Conclusion

Congratulations! You now have successfully completed the following:

- Created an Amazon EC2 Auto Scaling group and registered it with an Application Load Balancer spanning across multiple Availability Zones.
- Created a highly available Aurora DB cluster.
- Modified an Aurora DB cluster to be highly available.
- Modified an Amazon VPC configuration to be highly available using redundant NAT gateways.
- Confirmed your database can perform a failover to a read replica instance.

End lab

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

121. Return to the **AWS Management Console**.
122. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
123. 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*.

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

aws training and certification

Lab 5: Build a Serverless Architecture

© 2025 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.

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

- 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 into different formats and stores the output in a separate S3 bucket. This proposed design is more cost effective.

The following diagram shows the workflow:

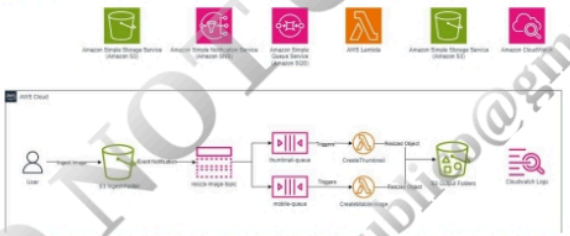


Image description: The preceding diagram depicts serverless architecture in which data is sent to Amazon Simple Storage Service (Amazon S3) which is a serverless object storage. The moment data is uploaded to S3 using S3 Event Notifications the presence of new data is sent to an Amazon SNS topic which broadcasts this information to 2 different Amazon SQS Queues which each calls AWS Lambda to transform the data and the modified information is stored in Amazon S3. All the logs and metrics of all the services included in this architecture is visible using Amazon CloudWatch Metrics and Amazon CloudWatch Logs.

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 separate SQS queues.
- The Lambda functions process the images into formats and stores the output in S3 bucket folder.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

- You validate the processed images in the S3 bucket folders and the logs in Amazon CloudWatch.

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)

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Start lab

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

   ! **Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

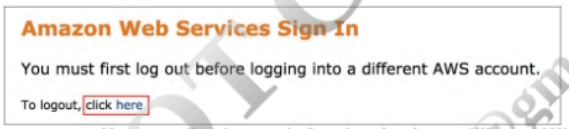
2. To open the lab, choose Open Console ::icon {{name=external}}::.

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

   ⚠ **Warning:** Do not change the **Region** unless instructed.

Common sign-in errors

Error: You must first sign out



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.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Task 1: Create 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.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Task 2: Create two Amazon SQS queues

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

Task 2.1: Create an Amazon SQS queue for the thumbnail image

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.

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 2.2: Create an Amazon SQS queue for the mobile image

19. On the SQS console, expand the navigation menu on the left, and choose **Queues**.

20. Choose Create queue.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The **Create queue** page is displayed.

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 `mobile-queue`.

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

23. Choose Create queue.

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

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

25. Choose Subscribe to Amazon SNS topic.

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

26. 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.

27. Choose Save.

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

Task 2.3: Verifying the AWS SNS subscriptions

To verify the result of the subscriptions, publish to the topic and then view the message that the topic sends to the queue.

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

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

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

31. Choose Publish message.

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

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

- **Subject - optional:**: Enter `Hello world`.

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

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

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

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

- For **Type**, choose **String**.
- For **Name**, enter `Message`.
- For **Value**, enter `Hello World`.

35. 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.

36. At the top of the AWS Management Console, in the search box, search for and choose `Simple Queue Service`.
37. Choose any queue from the list.
38. Choose Send and receive messages .
39. On **Send and receive messages** page, in the **Receive messages** section, choose Poll for messages .
40. Locate the Message section.Choose any ID link in the list to review 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.

41. Choose Done .

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

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Task 3: Create 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 event happen in the S3 bucket.

