

aws.qwiklabs.com

Architecting on AWS - Lab 5 - Build a Serverless Architecture | Qwiklabs

Qwiklabs

22-28 minutes



© 2022 Amazon Web Services, Inc. and its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [AWS Training and Certification](#).

Lab overview

AWS solutions architects increasingly adopt event-driven architectures to decouple distributed applications. Often, these events must be propagated in a strictly ordered manner to all subscribed applications. Using Amazon SNS topics and Amazon SQS queues, you can address use cases that require end-to-end message ordering, deduplication, filtering, and

encryption. In this lab you configure an S3 bucket to invoke an Amazon SNS notification whenever an object is added to an S3 bucket. You learn how to create and interact with SQS queues, and learn how to invoke a Lambda function using SQS. This scenario will help you understand how you can architect your application to respond to S3 bucket events using serverless services such as SNS, AWS Lambda, and SQS.

Objectives

After completing this lab, you should be able to:

- Understand the value of decoupling resources
- Understand the potential value of replacing EC2 instances with Lambda functions
- Create an Amazon SNS topic
- Create Amazon SQS queues
- Create event notifications in Amazon S3
- Create AWS Lambda functions using pre-existing code
- Trigger an AWS Lambda function from SQS Queues
- Monitor AWS Lambda S3 functions through Amazon CloudWatch Log

Prerequisites

This lab requires:

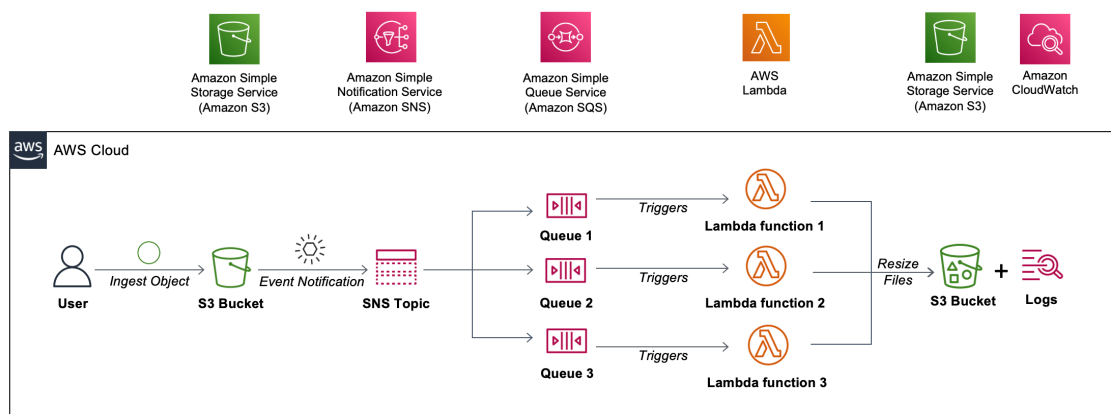
- Access to a notebook computer with Wi-Fi and Microsoft Windows, macOS, or Linux (Ubuntu, SuSE, or Red Hat)
- An internet browser such as Chrome, Firefox, or Microsoft Edge
- A plaintext editor

Scenario

Your team has been tasked with evaluating fully serverless architectures. Based on the proposed architecture to achieve loose coupling, there is an opportunity to replace EC2 Auto Scaling Groups with more cost effective Lambda functions.

Customer Care professionals take snapshots of products and upload them to our network. We store the images, then run Python scripts to resize the images after they are uploaded into the ingest S3 bucket. Uploading a file to the ingest bucket invokes an event notification to an SNS topic. SNS then distributes the notifications to three separate SQS queues. The initial design was to run EC2 instances in auto scaling groups for each resizing operation. However, based on your recommendations, you are replacing the EC2 instances with Lambda functions. The Lambda functions process the images in three formats and stores the output in an S3 bucket.

The following diagram shows the workflow:



The scenario workflow is as follows:

- 1 You upload an image file to an Amazon S3 bucket.
- 2 Uploading a file to the ingest folder in the bucket invokes an event notification to an SNS topic.

