

Your AWS Services are ready.



Lab 5: Building a Serverless Architecture

© 2023 Amazon Web Services, Inc. or 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 way to all subscribed applications. Using Amazon Simple Notification Service (Amazon SNS) topics and Amazon Simple Queue Service (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 Amazon Simple Storage Service (Amazon 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 an AWS Lambda function using Amazon SQS. This scenario will help you understand how you can architect your application to respond to Amazon S3 bucket events using serverless services such as Amazon SNS, AWS Lambda, and Amazon SQS.

OBJECTIVES

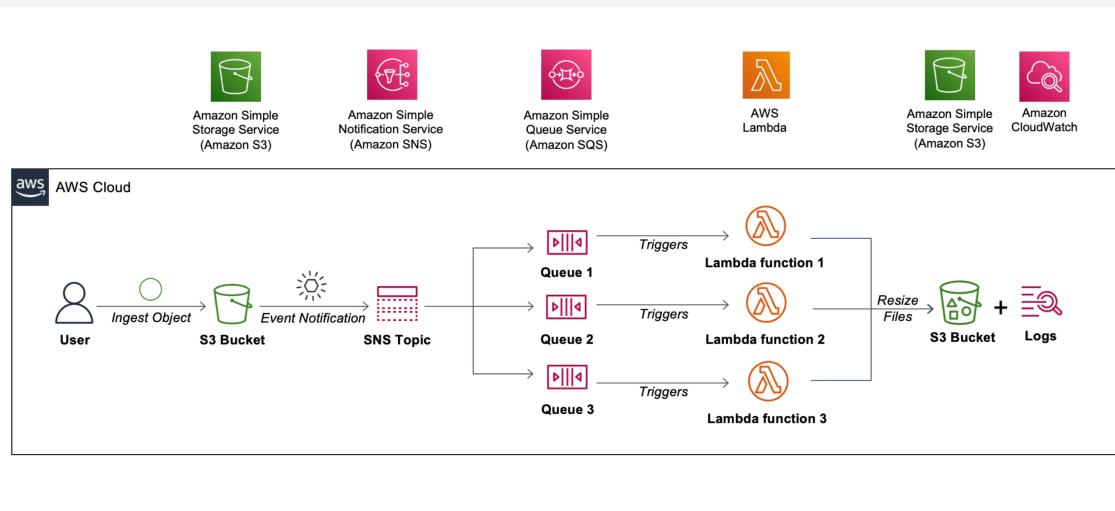
By the end of this lab, you should be able to do the following:

- Understand the value of decoupling resources.
- Understand the potential value of replacing Amazon Elastic Compute Cloud (Amazon 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 preexisting code.
- Invoke an AWS Lambda function from SQS queues.
- Monitor AWS Lambda S3 functions through Amazon CloudWatch Logs.

LAB ENVIRONMENT

You are tasked with evaluating and improving an event-driven architecture. Currently, Customer Care professionals take snapshots of products and upload them into a specific S3 bucket to store the images. The development team runs Python scripts to resize the images after they are uploaded to the ingest S3 bucket. Uploading a file to the ingest bucket invokes an event notification to an Amazon SNS topic. Amazon 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. After reviewing the initial design, you recommend replacing the EC2 instances with Lambda functions. The Lambda functions process the stored images in three formats and store the output in a separate S3 bucket. This proposed design is more cost effective.

The following diagram shows the workflow:



The scenario workflow is as follows:

- You upload an image file to an Amazon S3 bucket.
- Uploading a file to the ingest folder in the bucket invokes an event notification to an Amazon SNS topic.
- Amazon SNS then distributes the notifications to three separate SQS queues.
- The Lambda functions process the images in three formats and store the output in S3 bucket folder.
- You validate the processed images in the S3 bucket folders and the logs in Amazon CloudWatch.

DURATION

This lab requires approximately 45 minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

- 💡 Note: A hint, tip, or important guidance
- ⓘ Learn more: Where to find more information
- ⚠ WARNING: An action that is irreversible and can potentially impact the failure of a command or process (including warnings about configurations that cannot be changed after they are made)

Start lab

1. To launch the lab, at the top of the page, choose Start lab.

ⓘ You must wait for the provisioned AWS services to be ready before you can continue.

