Simulation: Getting Started with AWS Lambda

Simulation overview

In this simulation, you create and configure an AWS Lambda function to be initiated when an image is uploaded to an Amazon Simple Storage Service (Amazon S3) bucket. You review related logs to determine how the function is performing. You modify the function configuration to optimize for performance.

Objectives

After completing this simulation, you should be able to do the following:

- Create a Lambda function.
- Configure an Amazon S3 trigger on a Lambda function.
- Observe logs and metrics for a Lambda function to determine performance.
- Size and configure a Lambda function for optimal performance.

Duration

This simulation requires approximately **45 minutes** to complete.

Task 1: Creating a Lambda function

1. On the **AWS Management Console**, in the **Search** box, enter **Lambda,** and then press Enter on your keyboard.
2. From the search results, choose **Lambda**.
3. Choose **Create function**.
4. For the **Function name**, enter **resize_image**, and then press Enter on your keyboard.
5. From the **Runtime** dropdown list, choose **Python 3.9**.
6. Expand the **Change default execution role** section.
7. Choose **Use an existing role**.
8. From the **Existing role** dropdown list, choose **ResizeImageLambdaRole**.
9. Choose **Create function**.

10. Select the **scroll bar** to scroll to the **Layers** section.
11. Choose **Add a layer**.
12. Choose **Custom layers**.
13. From the **Custom layers** dropdown list, choose **PillowPythonLambdaLayer**.
14. From the **Version** dropdown list, choose the available option.
15. Choose **Add**.
16. Select the scroll bar to scroll to the **Code source** section.
17. Press Shift+S on your keyboard to switch to a virtual version of these simulation instructions.

18. Press Shift+C to select the following code in the simulation instructions.

```
import boto3
import os
import sys
import uuid
from urllib.parse import unquote_plus
from PIL import Image
import PIL.Image

s3_client = boto3.client('s3')

def resize_image(image_path, resized_path):
  with Image.open(image_path) as image:
    image.thumbnail(tuple(x / 2 for x in image.size))
    image.save(resized_path)

def lambda_handler(event, context):
  print('Begin resizing image')
  for record in event['Records']:
    bucket = record['s3']['bucket']['name']
    key = unquote_plus(record['s3']['object']['key'])
    print(bucket)
    print(key)
    tmpkey = key.replace('/', '')
    download_path = '/tmp/{}{}'.format(uuid.uuid4(), tmpkey)
    upload_path = '/tmp/resized-{}'.format(tmpkey)
    s3_client.download_file(bucket, key, download_path)
    resize_image(download_path, upload_path)
    s3_client.upload_file(upload_path, os.environ['RESIZED_BUCKET'], key)
  print('Resizing image complete')
```

19. Choose (right-click) the selected text to open the context menu, and copy the code.
20. Press Shift+S on your keyboard to switch back to the AWS Management Console.
21. Press Shift+C on your keyboard to copy the text in the **Code source** section.

22. Choose (right-click) the selected text to open the context menu to paste and replace the default source code in the lambda_function.py file.
23. To deploy your Lambda function, choose **Deploy**.
24. Choose the **Configuration** tab.
25. Choose **Edit**.
26. For **Memory**, enter **512** MB, and then press Enter on your keyboard.
27. Choose **Save**.
28. In the **Search** box, enter **S3**, and then press Enter on your keyboard.
29. From the search results, choose **S3**.
30. From the list of buckets, open the bucket with **resizedbucket** as part of the title.
31. Press Shift+C on your keyboard to select the name of the bucket, and then open (right-click) the context menu to copy its name.

    You will use it as the value later.

32. In the **Search** box, enter **Lambda**, and then press Enter on your keyboard.
33. From the search results, choose **Lambda**.
34. Choose the **resize_image** function.
35. On the **Configuration** tab, choose **Environment variables**.
36. In **Environment variables** section, choose **Edit**.
37. Choose **Add environment variable**.
38. For **Key,** enter **RESIZED_BUCKET**, and press Enter on your keyboard.
39. For **Value,** open (right-click) the context menu, and paste in the value you retrieved.
40. Choose **Save**.

You have created a Lambda function.

41. Choose **Continue** to move on to task 2.

Task 2: Configuring an Amazon S3 trigger to invoke a Lambda function

In this task, you configure an Amazon S3 trigger on an existing S3 bucket and your Lambda function. The Lambda function resizes images and places them in another bucket.

42. In the **Function overview** section of the Lambda console, choose **Add trigger**.
43. In the **Trigger configuration** section, choose **S3** from the dropdown list.

44. For **Bucket**, choose the bucket with **original** in the name.
45. For **Event types**, choose **All object create events** if it is not already selected.
46. For **Recursive invocation,** select the check box to acknowledge the statement
47. Choose **Add**.
48. Select the scroll bar to scroll to the bottom of the page.
49. In the **Triggers** section, for the S3 trigger that was just created, expand **Details**.
50. Verify that the **Bucket arn** has **originalbucket** in the name and that **Event types** contains **s3-ObjectCreated**.

You have configured your Lambda function to be initiated when a new object is uploaded to the S3 bucket.

51. Choose **Continue** to move on to task 3.

Task 3: Uploading an image to the Amazon S3 bucket

In this task, you upload an image file to your bucket.

52. Choose (right-click) the link to open the context menu, choose **Save Link As...,** and then choose **Save** to download the file to your computer:

    - large-image.jpg

53. Press Shift+S on your keyboard to return to the AWS Management Console.
54. On the AWS Management Console, in the **Search** box, enter **S3**, and then press Enter on your keyboard.
55. From the search results, choose **S3**.
56. Choose the link for the bucket that has **original** in the name.
57. Choose **Upload**.
58. Choose **Add files**.
59. Choose the file that you downloaded, and choose **Open**.
60. Choose **Upload**.

