

# 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.

## 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:

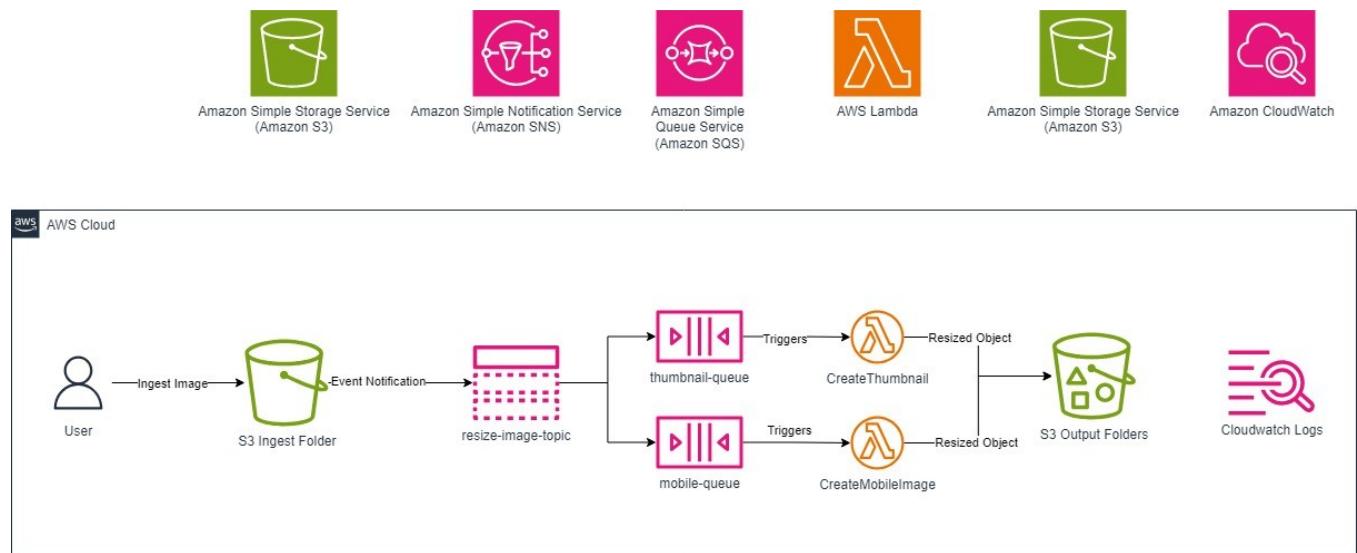
- Understand the value of decoupling resources.
- Understand the potential value of replacing Amazon Elastic Compute Cloud (Amazon EC2) instances with Lambda functions.
- Create an Amazon SNS topic.
- Create Amazon SQS queues.
- Create event notifications in Amazon S3.
- Create AWS Lambda functions using 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

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

## 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.

- At the top of the AWS Management Console, in the search box, search for and choose [Simple Notification Service](#).

- Expand the navigation menu by choosing the menu icon in the upper-left corner.
- From the left navigation menu, choose **Topics**.

The screenshot shows the AWS SNS Topics page. The left sidebar has 'Amazon SNS' selected under 'Topics'. The main area is titled 'Topics (0)' with a search bar. Below it is a table header with columns 'Name' and 'Type'. A message says 'No topics' and 'To get started, create a topic.' A blue 'Create topic' button is highlighted at the bottom right.

- Choose **Create topic**.

The **Create topic** page is displayed.

The screenshot shows the 'Create topic' page. Under 'Details', the 'Type' dropdown is set to 'Info'. The 'FIFO (first-in, first-out)' section is selected, showing:
 

- Strictly-preserved message ordering
- Exactly-once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

 The 'Standard' section is also shown:
 

- Best-effort message ordering
- At-least once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

 The 'Name' field is filled with 'MyTopic.fifo'. Other sections like 'High throughput' and 'Topic scope' are partially visible below.

- On the **Create topic** page, in the **Details** section, configure the following:
  - o **Type:** Choose **Standard**.
  - o **Name:** Enter a unique SNS topic name, such as `resize-image-topic-`, followed by four random numbers. (`resize-image-topic-1992`)

**Create topic**

**Details**

**Type** | [Info](#)  
Topic type cannot be modified after topic is created

FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- Subscription protocols: SQS

Standard

- Best-effort message ordering
- At-least once message delivery
- Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints

**Name**  
`resize-image-topic-1992`  
Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (\_).

**Display name - optional** | [Info](#)  
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message.  
`My Topic`  
Maximum 100 characters.

- Choose **Create topic**.

**Access policy - optional** | [Info](#)  
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

**Data protection policy - optional** | [Info](#)  
This policy defines which sensitive data to monitor and to prevent from being exchanged via your topic.

**Delivery policy (HTTP/S) - optional** | [Info](#)  
The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section.

**Delivery status logging - optional** | [Info](#)  
These settings configure the logging of message delivery status to CloudWatch Logs.

**Tags - optional**  
A tag is a metadata label that you can assign to an Amazon SNS topic. Each tag consists of a key and an optional value. You can use tags to search and filter your topics and track your costs. [Learn more](#)

**Active tracing - optional** | [Info](#)  
Use AWS X-Ray active tracing for this topic to view its traces and service map in Amazon CloudWatch. Additional costs apply.

[Cancel](#) [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.

A screenshot of the AWS Management Console. The top navigation bar shows 'aws' and various service icons like EC2, VPC, S3, and IAM. The main title is 'Amazon SNS &gt; Topics &gt; resize-image-topic-1992'. A green success message box says 'Topic resize-image-topic-1992 created successfully. You can create subscriptions and send messages to them from this topic.' Below it, the topic details are listed: Name: resize-image-topic-1992, ARN: arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992, and Type: Standard. On the right, there are 'Edit', 'Delete', and 'Publish message' buttons. Below the details, there are tabs for 'Subscriptions', 'Access policy', 'Data protection policy', 'Delivery policy (HTTP/S)', 'Delivery status logging', 'Encryption', 'Tags', and 'Integrations'. The 'Subscriptions' tab is selected and shows a table with 0 rows. The table has columns for 'ID', 'Endpoint', 'Status', and 'Protocol'. A button 'Create subscription' is at the bottom of the table.

- 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-west-2:903564999125:resize-image-topic-1992  
Topic owner: 903564999125 (12 digit AWS Account ID)*

You have created an Amazon SNS topic.

## 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

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

The screenshot shows the AWS SQS home page. At the top, there's a navigation bar with links for EC2, VPC, S3, and IAM. Below the navigation, the title "Amazon SQS" and subtitle "A message queuing service" are displayed. A brief description follows: "Amazon SQS provides queues for high-throughput, system-to-system messaging. You can use queues to decouple heavyweight processes and to buffer and batch work. Amazon SQS stores messages until microservices and serverless applications process them." To the right, there are three callout boxes: "Get started" (with a "Create queue" button), "Pricing (US)" (with a "Cost calculator" link), and "Documentation" (with a "Developer guide" link). Below these, a large diagram titled "How it works" illustrates the flow: Producers send messages to an "Amazon SQS" queue, which then sends messages to Consumers (represented by a grid icon).

- On the SQS home page, choose **Create queue**.

The **Create queue** page is displayed.

The screenshot shows the "Create queue" configuration page. At the top, the URL is "Amazon SQS > Queues > Create queue". The main section is titled "Create queue" and contains two tabs: "Details" and "Configuration". In the "Details" tab, under "Type", the "Standard" option is selected (info: "At-least-once delivery, message ordering isn't preserved"). The "Name" field is set to "MyQueue". A note says: "You can't change the queue type after you create a queue." In the "Configuration" tab, settings include: "Visibility timeout": 30 seconds (info: "Should be between 0 seconds and 12 hours."); "Message retention period": 4 days (info: "Should be between 1 minute and 14 days."); "Delivery delay": 0 seconds (info: "Should be between 0 and 120 seconds."); and "Maximum message size": 256 KB (info: "Should be between 1 KB and 16 MB").

- 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**.

aws | Search [Alt+S] | United States (Oregon) ▾

EC2 VPC S3 IAM

Amazon SQS > Queues > Create queue

### Create queue

#### Details

Type  
Choose the queue type for your application or cloud infrastructure.

Standard [Info](#)  
At-least-once delivery, message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

FIFO [Info](#)  
First-in-first-out delivery, message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing

! You can't change the queue type after you create a queue.

Name  
  
A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (\_).

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

aws | Search [Alt+S] | United States (Oregon) ▾

Amazon SQS > Queues > Create queue

#### Configuration [Info](#)

Set the maximum message size, visibility to other consumers, and message retention.

Visibility timeout [Info](#)  
 Seconds  
Should be between 0 seconds and 12 hours.

Message retention period [Info](#)  
 Days  
Should be between 1 minute and 14 days.

Delivery delay [Info](#)  
 Seconds  
Should be between 0 seconds and 15 minutes.

Maximum message size [Info](#)  
 KB  
Should be between 1 KB and 256 KB.

Receive message wait time [Info](#)  
 Seconds  
Should be between 0 and 20 seconds.

aws | Search [Alt+S] | United States (Oregon) ▾

Amazon SQS > Queues > Create queue

#### Encryption [Info](#)

Amazon SQS provides in-transit encryption by default. To add at-rest encryption to your queue, enable server-side encryption.

Server-side encryption  
 Disabled  
 Enabled

Encryption key type  
 Amazon SQS key (SSE-SQS)  
An encryption key that Amazon SQS creates, manages, and uses for you.  
 AWS Key Management Service key (SSE-KMS)  
An encryption key protected by AWS Key Management Service (AWS KMS).

aws | Search [Alt+S] | United States (Oregon) ▾

Amazon SQS > Queues > Create queue

#### Access policy [Info](#)

Define who can access your queue.

Choose method

Basic  
Use simple criteria to define a basic access policy.

Advanced  
Use a JSON object to define an advanced access policy.

Define who can send messages to the queue

Only the queue owner  
Only the owner of the queue can send messages to the queue.

Only the specified AWS accounts, IAM users and roles  
Only the specified AWS account IDs, IAM users and roles can send messages to the queue.

Define who can receive messages from the queue

Only the queue owner  
Only the owner of the queue can receive messages from the queue.

Only the specified AWS accounts, IAM users and roles  
Only the specified AWS account IDs, IAM users and roles can receive messages from the queue.

JSON (read-only)

```
{
  "Version": "2012-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "903564999125"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:us-west-2:903564999125:thumbnail-queue"
    }
  ]
}
```

The screenshot shows the 'Create queue' wizard in the AWS SQS console. The first step, 'Redrive allow policy', is set to 'Disabled'. The second step, 'Dead-letter queue', is also set to 'Disabled'. The third step, 'Tags', shows a single tag named 'Thumbnail-queue' with the value 'Thumbnail-queue'. At the bottom right are 'Cancel' and 'Create queue' buttons.