2. To open the lab, choose Open Console.

You are automatically signed in to the AWS Management Console in a new web browser tab.

⚠ Do not change the Region unless instructed.

COMMON SIGN-IN ERRORS

Error: You must first sign 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 the click here link.
- Close your Amazon Web Services Sign In web browser tab and return to your initial lab page.
- Choose Open Console again.

Error: Choosing Start Lab has no effect

In some cases, certain pop-up or script blocker web browser extensions might prevent the Start Lab button from working as intended. If you experience an issue starting the lab:

- Add the lab domain name to your pop-up or script blocker's allow list or turn it off.
- Refresh the page and try again.

Task 1: Creating a standard Amazon SNS topic

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

3. At the top of the AWS Management Console, in the search box, search for and choose Simple Notification Service.

4. Expand the navigation menu by choosing the menu icon ☰ in the upper-left corner.

5. From the left navigation menu, choose Topics.

6. Choose Create topic.

The Create topic page is displayed.

7. On the Create topic page, in the Details section, configure the following:

- Type: Choose Standard.
- Name: Enter a unique SNS topic name, such as resize-image-topic-, followed by four random numbers.

8. Choose Create topic.

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

9. Copy the topic ARN and Topic owner values to a notepad. You need these values later in the lab.

✓ Example:

ARN example: arn:aws:sns:us-east-2:123456789012:resize-image-topic Topic owner: 123456789123 (12 digit AWS Account ID)

👉 Congratulations! You have created an Amazon SNS topic.

Task 2: Creating three Amazon SQS queues

In this task, you create three Amazon SQS queues each for a specific purpose and then subscribe the queues to the previously created SNS topic.

TASK 2.1: CREATING AN AMAZON SQS QUEUE FOR THE THUMBNAIL

10. At the top of the AWS Management Console, in the search box, search for and choose Simple Queue Service.

11. On the SQS home page, choose Create queue.

The Create queue page is displayed.

12. On the Create queue page, in the Details section, configure the following:

- Type: Choose Standard (the Standard queue type is set by default).
- Name: Enter thumbnail-queue.

13. The console sets default values for the queue Configuration parameters. Leave the default values.

14. Choose **Create queue**.

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

TASK 2.2: SUBSCRIBING THE SQS QUEUE TO THE SNS TOPIC

15. On the queue's detail page, choose the **SNS subscriptions** tab.

16. Choose **Subscribe to Amazon SNS topic**.

A new **Subscribe to Amazon SNS topic** page opens.

17. From the **Specify an Amazon SNS topic available for this queue** section, choose the **resize-image-topic** SNS topic you created previously 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.

18. Choose **Save**.

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

TASK 2.3: CREATING TWO MORE SQS QUEUES AND SUBSCRIBING THEM TO THE SNS TOPIC

Repeat the steps to create an SQS queue to create two more standard SQS queues, one named *web-queue* and one named *mobile-queue*.

Use the following configurations for the two additional queues:

Create an SQS queue for web-sized images with the following steps:

19. In the left navigation pane, choose **Queues**.

20. Choose **Create queue**.

21. On the **Create queue** page, in the **Details** section, configure the following:

- Type:** Choose **Standard** (the Standard queue type is set by default).
- Name:** Enter .

22. Choose **Create queue**.

Create an SQS Queue for mobile-sized images with the following steps:

23. On the **Create queue** page, in the **Details** section, configure the following:

- Type:** Choose **Standard** (the Standard queue type is set by default).
- Name:** Enter .

24. Choose **Create queue**.

Now repeat the previous steps to subscribe them both to the existing SNS topic named *resize-image-topic-XXXX*.

TASK 2.4: VERIFYING 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.

25. At the top of the AWS Management Console, in the search box, search for and choose .

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

27. On the **Topics** page, choose *resize-image-topic-XXXX*.

28. Choose **Publish message**.

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

29. In the **Message details** section, configure the following:

- Subject - optional:** Enter .