3 SNS then distributes the notifications to three separate SQS queues.

4 The Lambda functions processes the images in three formats and store the output in S3 bucket folder.

5 Validate the processed images in the S3 bucket folders and the logs in Amazon CloudWatch.

Duration

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

Start Lab

1. At the top of your screen, launch your lab by choosing Start Lab

This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing Open Console

This opens an AWS Management Console sign-in page.

3. On the sign-in page, configure:

- **IAM user name:**
- **Password:** Paste the value of **Password** from the left side of the lab page
- Choose Sign In

Do not change the Region unless instructed.

Common Login Errors

Error: You must first log out

Amazon Web Services Sign In

You must first log out before logging into a different AWS account.

To logout, [click here](#)

If you see the message, **You must first log out before logging into a different AWS account:**

- Choose **here**
- Close your browser tab to return to your initial lab window
- Choose Open Console again

Task 1: Create an Amazon SNS topic

Task 1.1: Create a Standard SNS topic

In this task you create an SNS topic and subscribe to an Amazon SNS topic.

4. On the Services menu, choose **Simple Notification Service**.
5. Expand the navigation menu by selecting the menu icon near the upper left corner.
6. Select **Topics** from the navigation menu.
7. Choose the **Create topic** button.

The **Create topic** page is displayed.

8. On the **Create topic** page, in the **Details section**, do the following:
 - For **Type**, choose *Standard*
 - **Name:** Input a unique SNS topic name. Such as followed by 4 random numbers.

9. Choose the Create topic button.

The topic is created and the **resize-image-topic-XXXX** page is displayed. The topic's Name, ARN, (optional) Display name, and Topic owner's AWS account ID are displayed in the Details section.

10. Copy the topic **ARN** and **Topic owner** values to a notepad, you need these values later in the lab.

- ARN example - *arn:aws:sns:us-east-2:123456789012:MyTopic*
- Topic owner - *123456789123* (12 digit number)

Task 2: Create three Amazon SQS queues and subscribe to SNS topic

Task 2.1: Create SQS Queue for Thumbnail

11. On the Services menu, choose **Simple Queue Service**.

12. On the SQS home page, choose the Create queue button.

The **Create queue** page is displayed.

13. On the **Create queue** page, in the **Details section**, do the following:

- For **Type**, choose *Standard* (Standard queue type is set by default.)
- **Name:**

14. The console sets default values for the queue **Configuration** parameters. Accept the default values.

15. Choose the Create queue button.

Amazon SQS creates the queue and displays a page with details about the queue.

Task 2.2: Subscribe the SQS queue to SNS topic

16. On the queue's detail page, choose **SNS subscriptions** tab
17. Choose the Subscribe to Amazon SNS topic button.
 - A new **Subscribe to Amazon SNS topic** page opens,
18. From **Specify an Amazon SNS topic available** menu, choose *resize-image-topic* SNS topic under **Use existing resource**

Note: If the SNS topic is not listed in the menu, choose **Enter Amazon SNS topic ARN** and then enter the topic's **ARN** that was copied earlier.
19. Choose the Save button.

Your SQS queue is now subscribed to the SNS topic named *resize-image-topic-XXXX*.

Task 2.3: Create two more SQS queues and subscribe them to the SNS topic

Repeat tasks 2.1 and 2.2 to create two more standard SQS Queues. One named *web-queue* and one named *mobile-queue* and subscribe them to the *existing* SNS topic named *resize-image-topic-XXXX*. Use the following configurations for the two additional queues.

20. Create SQS Queue for web sized images:
 - **Create queue with Name:**
21. Create SQS Queue for mobile sized images:
 - **Create queue with Name:**

Task 2.4: Verify the subscription

To verify the result of the subscription, publish to the topic and

then view the message that the topic sends to the queue.

22. Return to the AWS Management Console browser tab. On the Services menu, select **Simple Notification Service**.

23. In the left navigation pane, choose **Topics**.

24. On the **Topics** page, select **resize-image-topic-XXXX**.

25. Choose the Publish message button.

The console opens the **Publish message to topic** page.

26. In the Message details section, do the following:

- **Subject - optional:**

27. In the Message body section, do the following:

- Choose **Identical payload for all delivery protocols**, and then enter the message or any message of your choice.