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

42. At the top of the AWS Management Console, in the search box, search for and choose `Simple Notification Service`.
43. From the left navigation menu, choose **Topics**.
44. Choose the **resize-image-topic-XXXX** topic.
45. Choose Edit .
46. Navigate to the **Access policy - optional** section and expand it, if necessary
47. Delete the existing content of the JSON editor panel.
48. 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"
      ],
      "Resource": "SNS_TOPIC_ARN",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "SNS_TOPIC_OWNER"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
```

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

```
        "Action": "SNS:Publish",
        "Resource": "SNS_TOPIC_ARN",
        "Condition": {
          "StringEquals": {
            "AWS:SourceAccount": "SNS_TOPIC_OWNER"
          }
        }
      }
    ]
}
```

49. 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.

50. 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.

51. Choose Save changes .

Task 3.2: Create a single S3 event notification on uploads to the ingest S3 bucket

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

53. On the **Buckets** page, choose the bucket hyperlink with a name like *xxxxx-labbucket-xxxxx*.

54. Choose the **Properties** tab.

55. Scroll to the **Event notifications** section.

56. Choose Create event notification .

57. 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.

58. In the **Event types** section, select ☑ **All object create events**.

59. In the **Destination** section, configure the following:

- **Destination:** Select **SNS topic**.
- **Specify SNS topic:** Select **Choose from your SNS topics**.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

- **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.

60. Choose Save changes.

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

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Task 4: Create and configure two AWS Lambda functions

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

Task 4.1: Create a Lambda function to generate a thumbnail image

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.

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

62. Choose Create function.

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

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

- **Function name:** Enter `CreateThumbnail`.
- **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-XXXXX*.

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.

65. Choose Create function.

At the top of the page there is a message like, *Successfully created the function CreateThumbnail. You can now change its code and configuration. To invoke your function with a test event, choose "Test".*

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Task 4.2: Configure the CreateThumbnail 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.

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

- For **Select a source**, choose **SQS**.
- For **SQS Queue**, choose **thumbnail-queue**.
- For **Batch size - optional**, enter `1`.

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

At the top of the page there is a message like, *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.

68. Choose the **Code** tab.

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

70. Copy the **CreateThumbnailZIPLocation** value that is listed to the left of these instructions

71. Choose Upload from ▼ , and choose **Amazon S3 location**.

72. Paste the **CreateThumbnailZIPLocation** value you copied from the instructions in the **Amazon S3 link URL** field.

73. Choose Save .

The *CreateThumbnail.zip* file contains the following code:

⚠ **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):
```

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

```
    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 =
json.loads(sqs_message["Message"])["Records"][0]["s3"]["bucket"][
            "name"
        ]
        print(bucket_name)
        key = json.loads(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]
        )
```

74. 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

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

- For **Handler**, enter `CreateThumbnail.handler`.

76. Choose Save.

At the top of the page there is a message like, *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.

77. Choose the **Configuration** tab.

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

79. Choose Edit .

- For **Description** - optional, enter `Create a thumbnail-sized image.`

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

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.

80. Choose Save.

A message is displayed at the top of the page with text like, *Successfully updated the function CreateThumbnail.*

The *CreateThumbnail* Lambda function has now been configured.

Task 4.3: Create a Lambda function to generate a mobile image

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.

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

82. Choose Create function.

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

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

- **Function name:** Enter `CreateMobileImage`.
- **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-XXXXX*.

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.

85. Choose Create function.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

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

Task 4.4: Configure the CreateMobileImage 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.

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

- For **Select a source**, choose **SQS**.
- For **SQS Queue**, choose **mobile-queue**.
- For **Batch size - optional**, enter `1`.

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

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

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

88. Choose the **Code** tab.

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

90. Copy the **CreateMobileImageZIPLocation** value that is listed to the left of these instructions

91. Choose Upload from ▼ , and choose **Amazon S3 location**.

92. Paste the **CreateMobileImageZIPLocation** value you copied from the instructions in the **Amazon S3 link URL** field.