- Choose **Create queue**.

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

The screenshot shows the 'Details' tab for the 'thumbnail-queue'. It includes fields for Name (thumbnail-queue), Type (Standard), ARN (arn:aws:sqs:us-west-2:903564999125:thumbnail-queue), URL (https://sqs.us-west-2.amazonaws.com/903564999125/thumbnail-queue), and Dead-letter queue (-). Below the tabs are sections for SNS subscriptions, Lambda triggers, EventBridge Pipes, Dead-letter queue, Monitoring, Tagging, Queue policies, Encryption, and Dead-letter queue redrive tasks. The SNS subscriptions section shows 0 subscriptions and a 'Subscribe to Amazon SNS topic' button.

- On the queue's detail page, choose the **SNS subscriptions** tab.
- Choose **Subscribe to Amazon SNS topic**.

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

The screenshot shows the 'Subscribe to Amazon SNS topic' dialog box. At the top, there's a header with the AWS logo, search bar, and navigation links for EC2, VPC, S3, and IAM. Below the header, the path is shown as 'Amazon SQS > Queues > thumbnail-queue > Subscribe to Amazon SNS topic'. The main content area has a title 'Amazon SNS topic' and a sub-instruction: 'To allow your queue to receive messages from an Amazon SNS topic, subscribe it to an Amazon SNS topic.' A section titled 'Specify an Amazon SNS topic available for this queue.' contains a dropdown menu with the placeholder 'Choose a topic'. At the bottom right are 'Cancel' and 'Save' buttons.

- 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**.

The screenshot shows a dropdown menu for selecting an Amazon SNS topic. The menu has a placeholder 'Choose a topic' at the top. Below it is a search bar with the placeholder 'Search Amazon SNS topics.' Underneath the search bar are two options: 'Enter Amazon SNS topic ARN' and 'Use existing resource'. The 'Use existing resource' option is expanded, showing a list containing the ARN 'arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992'. At the bottom of the list is the text 'End of results'. On the right side of the menu, there are 'Cancel' and 'Save' buttons.

**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-1992*.

The screenshot shows the 'Details' tab of the 'thumbnail-queue' queue configuration. The queue name is 'thumbnail-queue'. Key details shown include:

- Name:** thumbnail-queue
- Type:** Standard
- ARN:** arn:aws:sqs:us-west-2:903564999125:thumbnail-queue
- URL:** https://sqs.us-west-2.amazonaws.com/903564999125/thumbnail-queue
- Dead-letter queue:** -

At the bottom left, there's a 'More' link. On the right, there are buttons for 'Edit', 'Delete', 'Purge', 'Send and receive messages', and 'Start DLQ redrive'.

## Task 2.2: Create an Amazon SQS queue for the mobile image

- On the SQS console, expand the navigation menu on the left, and choose **Queues**.
- Choose **Create queue**.

The **Create queue** page is displayed.

The screenshot shows the 'Create queue' page in the AWS SQS console. In the 'Details' section, the 'Type' dropdown is set to 'Standard Info'. The 'Name' field contains 'MyQueue'. Below the name, a note states: 'A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (\_).'. In the 'Configuration' section, 'Visibility timeout' is set to 30 seconds, 'Message retention period' is 4 days, 'Delivery delay' is 0 seconds, and 'Maximum message size' is 256 KB.

- 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**.

The screenshot shows the 'Create queue' page in the AWS SQS console. In the 'Details' section, the 'Type' dropdown is set to 'Standard Info'. The 'Name' field now contains 'mobile-queue'. A note below the name states: 'A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (\_).'. The configuration parameters remain the same as in the previous screenshot.

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

**Configuration** [Info](#)

Set the maximum message size, visibility to other consumers, and message retention.

<b>Visibility timeout</b> <a href="#">Info</a>	<b>Message retention period</b> <a href="#">Info</a>
<input type="text" value="30"/> Seconds	<input type="text" value="4"/> Days
Should be between 0 seconds and 12 hours.	
<b>Delivery delay</b> <a href="#">Info</a>	<b>Maximum message size</b> <a href="#">Info</a>
<input type="text" value="0"/> Seconds	<input type="text" value="256"/> KB
Should be between 0 seconds and 15 minutes.	
<b>Receive message wait time</b> <a href="#">Info</a>	
<input type="text" value="0"/> Seconds	
Should be between 0 and 20 seconds.	

**Encryption** [Info](#)

Amazon SQS provides in-transit encryption by default. To add at-rest encryption to your queue, enable server-side encryption.

**Server-side encryption**

Disabled  
 Enabled

**Encryption key type**

Amazon SQS key (SSE-SQS)  
An encryption key that Amazon SQS creates, manages, and uses for you.

AWS Key Management Service key (SSE-KMS)  
An encryption key protected by AWS Key Management Service (AWS KMS).

**Access policy** [Info](#)

Define who can access your queue.

**Choose method**

Basic  
Use simple criteria to define a basic access policy.

Advanced  
Use a JSON object to define an advanced access policy.

**Define who can send messages to the queue**

Only the queue owner  
Only the owner of the queue can send messages to the queue.

Only the specified AWS accounts, IAM users and roles  
Only the specified AWS account IDs, IAM users and roles can send messages to the queue.

**Define who can receive messages from the queue**

Only the queue owner  
Only the owner of the queue can receive messages from the queue.

Only the specified AWS accounts, IAM users and roles  
Only the specified AWS account IDs, IAM users and roles can receive messages from the queue.

**JSON (read-only)**

```
{
  "Version": "2012-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "903564999125"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:us-west-2:903564999125:mobile-queue"
    }
  ]
}
```

**Redrive allow policy** - *Optional* [Info](#)

Identify which source queues can use this queue as the dead-letter queue.

**Select which source queues can use this queue as the dead-letter queue.**

Disabled  
 Enabled

**Dead-letter queue** - *Optional* [Info](#)

Send undeliverable messages to a dead-letter queue.

**Set this queue to receive undeliverable messages.**

Disabled  
 Enabled

**Tags** - *Optional* [Info](#)

A tag is a label assigned to an AWS resource. Use tags to search and filter your resources or track your AWS costs.

<b>Key</b>	<b>Value - optional</b>
<input type="text" value="Enter key"/>	<input type="text" value="Enter value"/>
<a href="#">Add new tag</a>	
You can add 49 more tags.	

[Cancel](#) [Create queue](#)

- Choose **Create queue**.

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

The screenshot shows the 'mobile-queue' queue details page. At the top, a green banner says 'Queue mobile-queue created successfully. You can now send and receive messages.' Below the banner, the queue name 'mobile-queue' is displayed. A row of buttons includes 'Edit', 'Delete', 'Purge', 'Send and receive messages' (highlighted in blue), and 'Start DLQ redrive'. The main area shows 'Details' with fields: Name (mobile-queue), Type (Standard), ARN (arn:aws:sqs:us-west-2:903564999125:mobile-queue), Encryption (Amazon SQS key (SSE-SQS)), URL (https://sqs.us-west-2.amazonaws.com/903564999125/mobile-queue), and Dead-letter queue (-). A 'More' link is shown. Below this is a navigation bar with tabs: SNS subscriptions (selected), Lambda triggers, EventBridge Pipes, Dead-letter queue, Monitoring, Tagging, Queue policies, Encryption, and Dead-letter queue redrive tasks. The 'Subscription region' dropdown is set to 'us-west-2'. The 'SNS subscriptions (0)' section has a search bar and a 'Subscribe to Amazon SNS topic' button. A note says 'No Amazon SNS subscriptions' and 'To allow your queue to receive messages, subscribe it to an Amazon SNS topic.'

- On the queue's detail page, choose the **SNS subscriptions** tab.
- Choose **Subscribe to Amazon SNS topic**.

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

The screenshot shows the 'Subscribe to Amazon SNS topic' dialog box. It includes a header with the AWS logo, search bar, and navigation links (EC2, VPC, S3, IAM). The path 'Amazon SQS > Queues > mobile-queue > Subscribe to Amazon SNS topic' is visible. The main content area is titled 'Subscribe to Amazon SNS topic' with a 'Info' link. It contains a section for 'Amazon SNS topic' with a note: 'To allow your queue to receive messages from an Amazon SNS topic, subscribe it to an Amazon SNS topic.' Below this is a 'Specify an Amazon SNS topic available for this queue.' section with a dropdown menu labeled 'Choose a topic'. At the bottom are 'Cancel' and 'Save' buttons.

- 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**.

**Amazon SNS topic**

To allow your queue to receive messages from an Amazon SNS topic, subscribe it to an Amazon SNS topic.

Specify an Amazon SNS topic available for this queue.

Choose a topic ▾

Search Amazon SNS topics.

Enter Amazon SNS topic ARN

Use existing resource

arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992

End of results

arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992

Cancel Save

**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.

- Choose **Save**.

Your SQS queue is now subscribed to the SNS topic named *resize-image-topic-1992*.

### 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.

- At the top of the AWS Management Console, in the search box, search for and choose **Simple Notification Service**.
- In the left navigation pane, choose **Topics**.
- On the **Topics** page, choose **resize-image-topic-1992**.
- Choose **Publish message**.

Topics (1)

Name Type ARN

resize-image-topic-1992 Standard arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992

Edit Delete Publish message Create topic

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

**Publish message to topic**

**Message details**

**Topic ARN**  
arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992

**Subject - optional**  
Enter message subject  
Maximum 100 printable ASCII characters

**Time to Live (TTL) - optional** Info  
This setting applies only to mobile application endpoints. The number of seconds that the push notification service has to deliver the message to the endpoint.

**Message body**

**Message structure**

Identical payload for all delivery protocols.  
The same payload is sent to endpoints subscribed to the topic, regardless of their delivery protocol.

Custom payload for each delivery protocol.  
Different payloads are sent to endpoints subscribed to the topic, based on their delivery protocol.

**Message body to send to the endpoint**  
Hello world

- In the **Message details** section, configure the following:

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

## Publish message to topic

**Message details**

**Topic ARN**  
arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992

**Subject - optional**  
Hello world  
Maximum 100 printable ASCII characters

**Time to Live (TTL) - optional** Info  
This setting applies only to mobile application endpoints. The number of seconds that the push notification service has to deliver the message to the endpoint.

- In the **Message body** section, configure the following:

- 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.

**Message body**

**Message structure**

Identical payload for all delivery protocols.  
The same payload is sent to endpoints subscribed to the topic, regardless of their delivery protocol.

Custom payload for each delivery protocol.  
Different payloads are sent to endpoints subscribed to the topic, based on their delivery protocol.