28. In the **Message attributes** section, configure the following:

- **Type:** *String*
- **Name:**
- **Value:**

29. Choose the Publish message button.

The message is published to the topic, and the console opens the topic's detail page. To investigate the published message, navigate to Amazon SQS.

30. On the Services menu, select **Simple Queue Service**.

31. Choose any **queue** from the list.

32. Choose the Send and receive messages button.

33. On **Send and receive messages** page, in **Receive messages** section, choose the Poll for messages button.

34. Locate the **Message** section. Select any message populated in the list to see the **Details**, **Body** and **Attributes** of the message.

The Message Details box contains a JSON document that contains the subject and message that you published to the topic.

35. Choose the **Done** button. You have successfully published to a topic that sends notification messages to a queue.

Task 3: Create an Amazon S3 Event Notification to SNS

Task 3.1: Configure SNS access policy to allow Amazon S3 bucket to publish to a topic

36. On the Services menu, select **Simple Notification Service**.
37. Choose **Topics** from the navigation menu.
38. Choose the **resize-image-topic-XXXX** Topic.
39. Choose the Edit button.
40. Navigate to **Access policy** and expand it, if necessary.
41. In the **JSON Editor** section, replace the access policy.

Copy the code block below paste it into the JSON Editor section. Replace **Trusted Owner Value** with the **Topic Owner** value, and replace **Topic-ARN** with the SNS topic **ARN** values copied earlier in task 1. Make sure to leave the double quotes. There is four substitutions to make in total.

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
```

```
"Sid": "__default_statement_ID",
"Effect": "Allow",
"Principal": {
  "AWS": "*"
},
"Action": [
  "SNS:GetTopicAttributes",
  "SNS:SetTopicAttributes",
  "SNS:AddPermission",
  "SNS:RemovePermission",
  "SNS:DeleteTopic",
  "SNS:Subscribe",
  "SNS:ListSubscriptionsByTopic",
  "SNS:Publish",
  "SNS:Receive"
],
"Resource": "Topic-ARN",
"Condition": {
  "StringEquals": {
    "AWS:SourceAccount": "Trusted Owner
Value"
  }
}
},
{
  "Effect": "Allow",
  "Principal": {
    "Service": "s3.amazonaws.com"
  },
  "Action": "SNS:Publish",
  "Resource": "Topic-ARN",
  "Condition": {
```