30. In the **Message body** section, configure the following:

- For **Message structure**, select **Identical payload for all delivery protocols**.
- For **Message body sent to the endpoint**, enter or any message of your choice.

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

- For **Type**, choose **String**.
- For **Name**, enter .
- For **Value**, enter .

32. Choose **Publish message**.

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.

33. At the top of the AWS Management Console, in the search box, search for and choose .

34. Choose any queue from the list.

35. Choose **Send and receive messages**.

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

37. Locate the **Message** section. Choose any ID link 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.

38. Choose **Done**.

Congratulations! You have successfully created three Amazon SQS queues and published to a topic that sends notification messages to a queue.

Task 3: Creating an Amazon S3 event notification

In this task, you create an Amazon S3 Event Notification and receive S3 event notifications using the event notification destination as Amazon SNS when certain events happen in the S3 bucket.

TASK 3.1: CONFIGURING THE AMAZON SNS ACCESS POLICY TO ALLOW THE AMAZON S3 BUCKET TO PUBLISH TO A TOPIC

39. At the top of the AWS Management Console, in the search box, search for and choose .

40. From the left navigation menu, choose **Topics**.

41. Choose the `resize-image-topic-XXXX` topic.
42. Choose [Edit](#).
43. Navigate to the `Access policy` section and expand it, if necessary.
44. Delete the existing content of the JSON editor panel.
45. Copy the following code block and paste it into the JSON Editor section.

```
{
  "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": "SNS_TOPIC_ARN",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "SNS_TOPIC_OWNER"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "SNS_TOPIC_ARN",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "SNS_TOPIC_OWNER"
        }
      }
    }
  ]
}
```

46. Replace the `two` occurrences of `SNS_TOPIC_OWNER` with the `Topic owner` (12-digit AWS Account ID) value that you copied earlier in Task 1. Make sure to leave the double quotes.
47. Replace the `two` occurrences of `SNS_TOPIC_ARN` with the SNS topic ARN value copied earlier in Task 1. Make sure to leave the double quotes.
48. Choose [Save changes](#).

TASK 3.2: CREATING A SINGLE S3 EVENT NOTIFICATION ON UPLOADS TO THE INGEST S3 BUCKET

49. At the top of the AWS Management Console, in the search box, search for and choose [S3](#).
50. On the [Buckets](#) page, choose the bucket hyperlink with a name like `xxxx-labbucket-xxxx`.
51. Choose the [Properties](#) tab.
52. Scroll to the [Event notifications](#) section.
53. Choose [Create event notification](#).
54. In the General configuration section, do the following:
 - Event name: Enter `resize-image-event`.
 - Prefix - optional: Enter `ingest/`.

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

Suffix - optional: Enter `.jpg`.

Note: In this lab, you set up a suffix filter so that you receive notifications only when jpg files are uploaded.
55. In the [Event types](#) section, select [All object create events](#).
56. In the [Destination](#) section, configure the following:
 - Destination: Select [SNS topic](#).
 - Specify SNS topic: Select [Choose from your SNS topics](#).
 - SNS topic: Choose the `resize-image-topic-XXXX` SNS topic from the dropdown menu.

Or, if you prefer to specify an ARN, choose [Enter ARN](#) and enter the ARN of the SNS topic copied earlier.
57. Choose [Save changes](#).

 **Congratulations!** You have successfully created an Amazon S3 event notification.

Task 4: Creating and configuring three AWS Lambda functions

In this task, you create three AWS Lambda functions and deploy the respective functionality code to each Lambda function by uploading a zip file archive and configure each Lambda function to add an SQS trigger.

TASK 4.1: CREATING A LAMBDA FUNCTION TO GENERATE A 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 an Amazon S3 bucket folder.

58. At the top of the AWS Management Console, in the search box, search for and choose [Lambda](#).
59. Choose [Create function](#).
60. In the [Create function](#) window, select [Author from scratch](#).

61. In the Basic information section, configure the following:

- Function name: Enter .
- Runtime: Choose Python 3.9.
- 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-XXXX**.

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

⚠ Caution: Make sure to choose **Python 3.9** under **Other supported** runtime. If you choose Python 3.10 or the **Latest supported**, the code in this lab fails as it is configured specifically for Python 3.9.