**Message body to send to the endpoint**  
1 Testing Hello world

- In the **Message attributes** section, configure the following:
    - For **Type**, choose **String**.
    - For **Name**, enter **Message**.
    - For **Value**, enter **Hello World**.

**Message attributes** [Info](#)

Message attributes let you provide structured metadata items (such as timestamps, geospatial data, signatures, and identifiers) for the message.

Type	Name	Value	
String	Message	Hello World	<a href="#">Remove</a>

[Add another attribute](#)

[Cancel](#) [Publish message](#)

- Choose **Publish message**.

The message is published to the topic, and the console opens the topic's detail page.

The screenshot shows the AWS SNS console with the following details:

- Left sidebar:** Shows navigation links for EC2, VPC, S3, IAM, Amazon SNS, Topics, Subscriptions, and Mobile.
- Header:** Includes the AWS logo, search bar, and system status (United States (Oregon)).
- Topic Overview:** The topic is named "resize-image-topic-1992".
  - Details:** Name: resize-image-topic-1992, Display name: -, ARN: arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992, Topic owner: 903564999125, Type: Standard.
  - Subscriptions:** There are 2 subscriptions listed.
- Message History:** A green box indicates a successful message publication.
  - Message ID: 3989f15a-bd62-5342-a35b-912334af2e4f
  - Request ID: 9efc2e08-2a60-59b1-8012-4e71a3ad54bc
- Actions:** Buttons for Publish another message, Edit, Delete, and Publish message.

To investigate the published message, navigate to Amazon SQS.

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

Queues (2)									
Queues									
Actions									
Create queue									
<input type="text" value="Search queues by prefix"/>									
Name	Type	Created	Messages available	Messages in flight	Encryption	Content-based deduplication			
<a href="#">mobile-queue</a>	Standard	2025-04-16T15:31:05:30	1	0	Amazon SQS key (SSE-SQS)	-			
<a href="#">thumbnail-queue</a>	Standard	2025-04-16T15:15+05:30	1	0	Amazon SQS key (SSE-SQS)	-			

- Choose any queue from the list.

The screenshot shows the AWS SQS console with the URL [https://console.aws.amazon.com/sqs/v2/home?region=us-west-2#queue:config/thumbnail-queue](#). The queue name is "thumbnail-queue". The "Send and receive messages" button is highlighted with a black border.

- Choose **Send and receive messages**.

The screenshot shows the "Send and receive messages" page for the "thumbnail-queue". The "Poll for messages" button is highlighted with a black border.

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

The screenshot shows the "Receive messages" page for the "thumbnail-queue". A message is listed in the "Messages" section. The "View details" link in the message row is highlighted with a black border.

- Locate the Message section. Choose any ID link in the list to review the Details, Body, and Attributes of the message.

Message: d99750c1-bd35-4a66-88ab-780049377d7d

**Body** Attributes Details

```
{
  "Type": "Notification",
  "MessageId": "3989f15a-bd62-5342-a358-912334af2e4f",
  "TopicArn": "arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992",
  "Subject": "Hello world",
  "Message": "Testing Hello world",
  "Timestamp": "2025-04-16T10:33:49.392Z",
  "SignatureVersion": "1",
  "Signature": "JWeDJVW8TR7fJ4MYidnswpBgAMqDhRXtt50yD33885o9R/KXdbR8UAVQFYj7YYtwgMrpjP4odX+rBTG0mO9osBlqrWTaLZ7MK1+wUBoT9jlTsQRBs8R2dsJA/Ygfnx8PA0UNjeUWgOWaRQWrUfpnXpOPP3hUUteMLf0336Xk+Kton/N3JkBugLtk/d7avFfzIz+W5Wn3HhCgPb6dVxdkU2pCdV2ada4lZTBY+muDKlQSqQt0kSiXzsbrClAlJaKMROn4UC4mRz8gQO6uzTr715JAXCGHTh5wf6yFQSDJom6uJoGuz3hjl4tvhTKJZxVcSWZm/YOQ9LD0gikjA==",
  "SigningCertURL": "https://sns.us-west-2.amazonaws.com/SimpleNotificationService-9c6465fa7f48f5cad23014631ec1136.pem",
  "UnsubscribeURL": "https://sns.us-west-2.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992:2de22532-1132-406e-9c36-2b2a95095f67",
  "MessageAttributes": {
    "Message": {"Type": "String", "Value": "Hello World"}
  }
}
```

Done

Message: d99750c1-bd35-4a66-88ab-780049377d7d

**Body** **Attributes** Details

**Attributes (0)**

Name	Type	Value
No message attributes.		

Done

Message: d99750c1-bd35-4a66-88ab-780049377d7d

**Body** Attributes Details

ID	Size	MD5 of message body	Sender account ID
d99750c1-bd35-4a66-88ab-780049377d7d	1.04 KB	c99c874ae6b6bf8bf5c2c 749ba653e59	AIDAIYLAVTDLUXBIEIX46
Sent	First received	Receive count	Message attributes count
2025-04-16T16:03+05:30	2025-04-16T16:21+05:30	2	-
Message attributes size	MD5 of message attributes		
-	-		

Done

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

- Choose **Done**.

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

## 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 events 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

- At the top of the AWS Management Console, in the search box, search for and choose **Simple Notification Service**.
- From the left navigation menu, choose **Topics**.

Name	Type	ARN
resize-image-topic-1992	Standard	arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992

- Choose the **resize-image-topic-1992** topic.

ID	Endpoint	Status	Protocol
2de22532-1132-406e-9c36-2b2a95095f67	arn:aws:sqs:us-west-2:903564999125:th...	Confirmed	SQS
eb00d873-72e1-42c9-8ae0-f9ca98b38cb0	arn:aws:sqs:us-west-2:903564999125:mo...	Confirmed	SQS

- Choose **Edit**.

The screenshot shows the AWS SNS 'Edit topic' interface. At the top, there's a navigation bar with links for EC2, VPC, S3, and IAM. Below that, the breadcrumb trail shows 'Amazon SNS > Topics > resize-image-topic-1992 > Edit topic'. The main content area has a title 'Edit resize-image-topic-1992' and a 'Details' section. Under 'Name', it says 'resize-image-topic-1992'. Under 'Type', it says 'Standard'. There are three expandable sections: 'Encryption - optional' (with a note about in-transit encryption), 'Access policy - optional' (with a note about who can access the topic), and 'Data protection policy - optional'.

- Navigate to the **Access policy - optional** section and expand it, if necessary

**▼ Access policy - optional** Info  
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic.

JSON editor

```

1 {
2   "Version": "2008-10-17",
3   "Id": "__default_policy_ID",
4   "Statement": [
5     {
6       "Sid": "__default_statement_ID",
7       "Effect": "Allow",
8       "Principal": {
9         "AWS": "*"
10      },
11      "Action": [
12        "SNS:Publish",
13        "SNS:RemovePermission",
14        "SNS:SetTopicAttributes",
15        "SNS>DeleteTopic",

```

- Delete the existing content of the JSON editor panel.
- 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"
        },
        "Action": "SNS:Publish",
        "Resource": "SNS_TOPIC_ARN",
        "Condition": {
            "StringEquals": {
                "AWS:SourceAccount": "SNS_TOPIC_OWNER"
            }
        }
    }
]
}

```

- 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.
- 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.

#### Old Access policy:

```

{
    "Version": "2008-10-17",
    "Id": "__default_policy_ID",
    "Statement": [
        {
            "Sid": "__default_statement_ID",
            "Effect": "Allow",
            "Principal": {
                "AWS": "*"
            },
            "Action": [
                "SNS:Publish",
                "SNS:RemovePermission",
                "SNS:SetTopicAttributes",
                "SNS:DeleteTopic",
                "SNS>ListSubscriptionsByTopic",
                "SNS:GetTopicAttributes",
                "SNS>AddPermission",
                "SNS:Subscribe"
            ]
        }
    ]
}

```

```

    ],
    "Resource": "arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992",
    "Condition": {
        "StringEquals": {
            "AWS:SourceOwner": "903564999125"
        }
    }
}
]
}

```

### New Access policy:

```

{
    "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": "arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992",
            "Condition": {
                "StringEquals": {
                    "AWS:SourceAccount": "903564999125"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "SNS:Publish",
            "Resource": "arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992",

```

```

    "Condition": {
        "StringEquals": {
            "AWS:SourceAccount": "903564999125"
        }
    }
}