Your file is uploaded to the bucket. Notice the file size. The size of the file is around 4.9 MB.

61. Choose **Close**.

When your Lambda function runs correctly, the image file that you uploaded is reduced in size and placed in the S3 bucket that you specified when you set the environment variable for RESIZED_BUCKET.

62. To return to the **Buckets** section on the Amazon S3 console, choose **Buckets**.
63. Choose the link for the bucket that has **resized** in the name.

Notice the file size. It's significantly reduced from the original size of 4.9 MB.

When you uploaded the image file, Amazon S3 initiated the resize_image Lambda function. Review the logs to see how your function performed.

64. In the **Search** box, enter **Lambda**, and then press Enter on your keyboard.
65. From the search results, choose **Lambda**.
66. Choose the link for the **resize_image** function.
67. Choose the **Monitor** tab.
68. Select the scroll bar to scroll to **CloudWatch Logs**.
69. In the **Recent invocations** table, choose the most recent row (row 1) to expand the details.

Notice the metrics that are recorded for each function invocation. In the actual AWS Management Console, you might have to wait for up to 1 minute for the data to be updated.

The **DurationInMS** column tells you how long your function ran for this invocation.

The first time that your Lambda function is invoked, the Lambda execution environment has to download your code and start a new execution environment. This process is called a cold start. The **@initDuration** metric in the **Recent invocations** details signifies the cold start time.

Note the **MemorySetInMB** column. The amount of memory that's available to your Lambda function can be adjusted to affect the performance of your Lambda function.

The amount of memory also determines the amount of virtual CPU available to a function. Adding more memory proportionally increases the amount of CPU, which increases the overall computational power available. If a function is CPU-, network-, or memory-bound, then changing the memory setting can dramatically improve its performance.

Notice how long it took your function to run. It could be faster. Adjust your Lambda function so that it runs faster.

70. Choose **Continue** to move on to task 4.

Task 4: Optimizing Lambda function memory for performance

In this task, you first adjust the memory to 1024 MB.

71. Choose the **Configuration** tab.
72. Choose **General configuration**.
73. Choose **Edit**.
74. For **Memory**, enter **1024** MB, and then press Enter on your keyboard.
75. Choose **Save**.
76. In the **Search** box, enter **S3**, and then press Enter on your keyboard.
77. From the search results, choose **S3**.
78. Choose the link for the bucket that has **original** in the name.
79. Choose **Upload**.
80. Choose **Add files**.
81. Choose the file that you downloaded, and chose **Open**.
82. Choose **Upload**.

Your file is uploaded to the bucket.

83. Choose **Close**.
84. In the **Search** box, enter **Lambda**, and then press Enter on your keyboard.
85. From the search results, choose **Lambda**.
86. Choose the link for the **resize_image** function.
87. Choose the **Monitor** tab.
88. Select the scroll bar to scroll to **CloudWatch Logs**.
89. In the **Recent invocations** table, choose the most recent row (row 1) to expand the details.

    Note how long it took your function to run.

90. Choose **Continue** to try the next memory adjustment.

Next, you adjust the memory to 2048 MB.

91. Choose the **Configuration** tab.
92. Choose **Edit**.
93. For **Memory**, enter **2048** MB, and then press Enter on your keyboard.
94. Choose **Save**.
95. In the **Search** box, enter **S3**, and then press Enter on your keyboard.
96. From the search results, choose **S3**.

97. Choose the link for the bucket that has **original** in the name.
98. Choose **Upload**.
99. Choose **Add files**.
100. Choose the file that you downloaded, and choose **Open**.
101. Choose **Upload**.
102. Choose **Close**.
103. In the **Search** box, enter **Lambda**, and then press Enter on your keyboard.
104. From the search results, choose **Lambda**.
105. Choose the link for the **resize_image** function.
106. Choose the **Monitor** tab.
107. Select the scroll bar to scroll to **CloudWatch Logs**.
108. In the **Recent invocations** table, choose the most recent row (row 1) to expand the details.

     Note how long it took your function to run.

109. Choose **Continue** to try the next memory adjustment.

Next, you adjust the memory to 3008 MB.

110. Choose the **Configuration** tab.
111. Choose **Edit**.
112. For **Memory**, enter **3008** MB, and then press Enter on your keyboard.
113. Choose **Save**.
114. In the **Search** box, enter **S3, and then** press Enter on your keyboard.
115. From the search results, choose **S3**.
116. Choose the link for the bucket that has **original** in the name.
117. Choose **Upload**.
118. Choose **Add files**.
119. Choose the file that you downloaded, and choose **Open**.
120. Choose **Upload**.
121. Choose **Close**.
122. In the **Search** box, enter **Lambda**, and then press Enter on your keyboard.
123. From the search results, choose **Lambda**.
124. Choose the link for the **resize_image** function.
125. Choose the **Monitor** tab.
126. Select the scroll bar to scroll to **CloudWatch Logs**.
127. In the **Recent invocations** table, choose the most recent row (row 1) to expand the details.

Notice how long it took your function to run. Your function now runs in approximately 500 milliseconds with 3008 MB of memory and an image that is 5 MB.

Summary

In this simulation, you created a Lambda function and configured an Amazon S3 trigger on that Lambda function. You uploaded an image to an S3 bucket, which initiated the Lambda function to resize the image and place it in another bucket. After reviewing the logs to see performance metrics, you optimized the Lambda function by increasing the available memory for the function.

Excellent work!

Simulation complete

Congratulations! You have completed the simulation.

Your feedback is welcome and appreciated.

If you would like to share any suggestions or corrections, provide the details in the AWS Training and Certification Contact Form.