62. Choose **Create function**.

You should see a message at the top of the page reading, *Successfully created the function CreateThumbnail. You can now change its code and configuration. To invoke your function with a test event, choose "Test"*.

TASK 4.2: CONFIGURING THE LAMBDA FUNCTION TO ADD AN SQS TRIGGER AND UPLOAD THE PYTHON DEPLOYMENT PACKAGE

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

63. Choose **+ Add trigger**, and then configure the following:

- For **Select a trigger**, choose **SQS**.
- For **SQS Queue**, choose **thumbnail-queue**.
- For **Batch Size**, enter .

64. Scroll to the bottom of the page, and then choose **Add**.

At the top of the page you should see a message reading, *The trigger thumbnail-queue was successfully added to function CreateThumbnail. The trigger is in a disabled state.*

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

65. Choose the **Code** tab.

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

- Download and save the following zip file:

67. Open the context menu on the **CreateThumbnail.zip** file name and download the zip file to your computer.

68. Choose **Upload from**, and choose **.zip file**.

69. Choose **Upload**, and upload the downloaded zip file.

70. Choose **Save**.

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

⚠ Caution: Do not copy this code—it is just an example to show what is in the zip file.

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

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(payload)
        bucket_name = sqs_message["Message"]["Records"][0]["s3"]["bucket"][
            "name"
        ]
        print(bucket_name)
        key = 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/Thumbnail-" + key.split("/")[-1]
        )
```

71. Examine the preceding 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

72. In the **Runtime settings** section, choose **Edit**.

- For **Handler**, enter .

73. Choose **Save**.

At the top of the page you should see a message reading, *Successfully updated the function CreateThumbnail.*

⚠ Caution: Make sure you set the **Handler** field to the preceding value, otherwise the Lambda function will not be found.

74. Choose the **Configuration** tab.

75. From the left navigation menu, choose **General configuration**.

76. Choose **Edit**.

- For **Description**, enter [Create a thumbnail-sized image](#).

Leave the other settings at the default settings. 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 processing.

77. Choose **Save**.

You should see a message at the top of the page reading, *Successfully updated the function CreateThumbnail*.

Your Lambda function has now been configured.

TASK 4.3: CREATING AND CONFIGURING TWO MORE LAMBDA FUNCTIONS

Repeat the preceding tasks to create and configure two more Lambda functions (for creating web and mobile images) with the following configurations.

Create a Lambda function to generate a web image.

78. At the top of the Lambda Console, choose the **Functions** breadcrumb.

79. Choose **Create function**.

80. In the **Create function** window, select **Author from scratch**.

81. In the Basic information section, configure the following:

- For **Function name**, enter [CreateWebImage](#).
- For **Runtime**, choose **Python 3.9**.
- Choose the existing **Execution role** with the name like **XXXXX-LabExecutionRole-XXXXX**.

82. Choose **Create function**.

You should see a message at the top of the page reading, *Successfully created the function CreateWebImage. You can now change its code and configuration. To invoke your function with a test event, choose "Test"*.

83. Choose **+ Add trigger**, and then configure the following:

- For **Source**, choose **SQS**.
- For **SQS Queue**, choose **web-queue**.
- For **Batch Size**, enter [1](#).

84. Choose **Add**.

At the top of the page you should see a message reading, *The trigger web-queue was successfully added to function CreateWebImage. The trigger is in a disabled state.*

85. Download and save the following zip file:

- Open the context menu on the [CreateWebImage.zip](#) file name and download the zip file to your computer.

86. Choose the **Code** tab.

87. Choose **Upload from**, and choose **.zip file**.