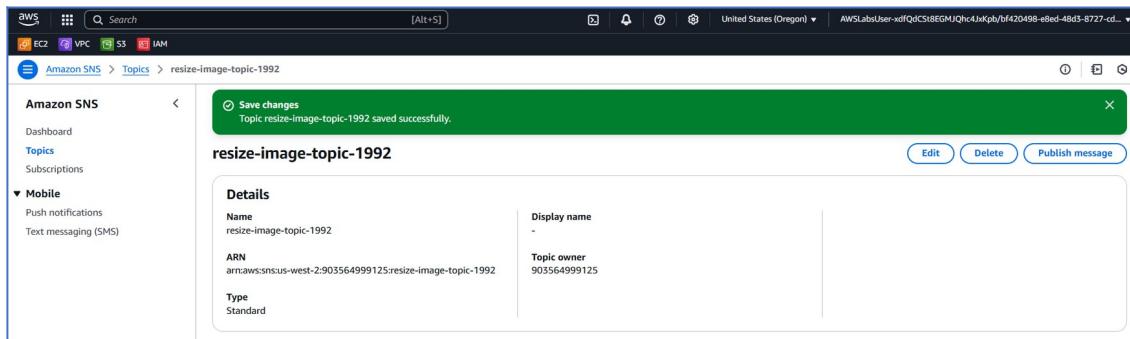
```

The new access policy introduces two key changes to enhance functionality and align with AWS security best practices. First, it replaces the condition "AWS:SourceOwner" with "AWS:SourceAccount" within the StringEquals clause. While both keys can be used to restrict access based on the AWS account of the calling service, "AWS:SourceAccount" is the more current and preferred condition, especially when configuring access for services like Amazon S3 and SNS. This change ensures that only AWS resources belonging to the account 903564999125 are allowed to interact with the SNS topic, thus providing a more explicit and secure method for cross-service communication. This update likely reflects AWS's recommendations to ensure compatibility and tighter control over resource access.

The second major addition is a new statement that explicitly allows the Amazon S3 service (s3.amazonaws.com) to publish messages to the SNS topic `resize-image-topic-1992`, but only if the publishing request originates from the same AWS account (903564999125). This addition is crucial for enabling S3 event-driven workflows, where an image upload to an S3 bucket can trigger an automatic notification to the SNS topic. This pattern is commonly used in image processing pipelines, where an uploaded image prompts further processing by downstream systems like AWS Lambda, SQS queues, or other microservices. The use of SourceAccount in the condition ensures that only the trusted, intended S3 buckets within the same account can publish to the topic, protecting against unauthorized or cross-account access.

Together, these changes make the policy more secure and functional, enabling specific integration between S3 and SNS for event notifications while following updated AWS best practices for access control.

- Choose **Save changes**.



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

- At the top of the AWS Management Console, in the search box, search for and choose **S3**.
- On the **Buckets** page, choose the bucket hyperlink with a name like `xxxxx-labbucket-XXXXX`.

The screenshot shows the AWS Management Console Buckets page. On the left, there's a sidebar with links for General purpose buckets, Storage Lens, and other AWS services. The main area displays an account snapshot and a list of general purpose buckets. One specific bucket, 'labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj', is highlighted with a red box.

Name	AWS Region	IAM Access Analyzer	Creation date
awslabs-resources-krxqqla59sui8d-us-east-1-903564999125	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	May 28, 2022, 01:12:34 (UTC+05:30)
awslabs-resources-r5b3y6ojjszcap-us-east-1-903564999125	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	October 25, 2023, 01:35:23 (UTC+05:30)
<b>labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj</b>	US West (Oregon) us-west-2	<a href="#">View analyzer for us-west-2</a>	April 16, 2025, 14:44:00 (UTC+05:30)
notes-bucket-689497816	Europe (Ireland) eu-west-1	<a href="#">View analyzer for eu-west-1</a>	November 14, 2023, 02:48:27 (UTC+05:30)

- Choose the **Properties** tab.

The screenshot shows the AWS Management Console Properties tab for the selected bucket. The sidebar remains the same. The main area shows the bucket's properties, including its overview, versioning settings, and multi-factor authentication (MFA) delete configuration.

**Bucket overview**

AWS Region	Amazon Resource Name (ARN)	Creation date
US West (Oregon) us-west-2	<a href="#">arn:aws:s3:::labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj</a>	April 16, 2025, 14:44:00 (UTC+05:30)

**Bucket Versioning**

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

**Bucket Versioning**  
Disabled

**Multi-factor authentication (MFA) delete**  
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)  
Disabled

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

Amazon S3 < Event notifications (0)

General purpose buckets

Directory buckets

Table buckets

Access Grants

Access Points

Object Lambda Access Points

Multi-Region Access Points

Name Event types Filters Destination type Destination

No event notifications

Choose Create event notification to be notified when a specific event occurs.

Create event notification

- Choose **Create event notification**.

Amazon S3 > Buckets > labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqq > Create event notification

**Create event notification** Info

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

**General configuration**

Event name  Event name can contain up to 255 characters.

Prefix - optional  Limit the notifications to objects with key starting with specified characters.

Suffix - optional  Limit the notifications to objects with key ending with specified characters.

**Event types**  
Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

**Object creation**

All object create events  Put

- In the General configuration section, do the following:

- o **Event name:** Enter **resize-image-event**.

- o **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).

- o **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.

## Create event notification Info

To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications.

### General configuration

#### Event name

Event name can contain up to 255 characters.

#### Prefix - optional

Limit the notifications to objects with key starting with specified characters.

#### Suffix - optional

Limit the notifications to objects with key ending with specified characters.

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

#### Event types

Specify at least one event for which you want to receive notifications. For each group, you can choose an event type for all events, or you can choose one or more individual events.

##### Object creation

All object create events  
s3:ObjectCreated:<sup>\*</sup>

- Put  
s3:ObjectCreated:Put
- Post  
s3:ObjectCreated:Post
- Copy  
s3:ObjectCreated:Copy
- Multipart upload completed  
s3:ObjectCreated:CompleteMultipartUpload

- 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-1992** 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.

#### Destination

Before Amazon S3 can publish messages to a destination, you must grant the Amazon S3 principal the necessary permissions to call the relevant API to publish messages to an SNS topic, an SQS queue, or a Lambda function. [Learn more](#)

##### Destination

Choose a destination to publish the event. [Learn more](#)

Lambda function

Run a Lambda function script based on S3 events.

SNS topic

Fanout messages to systems for parallel processing or directly to people.

SQS queue

Send notifications to an SQS queue to be read by a server.

##### Specify SNS topic

Choose from your SNS topics

Enter SNS topic ARN

##### SNS topic

[Cancel](#)

[Save changes](#)

- Choose **Save changes**.

The screenshot shows the AWS S3 Bucket Properties page for the bucket 'labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj'. A green success message at the top states: 'Successfully created event notification "resize-image-event". Operation successfully completed.' Below the message, the 'Properties' tab is selected. Under the 'Bucket overview' section, it shows the AWS Region as 'US West (Oregon) us-west-2', the Amazon Resource Name (ARN) as 'arn:aws:s3:::labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj', and the Creation date as 'April 16, 2025, 14:44:00 (UTC+05:30)'.

You have successfully created an Amazon S3 event notification.

## 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.

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

The screenshot shows the AWS Lambda Functions page. On the left sidebar, under 'Lambda', the 'Functions' section is selected. The main area displays a table of three existing Lambda functions: 'LabEnforcer\_Helper\_Function', 'LabEnforcer\_Function', and 'LabStack-bf420498-e8ed-48d3-8727-AWSLambdaFunction-eX2YT1DqCdx'. The table includes columns for Function name, Description, Package type, Runtime, and Last modified. A blue banner at the top provides information about Provisioned Mode for Kafka event source mapping (ESM).

- Choose **Create function**.
- In the **Create function** window, select **Author from scratch**.
- In the Basic information section, configure the following:
  - Function name:** Enter **CreateThumbnail**.
  - Runtime:** Choose **Python 3.9**.

The screenshot shows the AWS Lambda 'Create function' wizard. In the 'Basic information' step, the 'Author from scratch' option is selected. The 'Function name' field contains 'CreateThumbnail'. The 'Runtime' dropdown is set to 'Python 3.9'. The 'Execution role' section is expanded, showing the 'Use an existing role' option selected, with a role named 'LabStack-bf420498-e8ed-48d3-8727-c-LabExecutionRole-1In3DTqhtLm7' listed. The 'Additional Configurations' section is partially visible at the bottom.

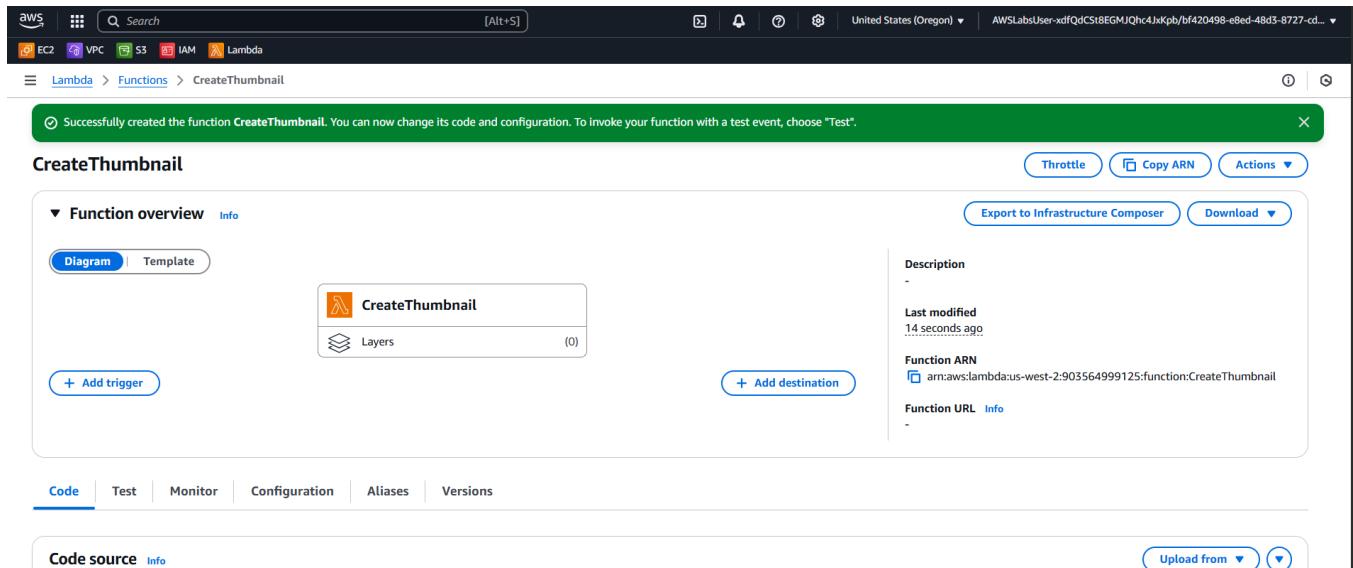
- o Expand the **Change default execution role** section.
- o **Execution role:** Select **Use an existing role**.
- o **Existing role:** Choose the role with the name like *XXXXX-LabExecutionRole-XXXXX*.

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**.

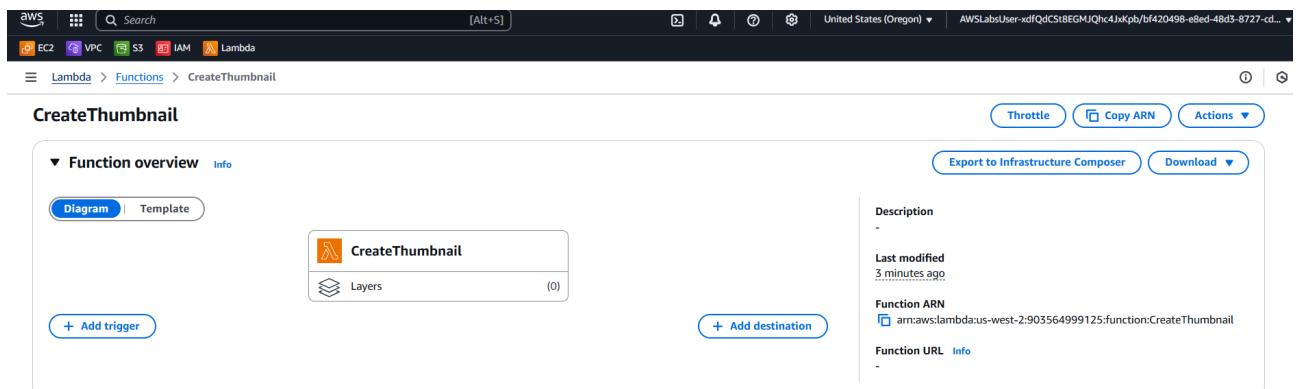
At the top of the page there is a message like, *Successfully created the function CreateThumbnail. You can now change its code and configuration. To invoke your function with a test event, choose "Test".*



## 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.

- Choose **Add trigger**,



and then configure the following:

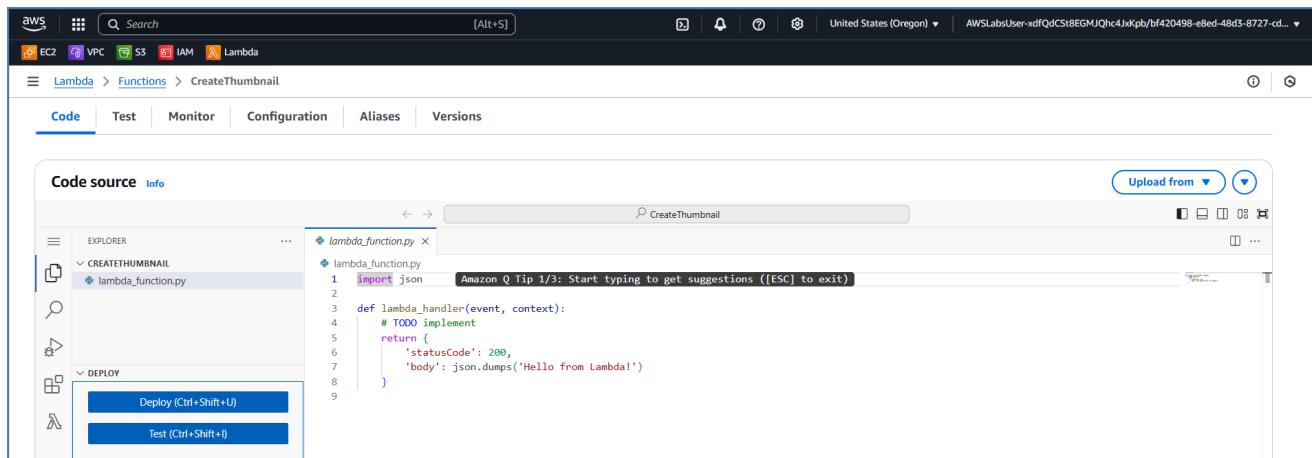
- For **Select a source**, choose **SQS**.
- For **SQS Queue**, choose **thumbnail-queue**.
- For **Batch size - optional**, enter **1**.

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

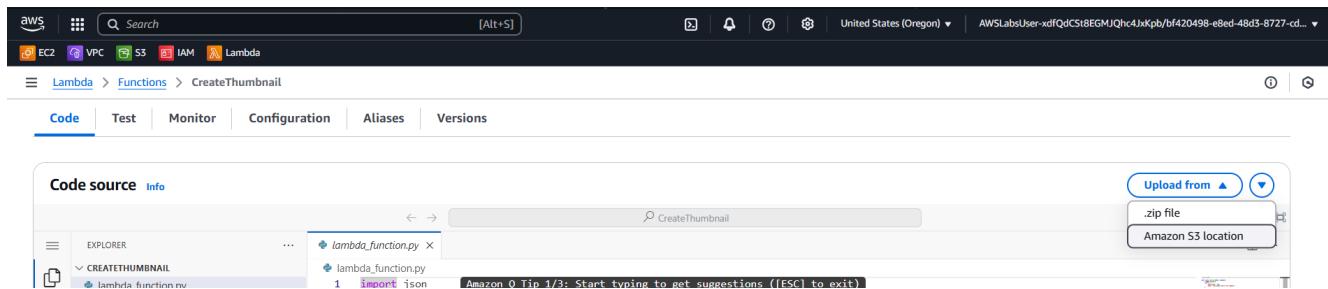
At the top of the page there is a message like, *The trigger thumbnail-queue was successfully added to function CreateThumbnail. The trigger is in a disabled state.*

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

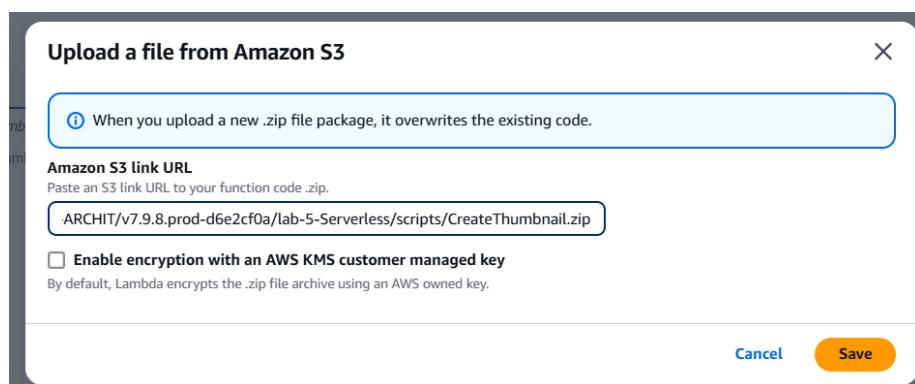
- Choose the **Code** tab.



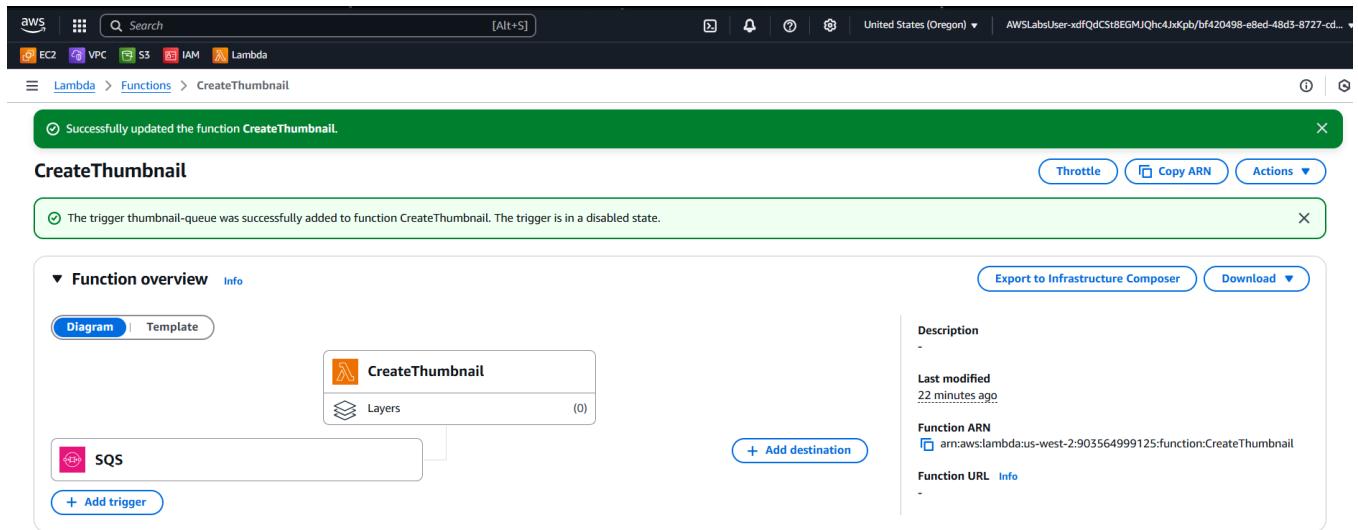
- Configure the following settings (and ignore any settings that are not listed):
- Copy the **CreateThumbnailZIPLocation** (`s3://us-west-2-tcprod/courses/ILT-TF-200-ARCHIT/v7.9.8.prod-d6e2cf0a/lab-5-Serverless/scripts/CreateThumbnail.zip`)
- Choose **Upload from**, and choose **Amazon S3 location**.



- Paste the **CreateThumbnailZIPLocation** (`s3://us-west-2-tcprod/courses/ILT-TF-200-ARCHIT/v7.9.8.prod-d6e2cf0a/lab-5-Serverless/scripts/CreateThumbnail.zip`) value you copied in the **Amazon S3 link URL** field.



- 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):
    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]
    )

