

Lab 1 – Introduction to Lambda

Important Lab Notes:

- All labs build upon the resources created in prior labs
- You will incur charges in the AWS console. When you complete the labs, remove any resources created to avoid charges.
- The lab is written to use the default VPC in the N. Virginia region. If the default VPC has been removed, re-create it to complete the labs.

Create a Basic Lambda Function

1. Log into the AWS console
2. Go to <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/begin>
3. Click the **Python** tab
4. Click **Run**
The code shown is being executed by a Lambda function
5. Click **Next: Lambda Responds to Events**
 - a. Click the camera a repeatedly.
Notice more functions run as more events call on Lambda functions. It scales itself!
6. Click **Next: Scale Seamlessly**
 - a. Click the camera a repeatedly.
Notice that more invocations results in a higher price.
There are no charges until we exceed 1 million because of the free tier.
Exam tip: Limit of 1000 concurrent executions per region (Can be increased)
7. Click **Create a Function**
8. Click **Use a blueprint**
9. Choose the **Hello world** function for **python**
Note: Do not choose the Nodejs function
10. Function name: **Lambda**
11. Choose **Create a new role with basic Lambda permissions**
This role will allow the Lambda function to write to CloudWatch Logs
12. Click **Create Function**

13. Double-click `lambda_function.py` to view your code

The screenshot shows the AWS Lambda function editor interface. The top navigation bar includes File, Edit, Find, View, Go, Tools, Window, Test (which is highlighted with a red box), and Deploy. Below the navigation bar is a search bar labeled "Go to Anything (Ctrl-P)". A sidebar on the left shows a file tree with "lambda_function.py" selected, also highlighted with a red box. The main content area displays the Python code for the function:

```
1 import json
2
3 print('Loading function')
4
5
6 def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
```

14. Click **Test**

15. Event Name: Event

16. Click **Save**

17. Click **Test** again

18. It doesn't look like much...but your code has executed.

The screenshot shows the AWS Lambda function test results. At the top, there are tabs for "lambda_function." and "Execution result". The "Execution result" tab is active and shows the following details:

- Test Event Name: RickCrisciEvent
- Status: Succeeded
- Max memory used: 36 MB
- Time: 5.00 ms

Under "Response", it shows "value1".

Under "Function Logs", it shows the following log output:

```
Loading function
START RequestId: e3d3db2a-133f-4462-8ca3-d5b58072db1d Version: $LATEST
value1 = value1
value2 = value2
value3 = value3
END RequestId: e3d3db2a-133f-4462-8ca3-d5b58072db1d
REPORT RequestId: e3d3db2a-133f-4462-8ca3-d5b58072db1d Duration: 5.00 ms Billed Duration: 5 ms Memory Size: 128 MB Max
Request ID
e3d3db2a-133f-4462-8ca3-d5b58072db1d
```

Notice the Duration and Memory Size. These determine how much the function costs.

19. Click **Test** again

Notice the Duration is lower the next time you run it.

According to AWS “Since the execution environment already exists and it's not necessary to download the code and run the initialization code. This is called a “warm start”.

According to an analysis of production Lambda workloads, cold starts typically occur in under 1% of invocations.”

20. Click **Configuration**, then on **Edit**

- Notice we can change the amount of memory allocated to the function.
- Notice that the maximum timeout is 15 minutes
- Click **Cancel**

21. Click **Monitor**

- Review the CloudWatch metrics that can be viewed here

22. Under **Monitor**, click **View CloudWatch logs**

- This opens a new browser tab.

- b. Click the Log stream to review the output of the CloudWatch Logs for your Lambda function.
- c. Close the CloudWatch browser tab to return to the Lambda tab.

23. Click **Code** and choose the **lambda_function** tab

See the screen shot below if you are having trouble finding this.

- a. Modify the last two lines of code as follows:

```
#return event['key1'] # Echo back the first key value
raise Exception('Something went wrong')
```

Notice I added a # to the first line and removed it from the second line.

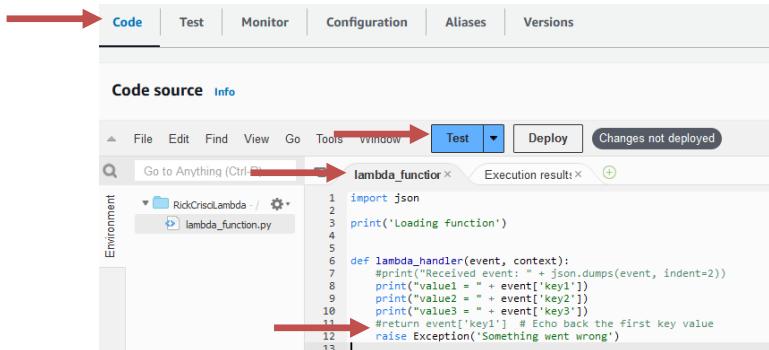
See the screen shot below for more details.

- b. Click **Deploy**

This pushes the changes you just made to the Lambda function. The next time we execute it, the code will reflect these changes.

- c. Click **Test**

- d. Note that in the execution results we now have an error message because the code was broken.



24. At the top right click **Actions**

25. Click **Delete Function**

26. Click **Delete**