88. Choose **Upload**, and upload the downloaded zip file.

89. Choose **Save**.

At the top of the page you should see a message reading, *Successfully updated the function CreateWebImage*.

90. In the **Runtime settings** section, choose **Edit**.

91. For **Handler**, enter [CreateWebImage.handler](#).

92. Choose **Save**.

At the top of the page you should see a message reading, *Successfully updated the function CreateWebImage*.

93. Choose the **Configuration** tab.

94. From the left navigation menu, choose **General configuration**.

95. Choose **Edit**.

96. For **Description**, enter [Create a web-sized image](#).

97. Choose **Save**.

You should see a message at the top of the page reading, *Successfully updated the function CreateWebImage*.

Create a Lambda function to generate a mobile image:

98. At the top of the Lambda Console, choose the **Functions** breadcrumb.

99. Choose **Create function**.

100. In the **Create function** window, select **Author from scratch**.

101. In the Basic information section, configure the following:

- **Function name:** Enter [CreateMobileImage](#).
- **Runtime:** Choose **Python 3.9**.
- Choose the existing **Execution role** with the name like **XXXXX-LabExecutionRole-XXXXX**.

102. Choose **Create function**.

At the top of the page you should see a message reading, *Successfully created the function CreateMobileImage. You can now change its code and configuration. To invoke your function with a test event, choose "Test"*.

103. Choose **+ Add trigger**, and then configure the following:

- For **Source**, choose **SQS**.
- For **SQS Queue**, choose **mobile-queue**.
- For **Batch Size**, enter [1](#).

104. Choose **Add**.

At the top of the page you should see a message reading, *The trigger mobile-queue was successfully added to function CreateMobileImage. The trigger is in a disabled state.*

105. Download and save the following zip file:

- Open the context menu on the [CreateMobileImage.zip](#) file name and download the zip file to your computer.

106. Choose the **Code** tab.

107. Choose **Upload from**, and choose **.zip file**.

108. Choose **Upload**, and upload the downloaded zip file.

109. Choose **Save**.

At the top of the page you should see a message reading, *Successfully updated the function CreateMobileImage.*

110. In the **Runtime settings** section, choose **Edit**.

111. For **Handler**, enter `CreateMobileImage.handler`.

112. Choose **Save**.

At the top of the page you should see a message reading, *Successfully updated the function CreateMobileImage.*

113. Choose the **Configuration** tab.

114. From the left navigation menu, choose **General configuration**.

115. Choose **Edit**.

116. For **Description**, enter `Create a mobile-sized image`.

117. Choose **Save**.

You should see a message at the top of the page reading, *Successfully updated the function CreateMobileImage.*

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

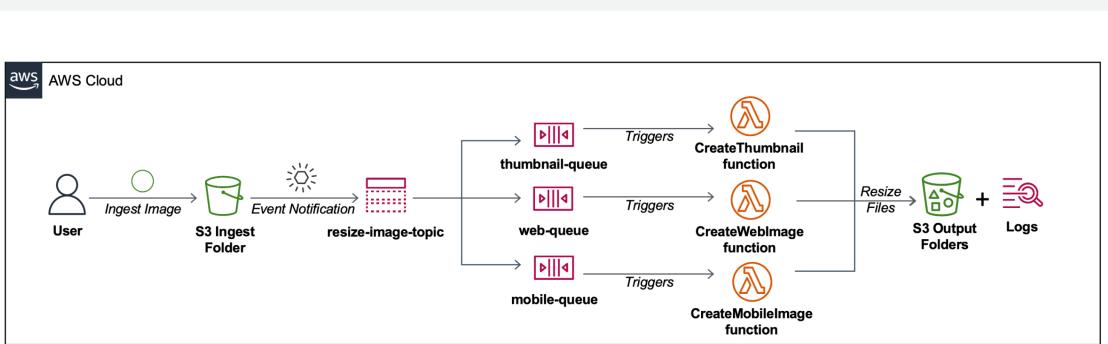
Congratulations! You have successfully created and configured three Lambda functions.

Task 5: Uploading an object to an Amazon S3 bucket

In this task, you upload an object to the previously created S3 bucket using the S3 console.

TASK 5.1: UPLOADING AN IMAGE TO THE S3 BUCKET FOLDER FOR PROCESSING

The following diagram shows the workflow:



Upload a picture to test what you have built.

118. Choose to download *one* image from the following options:

- Open the context menu for the [AWS.jpg](#) link to download the picture to your computer.
- Open the context menu for the [MonaLisa.jpg](#) link to download the picture to your computer.
- Open the context menu for the [HappyFace.jpg](#) link to download the picture to your computer.
- Name your file similar to *InputFile.jpg*.

Caution: Firefox users – Make sure the saved file name is *InputFile.jpg* (not *.jpeg*).

119. At the top of the AWS Management Console, in the search box, search for and choose **S3**.

120. In the **S3 Management Console**, choose the **xxxxx-labbucket-xxxxx** bucket hyperlink.

121. Choose the **ingest/** link.

122. Choose **Upload**.

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