```

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

**Code properties**

Package size  
16.5 MB

SHA256 hash  
pc7VUEGBnFYB1TSfxV1ZqXJE9jB5eYVzcRAkG2+/Ifc=

Last modified  
28 minutes ago

**Runtime settings**

Runtime  
Python 3.9

Handler  
lambda\_function.lambda\_handler

Architecture  
x86\_64

**Runtime management configuration**

**Runtime management configuration**

**Info**

**Edit** **Edit runtime management configuration**

**New runtime available**

A new runtime is available for your function's language: Python 3.13

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

- Choose **Save**.

**Edit runtime settings**

**Runtime settings**

Runtime  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.9

**New runtime available**

A new runtime is available for your function's language: Python 3.13

**Handler**

CreateThumbnail.handler

**Architecture**

Choose the instruction set architecture you want for your function code.

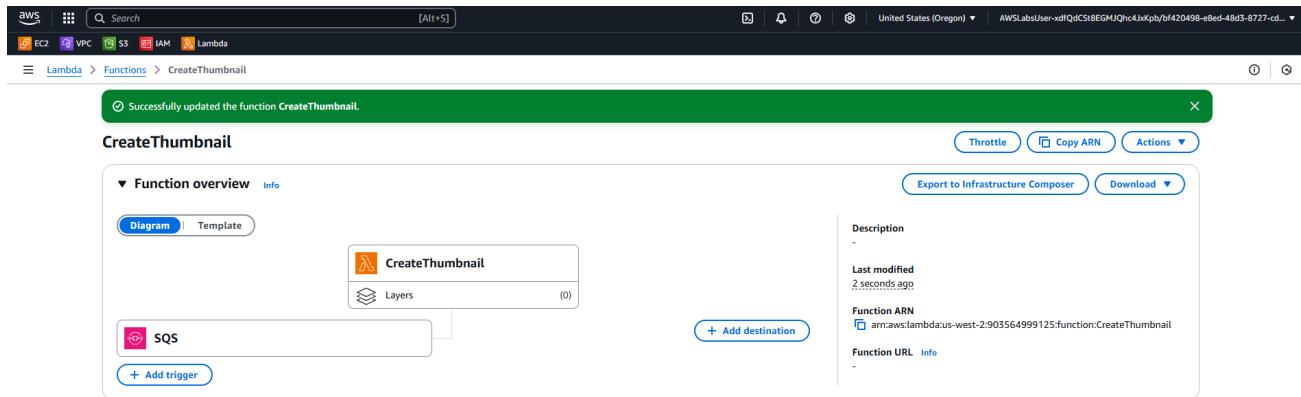
x86\_64  
arm64

**Note**

You can change either the function's runtime or the instruction set architecture in one update. To update both, you must repeat the update process.