93. Choose Save .

The *CreateMobileImage.zip* file contains the following code:

⚠ **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((640, 320))
        image.save(resized_path)


def handler(event, context):
    for record in event["Records"]:
        payload = record["body"]
        sqs_message = json.loads(payload)
        bucket_name =
json.loads(sqs_message["Message"])["Records"][0]["s3"]["bucket"][
            "name"
        ]
        print(bucket_name)
        key = json.loads(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, "mobile/MobileImage-" + key.split("/")[-1]
        )
```

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

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

95. Choose Save.

At the top of the page there is a message like, *Successfully updated the function CreateMobileImage.*

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

96. Choose the **Configuration** tab.

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

98. Choose Edit .

- For **Description** - optional, enter `Create a mobile friendly image.`

Leave the other settings at the default settings. Here is a brief explanation of these settings:

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

- *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.

99. Choose Save.

A message is displayed at the top of the page with text like, *Successfully updated the function CreateMobileImage.*

The *CreateMobileImage* Lambda function has now been configured.

Congratulations! You have successfully created 2 AWS Lambda functions for the serverless architecture and set the appropriate SQS queue as trigger for their respective functions.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

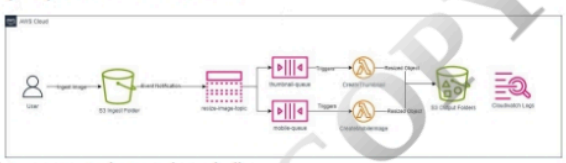
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Task 5: Upload 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: Upload an image to the S3 bucket folder for processing

The following diagram shows the workflow:



Upload a picture to test what you have built.

100. 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*).

101. At the top of the AWS Management Console, in the search box, search for and choose `S3`.
102. In the **S3 Management Console**, choose the **xxxxx-labbucket-xxxxx** bucket hyperlink.
103. Choose the **ingest/** link.
104. Choose Upload.
105. In the **Upload** window, choose Add files.
106. Browse to and choose the **XXXXX.jpg** picture you downloaded.
107. Choose Upload.

At the top of the page, there is a message like, *Upload succeeded.*

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

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Task 6: Validate 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: Review Amazon CloudWatch Logs for Lambda activity

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

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

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

110. 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*.

111. Choose View CloudWatch logs .

112. Choose the hyperlink for the newest Log stream that appears.

113. Expand ▶ each message to view the log message details.

The REPORT line provides the following details:

- **RequestId:** The unique request ID for the invocation

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

- **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.

Note: Reviewing the logs you may notice that the Lambda function has been executed multiple times. This is because the Lambda function is receiving the test message posted to the SNS topic in task 2. Another one of logs was generated when the event notifications for your S3 bucket was created. The third log was generated when an object was uploaded the S3 bucket, and triggered the functions.

Task 6.2: Validate the S3 bucket for processed files

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

115. Choose the hyperlink for **xxxxx-labbucket-xxxxx** to enter the bucket.

116. Navigate through these folders to find the resized images (for example, *Thumbnail-AWS.jpg, MobileImage-MonaLisa.jpg*).

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

Congratulations! You have successfully validated the processed image file from the logs generated by the function code through browsing Amazon S3 and 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.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

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.

117. Return to the **AWS Management Console**.

118. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.

119. 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

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

121. On the **Buckets** page, choose the **xxxxx-labbucket-xxxxx** bucket.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

122. Choose the **Management** tab.

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

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

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

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

- **Prefix:** Enter `ingest/`.

126. 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 `30`.
  - **Days after objects become noncurrent:** Enter `1`.

127. Choose Create rule

Optional Task 2 Solution: Add an SNS email notification to the existing SNS topic

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

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

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

131. 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.

132. Choose Create subscription.

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

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Note: You must confirm the subscription before the email address can start to receive messages.

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

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

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

aws training and certification

Lab 6: Configure an Amazon CloudFront Distribution with an Amazon S3 Origin

© 2025 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

Amazon Web Services (AWS) solutions architects must frequently design and build secure, high-performing, resilient, efficient architectures for applications and workloads to deliver content. Amazon CloudFront is a web service that provides a cost-effective way to distribute content with low latency and high data transfer speeds. You can use CloudFront to accelerate static website content delivery, serve video on demand or live streaming video, and even run serverless code at the edge location. In this lab, you configure a CloudFront distribution in front of an Amazon Simple Storage Service (Amazon S3) bucket and secure it using origin access control (OAC) provided by CloudFront.

Objectives

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

- Create an S3 bucket with default security settings.
- Configure an S3 bucket for public access.
- Add an S3 bucket as a new origin to an existing CloudFront distribution.
- Secure an S3 bucket to permit access only through the CloudFront distribution.
- Configure OAC to lock down security to an S3 bucket.
- Configure Amazon S3 resource policies for public or OAC access.

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.