124. Browse to and choose the **XXXXXX.jpg** picture you downloaded.

125. Choose **Upload**.

At the top of the page you should see a message reading, *Upload succeeded.*

Congratulations! You have successfully uploaded JPG images to S3 bucket.

Task 6: Validating the processed file

In this task, you validate the processed file from the logs generated by the function code through Amazon CloudWatch Logs.

TASK 6.1: VIEWING AMAZON CLOUDWATCH LOGS FOR LAMBDA ACTIVITY

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

126. At the top of the AWS Management Console, in the search box, search for and choose **Lambda**.

127. Choose the hyperlink for one of your **Create-** functions.

128. Choose the **Monitor** tab.

The console displays graphs showing the following:

- **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](#).

129. Choose [View CloudWatch Logs](#).

130. Choose the hyperlink for the Log stream that appears.

131. 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, the logs display any logging messages or print statements from the functions. This assists in debugging Lambda functions.

TASK 6.2: VALIDATING THE S3 BUCKET FOR PROCESSED FILES

132. At the top of the AWS Management Console, in the search box, search for and choose [S3](#).

133. Choose the hyperlink for `xxxxx-labbucket-xxxxx` to enter the bucket.

134. You should now see the following three new folders:

- thumbnail
- web
- mobile

135. Navigate through these folders to find the resized images (for example, `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 three different formats.

 **Congratulations!** You have successfully validated the processed image file from the logs generated by the function code through Amazon CloudWatch Logs.

Optional Tasks

Challenge tasks are optional and are provided in case you have extra time remaining in your lab. You can complete the optional tasks or skip to the [end](#) of the lab.

- **(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 email 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 now have successfully:

- Created an Amazon SNS topic
- Created Amazon SQS queues
- Created event notifications in Amazon S3
- Created AWS Lambda functions using preexisting code
- Invoked an AWS Lambda function from SQS queues
- Monitored AWS Lambda S3 functions through Amazon CloudWatch Logs

End lab

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

136. Return to the [AWS Management Console](#).

137. At the upper-right corner of the page, choose `AWSLabUser`, and then choose [Sign out](#).

138. Choose [End lab](#) and then confirm that you want to end your lab.

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](#).

Appendix

OPTIONAL TASK 1 SOLUTION: CREATE A LIFECYCLE CONFIGURATION TO DELETE FILES IN THE INGEST BUCKET AFTER 30 DAYS

139. At the top of the AWS Management Console, in the search box, search for and choose [S3](#).

140. On the [Buckets](#) page, choose the `xxxxx-labbucket-xxxxx` bucket.

141. Choose the [Management](#) tab.

142. In the [Lifecycle rules](#) section, choose [Create lifecycle rule](#)

143. In the **Lifecycle rule configuration** section, configure the following:

- **Lifecycle rule name:** Enter .
- **Choose a rule scope:** Choose **Limit the scope of this rule using one or more filters**.

144. In the **Filter type** section, configure the following:

- **Prefix:** Enter .

145. In the **Lifecycle rule actions** section, configure the following:

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

146. Choose **Create rule**.

OPTIONAL TASK 2 SOLUTION: ADD AN SNS EMAIL NOTIFICATION TO THE EXISTING SNS TOPIC

147. At the top of the AWS Management Console, in the search box, search for and choose .

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

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

150. On the **Create subscription** page, in the Details section, configure the following:

- **Topic ARN:** Choose the ARN of the topic created.
- **Protocol:** Choose **Email**.
- **Endpoint:** Enter an email address.

151. Choose **Create subscription**.

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.

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

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