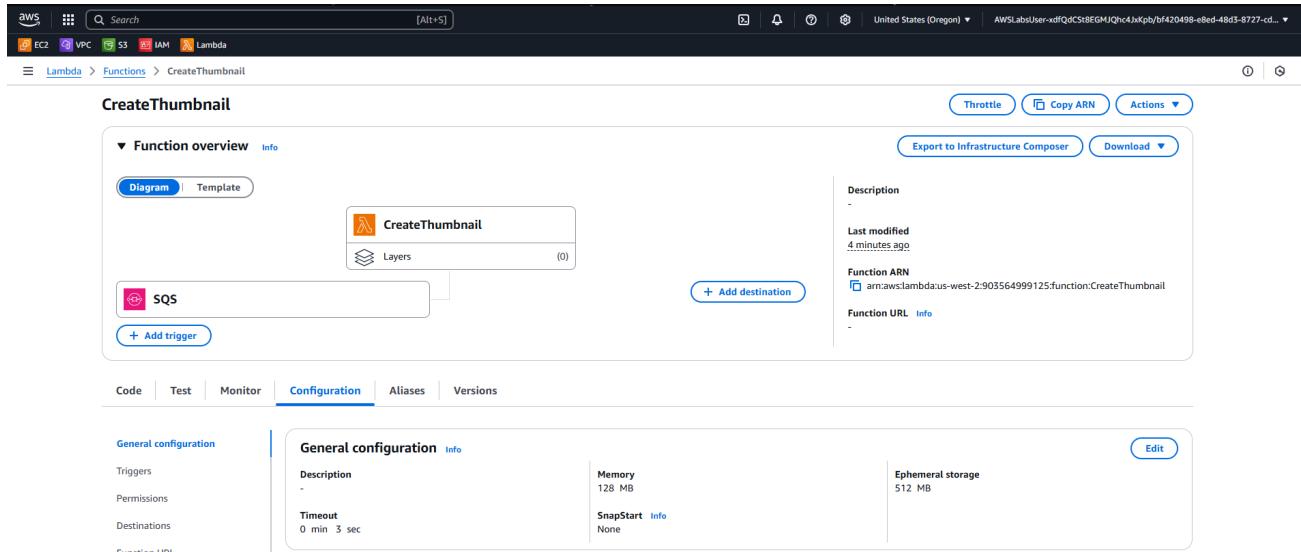
**Cancel** **Save**

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 (`CreateThumbnail.handler`), otherwise the Lambda function will not be found.

- Choose the **Configuration** tab.



- From the left navigation menu, choose **General configuration**.
- Choose **Edit**.

The screenshot shows the 'Edit basic settings' page for a Lambda function named 'CreateThumbnail'. The 'Basic settings' tab is selected. In the 'Description - optional' field, the text 'Create a thumbnail-sized image.' is entered. Other settings like Memory (128 MB), Ephemeral storage (512 MB), and Timeout (3 seconds) are also visible.

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

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

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

#### Edit basic settings

The screenshot shows the 'Edit basic settings' page again. This time, in the 'Execution role' section, the 'Existing role' dropdown is populated with 'LabStack-bf420498-e8ed-48d3-8727-c-LabExecutionRole-1In3DTqhtLm7'. The 'Save' button is highlighted in orange at the bottom right.

- Choose **Save**.

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

The screenshot shows the AWS Lambda Functions page. At the top, there's a green success message: "Successfully updated the function CreateThumbnail." Below it, the "CreateThumbnail" function is listed with its details. The "Description" is "Create a thumbnail-sized image". It was last modified "0 seconds ago". The "Function ARN" is arn:aws:lambda:us-west-2:903564999125:function:CreateThumbnail. The "Function URL" is listed as well. On the left, there's a diagram showing the function "CreateThumbnail" with a "Layers" section and an "SQS" trigger. Buttons for "Throttle", "Copy ARN", and "Actions" are at the top right.

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.

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

The screenshot shows the AWS Lambda Dashboard. On the left, there's a sidebar with "Lambda" selected, followed by "Dashboard", "Applications", and "Functions". Under "Additional resources", there are links for "Code signing configurations", "Event source mappings", "Layers", and "Replicas". Under "Related AWS resources", there's a link for "Step Functions state machines". The main area displays "Resources for United States (Oregon)" with 4 Lambda function(s), 16.5 MB of code storage (0% of 75 GB), 10 full account concurrency, and 5 unreserved account concurrency. Below this, there's a chart titled "Top 10 functions" showing errors, invocations, and concurrent executions over the last 3 hours. The chart indicates two functions: "1 - LabEnforcer\_Function" and "2 - CreateThumbnail".

- Choose **Create function**.
- In the **Create function** window, select **Author from scratch**.
- In the Basic information section, configure the following:
  - Function name:** Enter **CreateMobileImage**.
  - Runtime:** Choose **Python 3.9**.

The screenshot shows the AWS Lambda 'Create function' wizard. In the top navigation bar, 'Lambda' is selected under 'Functions'. The main section is titled 'Create function' with an 'Info' link. It asks to choose one of three options: 'Author from scratch' (selected), 'Use a blueprint', or 'Container image'. The 'Basic information' step is active, showing fields for 'Function name' (set to 'CreateMobileImage'), 'Runtime' (set to 'Python 3.9'), and 'Execution role'. A 'Change default execution role' section is expanded, showing options for 'Execution role' (with 'Use an existing role' selected) and 'Existing role' (with 'LabStack-bf420498-e8ed-48d3-8727-c-LabExecutionRole-1In3DTqhtLm7' selected). An 'Additional Configurations' section is also visible.

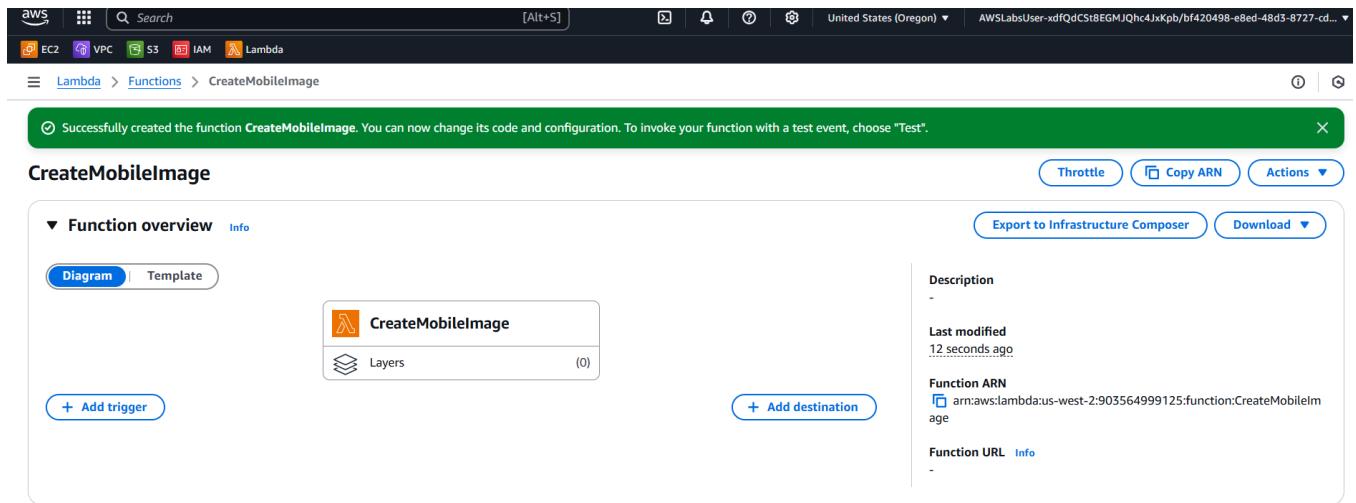
- o Expand the **Change default execution role** section.
- o **Execution role:** Select **Use an existing role**.
- o **Existing role:** Choose the role with the name like *XXXXX-LabExecutionRole-XXXXX*.

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.

- Choose **Create function**.

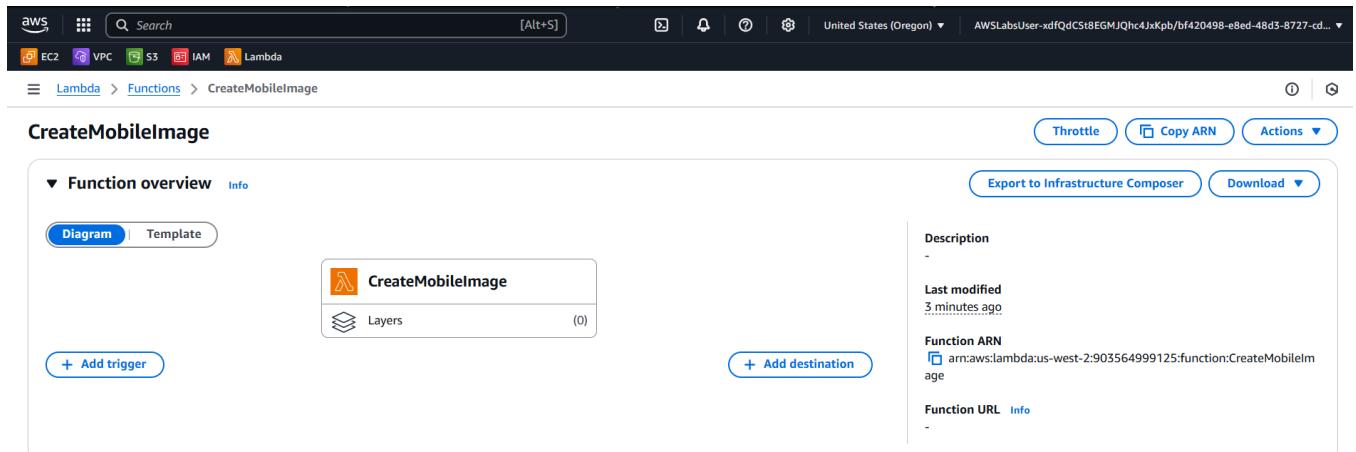
At the top of the page there is a message like, *Successfully created the function CreateMobileImage. You can now change its code and configuration. To invoke your function with a test event, choose "Test".*



## 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.

- 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**.

The screenshot shows the 'Trigger configuration' section of the Lambda trigger creation interface. It includes a dropdown menu for selecting the event source, which is set to 'SQS'. Below this, there's a search bar with the ARN of the SQS queue: 'arn:aws:sqs:us-west-2:903564999125:mobile-queue'. Under 'Event source mapping configuration', there are two checkboxes: 'Activate trigger' (checked) and 'Enable metrics' (unchecked). A dropdown for 'Batch size - optional' is set to '1'. A note at the bottom states: 'The maximum is 10,000 for standard queues and 10 for FIFO queues.'