```
        "StringEquals": {
            "AWS:SourceAccount": "Trusted Owner
Value"
        }
    }
}
]
```

42. Choose the Save changes button.

Task 3.2: Create a Single event notification on uploads into ingest S3 bucket

43. On the Services menu, select **S3**.

44. On the **Buckets** page, choose the bucket with a name like **xxxxxx-labbucket-xxxxxx**.

45. Choose **Properties** tab.

46. Navigate to **Event notifications** section.

47. Choose the Create event notification button.

48. In the General configuration section, do the following:

- **Event name:**
- **Prefix - optional:**

Note: In this lab, you need to set up a prefix filter so that you receive notifications only when files are added to a specific folder (**ingest**).

- **Suffix - optional:**

Note: In this lab, you need to set up a suffix filter so that you receive notifications only when **.jpg** files are uploaded.

49. In the **Event types** section, check **All object create events**

50. In the **Destination** section, do the following:

- For **Destination**: select *SNS Topic*
- For **Specify SNS topic**: select *Choose from your SNS topics*
- Under **SNS topic**, choose the *resize-image-topic-XXXX* SNS topic from the drop-down menu.

Or, if you prefer to specify an **ARN**, select **Enter ARN** and enter the ARN of the SNS topic copied earlier.

51. Choose the Save changes button.

Task 4: Create and configure three AWS Lambda functions

Task 4.1: Create a Lambda function to Generate thumbnail

In this task, you create an AWS Lambda function with an SQS trigger that reads an image from Amazon S3, resizes the image and then stores the new image in Amazon S3 bucket folder.

52. On the Services menu, select **Lambda**.

53. Choose the Create function button.

54. Choose **Author from scratch**

55. In the **Create function** window, configure:

- **Function name**:
- **Runtime**: *Python 3.7*
- Expand the **Change default execution role** section.
- **Execution role**: Select **Use an existing role**
- **Existing role**: Choose the role with the name like **XXXXX-LabExecutionRole-XXXXX**.

This **role** provides your Lambda function with the permissions it needs to access Amazon S3 and Amazon SQS.

Make sure to select Python 3.7 under **Other supported** runtime. If you select Python 3.8 under **Latest supported**, the code in this lab will fail.

56. Choose the Create function button.

A page is displayed with your function configuration.

Task 4.2: Configure the Lambda function - Add an SQS trigger and Upload python deployment package

AWS Lambda functions can be triggered automatically by activities such as data being received by Amazon Kinesis or data being updated in an Amazon DynamoDB database. For this lab, you trigger the Lambda function whenever a new object is pushed to your Amazon SQS queue.

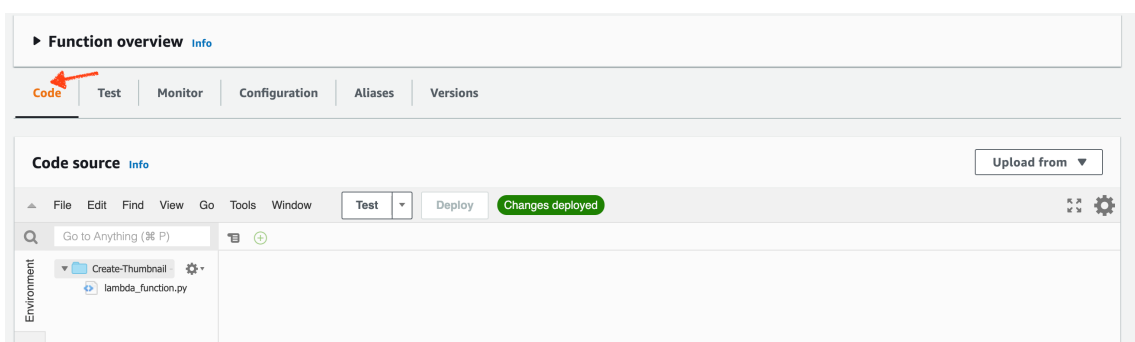
57. choose Add trigger then configure:

- **Select a trigger:** SQS
- **SQS Queue:** Select **thumbnail-queue**
- **Batch Size:**

58. Scroll to the bottom of the screen, then choose the Add button.

The **SQS trigger** is added to your **Function Overview** page. Now configure the Lambda function.

59. Select the **Code** tab as shown below:



60. Configure the following settings (and ignore any settings that are not listed):

- Download and save this zip file.

Right-click on the file name and download the zip file to your computer: [CreateThumbnail.zip](#)

- Select **Upload from** menu and select **.zip file**
- Choose **Upload** button and upload the downloaded zip file.
- Choose the Save button.

The *CreateThumbnail.zip* file contains the following Lambda function:

Do not copy this code -- it is just showing you what is in the Zip file.

```
import boto3
import os
import sys
import uuid
from PIL import Image
import PIL.Image
import json
import time
```

```
s3_client = boto3.client('s3')
s3 = boto3.resource('s3')
```

```
def resize_image(image_path, resized_path):
    with Image.open(image_path) as image:
```

```
        image.thumbnail((128, 128))
        image.save(resized_path)

def handler(event, context):
    for record in event['Records']:

        payload = record["body"]
        sqs_message=json.loads(str(payload))
        bucket_name =
json.loads(str(sqs_message["Message"]))
["Records"][0]["s3"]["bucket"]["name"]
        print(bucket_name)

key=json.loads(str(sqs_message["Message"]))
["Records"][0]["s3"]["object"]["key"]
        print(key)
        download_path = '/tmp/{}
{}'.format(uuid.uuid4(), key.split("/")[1])
        upload_path = '/tmp/resized-
{}'.format(key.split("/")[1])

