 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.
 [↻](#)

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
☒ x86_64
☐ arm64

4. Now in the permission section, click on use an existing role and choose your role. After that click on create function.

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.

▼ Change default execution role

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

☐ Create a new role with basic Lambda permissions

☒ 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.

[↻](#)

[View the demo-lambda-role](#) on the IAM console.

5. Once your function is created then scroll down a little and you need to make some changes in your code.
6. After click on Deploy. This will deploy your code and you will be able to test your code.

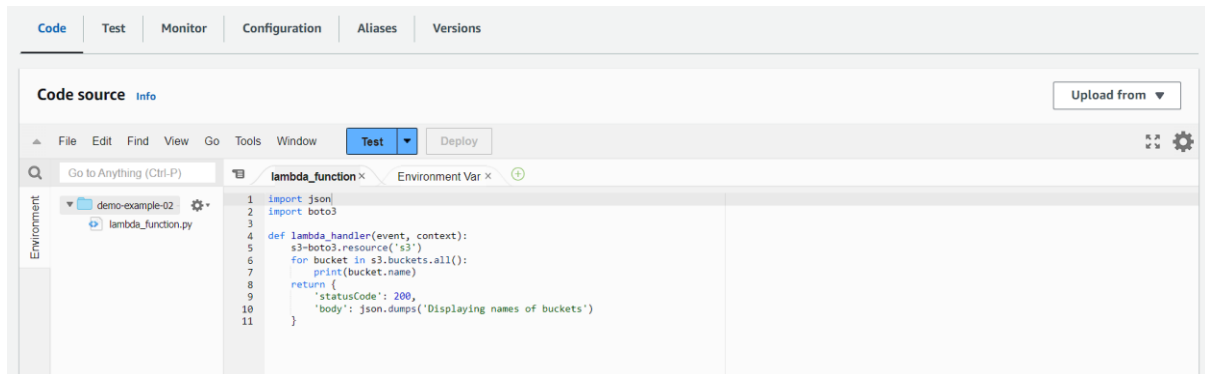
```
import json
import boto3
```

```
def lambda_handler(event, context):
```

```

s3=boto3.resource('s3')
for bucket in s3.buckets.all():
    print(bucket.name)
return {
    'statusCode': 200,
    'body': json.dumps('Displaying names of buckets')
}

```



7. Now you need to click on Test and create and test event.
8. Once you test event is create then click on test. You will see an output which says Displaying names of bucket.
9. You will the name of your buckets.