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

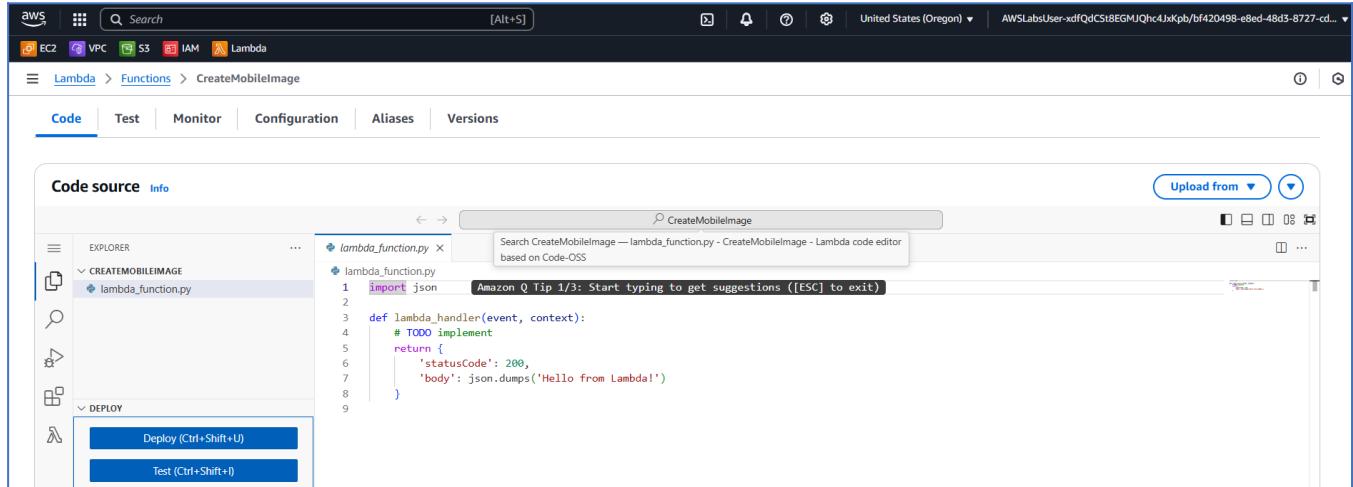
This screenshot shows the continuation of the trigger configuration page. It includes fields for 'Concurrent function instances' (set to 1), 'Report batch item failures - optional' (unchecked), and 'Filter criteria - optional' (unchecked). There is also a note about execution role permissions: 'In order to read from the SQS trigger, your execution role must have proper permissions.' At the bottom right are 'Cancel' and 'Add' buttons.

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.*

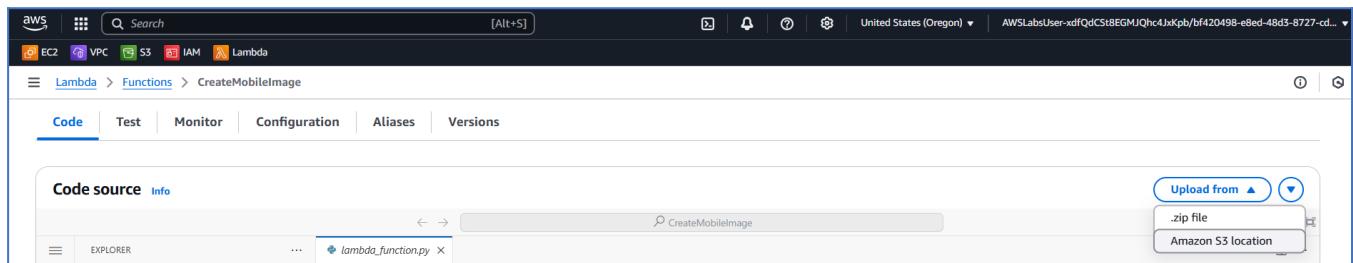
The screenshot shows the 'Function overview' section for the 'CreateMobileImage' function. It displays the function name, a diagram icon, and a note: 'The trigger mobile-queue was successfully added to function CreateMobileImage. The trigger is in a disabled state.' On the left, there are tabs for 'Diagram' (selected) and 'Template'. On the right, there are buttons for 'Throttle', 'Copy ARN', and 'Actions'. The 'Description' panel shows the last modified time as '6 minutes ago'. The 'Function ARN' is listed as 'arn:aws:lambda:us-west-2:903564999125:function:CreateMobileImage'. The 'Function URL' is shown as a link.

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

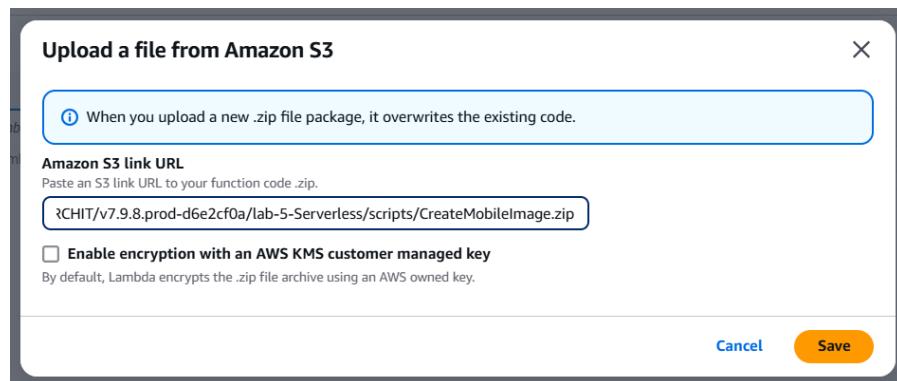
- Choose the **Code** tab.



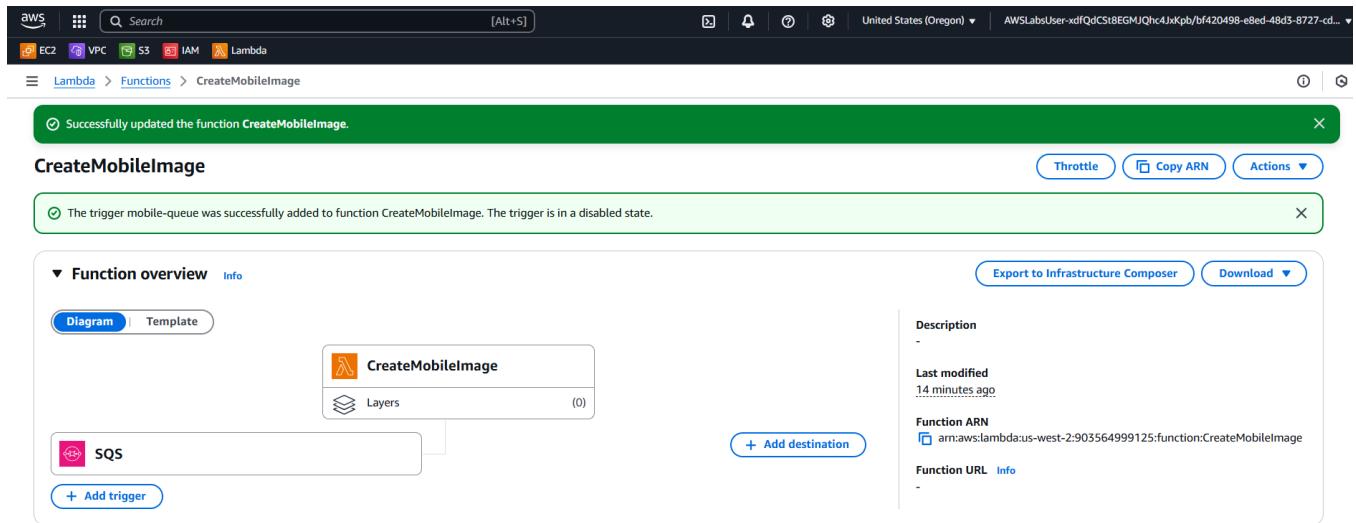
- Configure the following settings (and ignore any settings that are not listed):
- Copy the **CreateMobileImageZIPLocation** (`s3://us-west-2-tcprod/courses/ILT-TF-200-ARCHIT/v7.9.8.prod-d6e2cf0a/lab-5-Serverless/scripts/CreateMobileImage.zip`)
- Choose **Upload from**, and choose **Amazon S3 location**.



- Paste the **CreateMobileImageZIPLocation** (`s3://us-west-2-tcprod/courses/ILT-TF-200-ARCHIT/v7.9.8.prod-d6e2cf0a/lab-5-Serverless/scripts/CreateMobileImage.zip`) in the **Amazon S3 link URL** field.