        s3_client.download_file(bucket_name,
key, download_path)
        resize_image(download_path,
upload_path)
        s3.meta.client.upload_file(upload_path,
bucket_name, 'thumbnail/Thumbail-'+key.split("
/")[1])
```

61. Examine the above code. It is performing the following steps:

- Receives an Event, which contains the name of the incoming object (Bucket, Key)

- Downloads the image to local storage
- Resizes the image using the *Pillow* library
- Creates and uploads the resized image to a new folder

62. In the **Runtime settings** section, choose the Edit button.

- **Handler** enter:

`CreateThumbnail.handler`

63. Choose the Save button.

Make sure you set the **Handler** field to the above value, otherwise the Lambda function will not be found.

64. Select the **Configuration** tab.

65. Select **General configuration**,

66. Choose the Edit button.

- **Description** enter:

`Create a thumbnail-sized image`

Leave the other settings as default, but here is a brief explanation of these settings:

- **Memory** defines the resources that are allocated to your function. Increasing memory also increases CPU allocated to the function.
- **Timeout** sets the maximum duration for function execution.

67. Choose the Save button.

Your Lambda function has now been configured.

Task 4.3: Create and configure two more Lambda functions

Repeat tasks 4.1 and 4.2 to create and configure two more Lambda functions(for creating web and mobile image) with the

following configurations.

68. Create a Lambda function to Generate web image

- **Function name:**
- **Select a trigger:** SQS
- **SQS Queue:** Select **web-queue**
- **Batch Size:**
- Download and save this zip file.

Right-click on the file name and download the zip file to your computer: [CreateWebImage.zip](#)

- In the **Runtime settings** section, choose the Edit button.
- **Handler** enter:

`CreateWebImage.handler`

- **Description** enter:

Create a web-sized image

69. Create a Lambda function to Generate mobile image

- **Function name:**
- **Select a trigger:** SQS
- **SQS Queue:** Select **mobile-queue**
- **Batch Size:**
- Download and save this zip file.

Right-click on the file name and download the zip file to your computer: [CreateMobileImage.zip](#)

- In the **Runtime settings** section, choose the Edit button.
- **Handler** enter:

`CreateMobileImage.handler`

- **Description** enter:

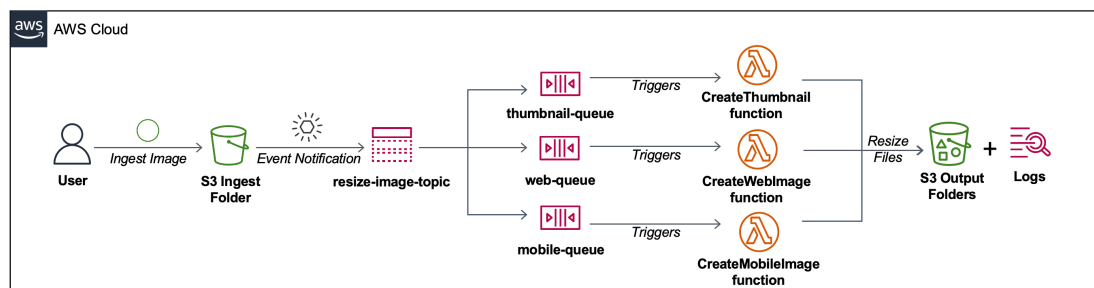
Create a mobile-sized image

With these configurations, you should have all of your three Lambda functions ready for the test.

Task 5: Upload an object to Amazon S3 bucket

Task 5.1: Upload an image to S3 bucket folder for processing

The following diagram shows the workflow:



Upload a picture for testing what you have built.

70. Choose to download **one** image from the options below:

- Right-choose this [link](#) and download the picture to your computer: [AWS.jpg](#)
- Right-choose this [link](#) and download the picture to your computer: [MonaLisa.jpg](#)
- Right-choose this [link](#) and download the picture to your computer: [HappyFace.jpg](#)
- Name your file similar to **InputFile.jpg**.

Firefox users: Make sure the saved filename is *InputFile.jpg* (not *.jpeg*).

71. In the **S3 Management Console**, select the **xxxxx-labbucket-**

xxxxxx bucket.

72. Inside the bucket, select the **ingest/** folder.

73. Choose the Upload button.

74. In the **Upload** window, choose Add files.

75. Browse to and select the **XXXXXX.jpg** picture you downloaded.

76. Choose the Upload button.

77. You will get an upload confirmation.

Task 6: Validate the processed file

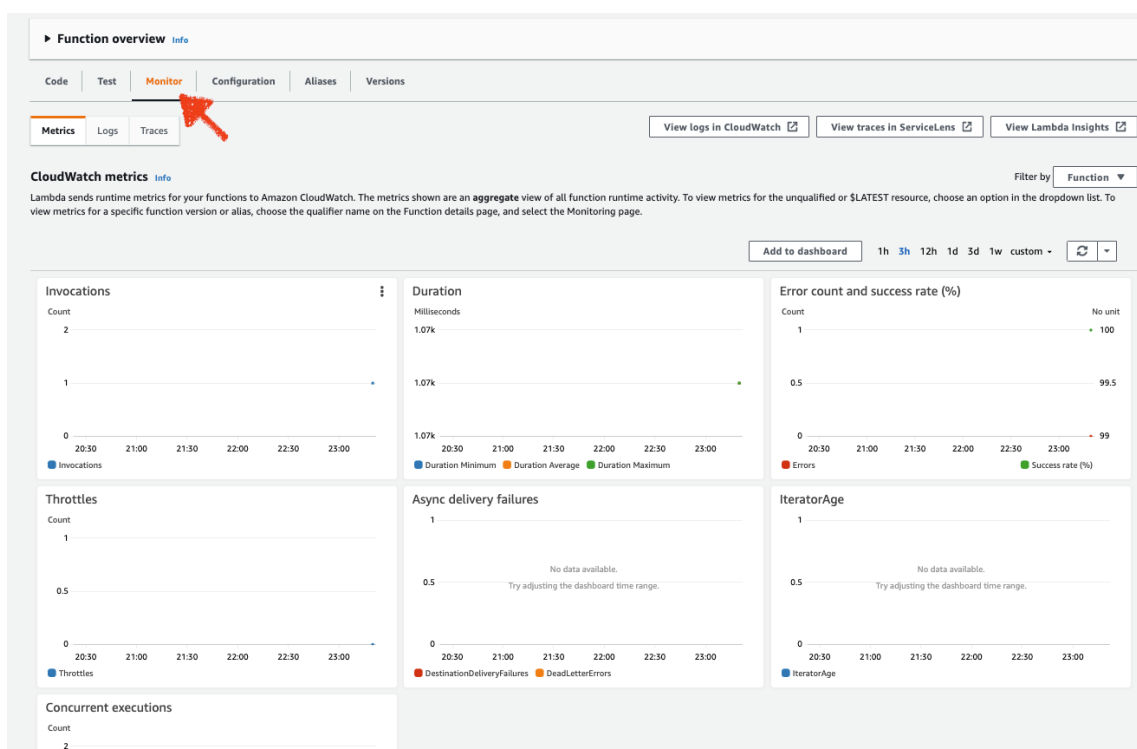
Task 6.1: View Amazon CloudWatch Logs for Lambda activity

You can monitor AWS Lambda functions to identify problems and view log files to assist in debugging.

78. On the Services menu, select **Lambda**.

79. Select one of your **Create-** functions.

80. Select the **Monitor** tab as shown below:



The console displays graphs showing:

- **Invocations:** The number of times that the function was invoked.
- **Duration:** The average, minimum, and maximum execution times.
- **Error count and success rate (%):** The number of errors and the percentage of executions that completed without error.
- **Throttles:** When too many functions are invoked simultaneously, they are throttled. The default is 1000 concurrent executions.
- **Async delivery failures:** The number of errors that occurred when Lambda attempted to write to a destination or dead-letter queue.
- **Iterator Age:** Measures the age of the last record processed from streaming triggers (Amazon Kinesis and Amazon DynamoDB Streams).
- **Concurrent executions:** The number of function instances that are processing events.

Log messages from Lambda functions are retained in **Amazon CloudWatch Logs**.

81. Choose the View logs in CloudWatch button.

82. Select the **Log Stream** that appears.

83. Expand each message to view the log message details.

The REPORT line provides the following details:

- **RequestId:** The unique request ID for the invocation.
- **Duration:** The amount of time that your function's handler method spent processing the event.

- **Billed Duration:** The amount of time billed for the invocation.
- **Memory Size:** The amount of memory allocated to the function.
- **Max Memory Used:** The amount of memory used by the function.
- **Init Duration:** For the first request served, the amount of time it took the runtime to load the function and run code outside of the handler method.

In addition, any logging messages or print statements from the functions are displayed in the logs. This assists in debugging Lambda functions.

Task 6.2: Validate the S3 bucket for processed files

84. On the Services menu, select **S3**.

85. Select **xxxxx-labbucket-xxxxx** to enter the bucket.

86. You should now see these 3 new folders,

- **thumbnail**
- **web**
- **mobile**

87. Navigate through these folders to find the **Resized Images** (e.g. *Thumbnail-AWS.jpg*, *WebImage-HappyFace.jpg*, *MobileImage-MonaLisa.jpg*)

If you find the resized image here, you have successfully resized the image from its original to 3 different formats.

Optional Tasks

Note: Challenge tasks are optional and are provided in case you have extra time remaining in your lab. You may complete

the optional tasks or skip to the end of the lab [here](#).

- *Optional Task 1:* Create a lifecycle configuration to delete files in the ingest bucket after 30 days.

Note: If you have trouble completing the optional task, refer to the [Optional Task 1 Solution](#) Appendix section at the end of the lab.

- *Optional Task 2:* Add an SNS e-mail notification to the existing SNS topic.

Note: If you have trouble completing the optional task, refer to the [Optional Task 2 Solution](#) Appendix section at the end of the lab.

Conclusion

Congratulations! You have successfully:

- Created an Amazon SNS topic
- Created Amazon SQS queues
- Created event notifications in Amazon S3
- Created AWS Lambda functions using pre-existing code
- Triggered a Lambda function from SQS Queues
- Monitored AWS Lambda S3 functions through Amazon CloudWatch Log

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

88. Return to the AWS Management Console.
89. On the navigation bar, choose

awsstudent@<AccountNumber>, and then choose **Sign Out**.

90. Choose End Lab

91. Choose OK

92. (Optional):

- Select the applicable number of stars
- Type a comment
- Choose **Submit**
- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied

You may close the window if you don't want to provide feedback.

Appendix

Optional Task 1 Solution: Create a lifecycle configuration to delete files in the ingest bucket after 30 days

93. On the Services menu, select **S3**.

94. On the **Buckets** page, choose the bucket with name **xxxxx-labbucket-xxxxx**

95. Choose the **Management** tab.

96. In the **Lifecycle rules** section, choose the Create lifecycle rule button.

97. In the **Lifecycle rule configuration** section, configure:

- **Lifecycle rule name:**

- For **Choose a rule scope**: choose **Limit the scope of this rule using one or more filters**.

98. In the **Filter type** section:

- For **Prefix**:

99. In the **Lifecycle rule actions** section, configure:

- check **Expire current versions of objects** and **Permanently delete noncurrent versions of objects**
- Enter the below values in the new boxes that open up.
- *Days after object creation*:
- *Days after objects become noncurrent*:

100. Choose the Create rule button.

Optional Task 2 Solution: Add an SNS e-mail notification to the existing SNS topic

101. On the Services menu, select **Simple Notification Service**.

102. In the left navigation pane, choose **Subscriptions**

103. On the **Subscriptions** page, choose **Create subscription**.

104. On the **Create subscription** page, in the Details section, do the following:

- For **Topic ARN**, choose the ARN of the topic created
- For **Protocol**, choose *Email*
- For **Endpoint**, enter an email address

105. Choose the Create subscription button.

The console creates the subscription and opens the subscription's Details page.

Note: You must confirm the subscription before the email

address can start to receive messages.

106. To confirm a subscription, check your email inbox and choose Confirm subscription in the email from Amazon SNS.

107. Amazon SNS opens your web browser and displays a subscription confirmation with your subscription ID.

For more information about AWS Training and Certification, see <https://aws.amazon.com/training/>.

Your feedback is welcome and appreciated.

If you would like to share any feedback, suggestions, or corrections, please provide the details in our [AWS Training and Certification Contact Form](#).