- 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]
)

```

- 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 temporary storage
  - Resizes the image to dimensions 640×320 pixels using the Pillow library
  - Creates and uploads the resized image to a new folder called mobile/ with a filename prefix MobileImage-
- In the **Runtime settings** section, choose **Edit**.

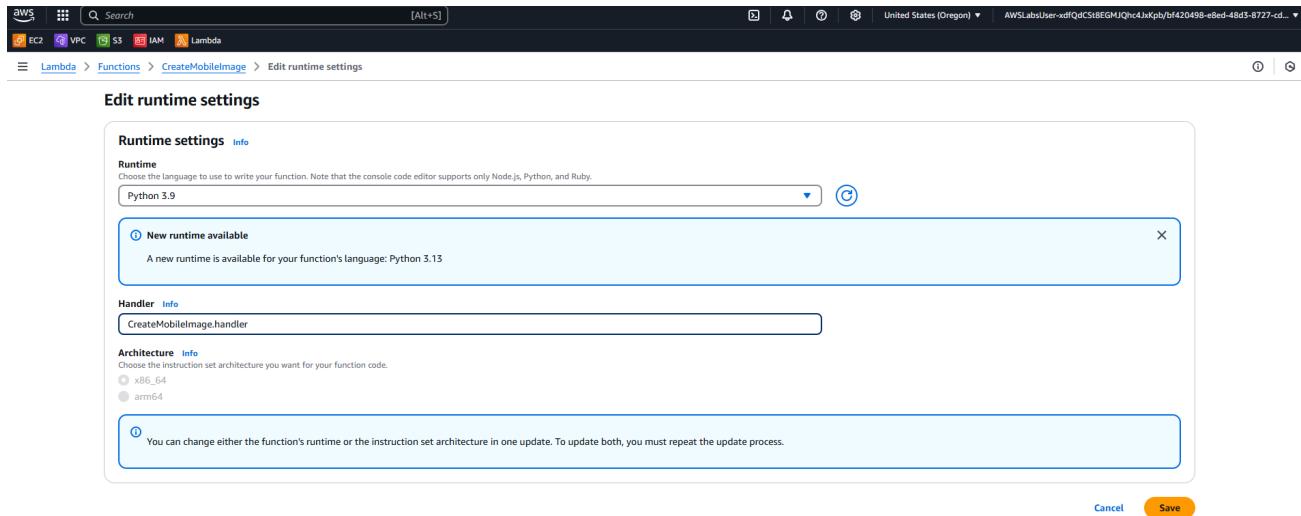
**Code properties**

- Package size: 16.5 MB
- SHA256 hash: YYLpDihvIvk7/qkdm8VSZxFlvKPQFxENx2pa6+UGFo=
- Last modified: 26 minutes ago
- Encryption with AWS KMS customer managed KMS key

**Runtime settings**

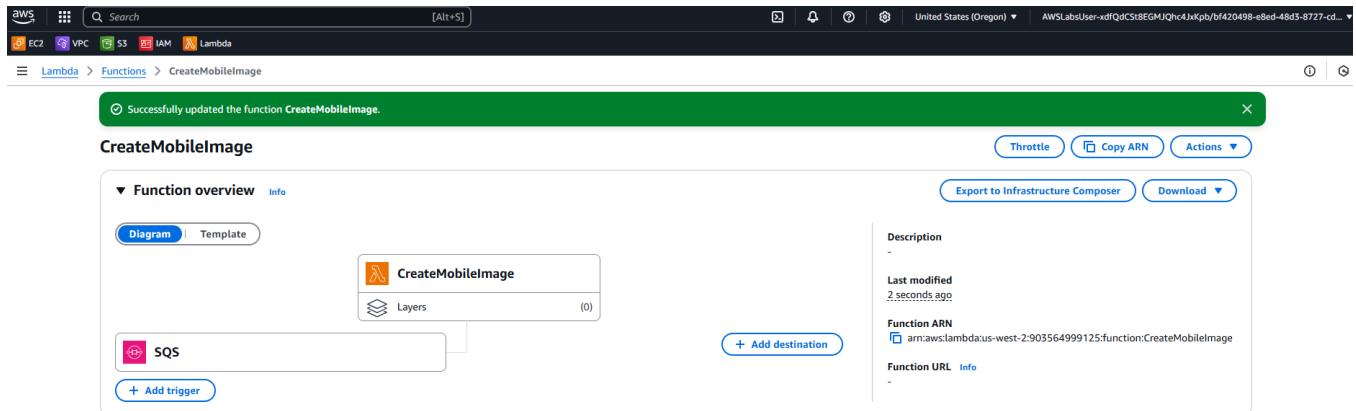
- Runtime: Python 3.9
- Handler: lambda\_function.lambda\_handler
- Architecture: x86\_64

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



- 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 (`CreateMobileImage.handler`), otherwise the Lambda function will not be found.

- Choose the **Configuration** tab.

- From the left navigation menu, choose **General configuration**.
- 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:

- 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.

## Edit basic settings

**Basic settings** [Info](#)

**Description - optional**  
Create a mobile friendly image

**Memory** [Info](#)  
Your function is allocated CPU proportional to the memory configured.  
 MB  
Set memory to between 128 MB and 10240 MB

**Ephemeral storage** [Info](#)  
You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#) [\\$](#)  
 MB  
Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

**SnapStart** [Info](#)  
Reduce startup time by having Lambda cache a snapshot of your function after it has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#). For Python and .NET runtimes, view [pricing](#) [\\$](#)  
 None  
Supported runtimes: .NET 8 (C#/F#/.NET), Java 11, Java 17, Java 21, Python 3.12, Python 3.13.

**Timeout**  
 min  sec

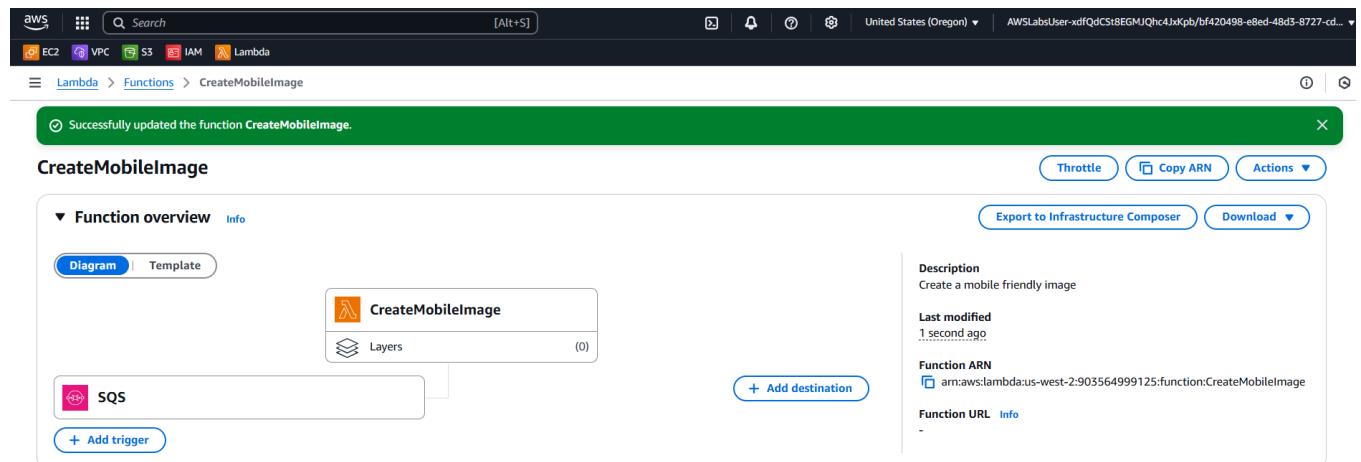
**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).  
 Use an existing role  
 Create a new role from AWS policy templates

**Existing role**  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.  
 [View](#) [Copy ARN](#)

[Cancel](#) [Save](#)

- Choose **Save**.

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

The screenshot shows the AWS Lambda Functions page. A green success message at the top reads "Successfully updated the function CreateMobileImage." Below it, the "CreateMobileImage" function card is visible. The "Function overview" tab is selected. The function name is "CreateMobileImage". It has one trigger, "SQS", and no layers. On the right side, there is a "Description" field containing "Create a mobile friendly image", a "Last modified" timestamp of "1 second ago", a "Function ARN" field with the value "arn:aws:lambda:us-west-2:903564999125:function:CreateMobileImage", and a "Function URL" link. There are also "Throttle", "Copy ARN", and "Actions" buttons.

The *CreateMobileImage* Lambda function has now been configured.

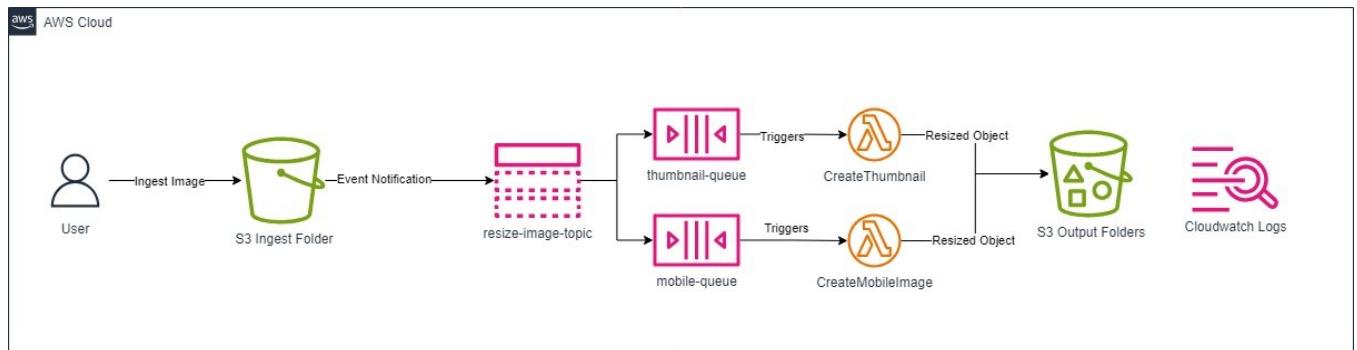
You have successfully created 2 AWS Lambda functions for the serverless architecture and set the appropriate SQS queue as trigger for their respective functions.

## 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 ( [MonaLisa.jpg](#) ) to test what you have built.

- At the top of the AWS Management Console, in the search box, search for and choose S3.
- In the S3 Management **Console**, choose the **xxxxx-labbucket-xxxxx** bucket hyperlink.

Screenshot of the AWS S3 Management Console. The left sidebar shows navigation options like EC2, VPC, S3, IAM, and Lambda. The main area displays the 'General purpose buckets' list. There are four buckets listed:

- awslabs-resources-krxqla59su18d-us-east-1-90356499125
- awslabs-resources-r5b3y6ojjszcap-us-east-1-90356499125
- labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj** (This bucket is highlighted with a red box.)
- notes-bucket-689497816

Each bucket row includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date.

Screenshot of the AWS S3 Management Console, showing the 'Objects' list for the 'labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj' bucket. The left sidebar shows the bucket's path: Amazon S3 > Buckets > labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj. The main area shows one object named 'ingest/'. The 'Actions' menu is open, showing options like Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.

- Choose the **ingest/** link.

The screenshot shows the AWS S3 console with the path `Amazon S3 > Buckets > labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqj > ingest/`. On the left, a sidebar lists 'General purpose buckets' and 'Storage Lens'. The main area is titled 'ingest/' and contains tabs for 'Objects' and 'Properties'. Under 'Objects', it says '(0)' and has a 'Find objects by prefix' search bar. Below is a table header with columns 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message states 'No objects' and 'You don't have any objects in this folder.' At the bottom right is a blue 'Upload' button.

- Choose **Upload**.

The screenshot shows the 'Upload' window within the AWS S3 console. The path is `Amazon S3 > Buckets > labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqj > ingest/ > Upload`. The title is 'Upload' with a 'Info' link. It says 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. [Learn more](#)'.

**Files and folders (0)**  
All files and folders in this table will be uploaded.  
A table with columns 'Name', 'Folder', 'Type', and 'Size'. A message at the bottom says 'No files or folders' and 'You have not chosen any files or folders to upload.'

**Destination** [Info](#)  
Destination  
`s3://labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqj/ingest/`  
▶ **Destination details**  
Bucket settings that impact new objects stored in the specified destination.

- In the **Upload** window, choose **Add files**.
- Browse to and choose the **MonaLisa.jpg** picture you downloaded.

The screenshot shows the AWS S3 console interface for uploading files to a bucket. A modal window is open, displaying a file selection dialog from a Windows file browser. The 'Downloads' folder is selected, showing files like 'MonaLisa.jpg'. The main S3 upload page shows the destination as 's3://labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqq/ingest/'. The 'Files and folders' table lists 'MonaLisa.jpg' with details: Type: image/jpeg, Size: 77.4 KB. The 'Upload' button is highlighted in orange at the bottom right.

- Choose **Upload**.

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

The screenshot shows the AWS S3 console after the upload has completed successfully. A green banner at the top says 'Upload succeeded'. Below it, the 'Upload: status' section provides a summary: 'Destination' is 's3://labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqq/ingest/' and the upload was 'Succeeded' with '1 file, 77.4 KB (100.0%)'. The 'Files and folders' table shows the uploaded file 'MonaLisa.jpg' with details: Name: MonaLisa.jpg, Type: image/jpeg, Size: 77.4 KB, Status: Succeeded.

You have successfully uploaded JPG images to S3 bucket.

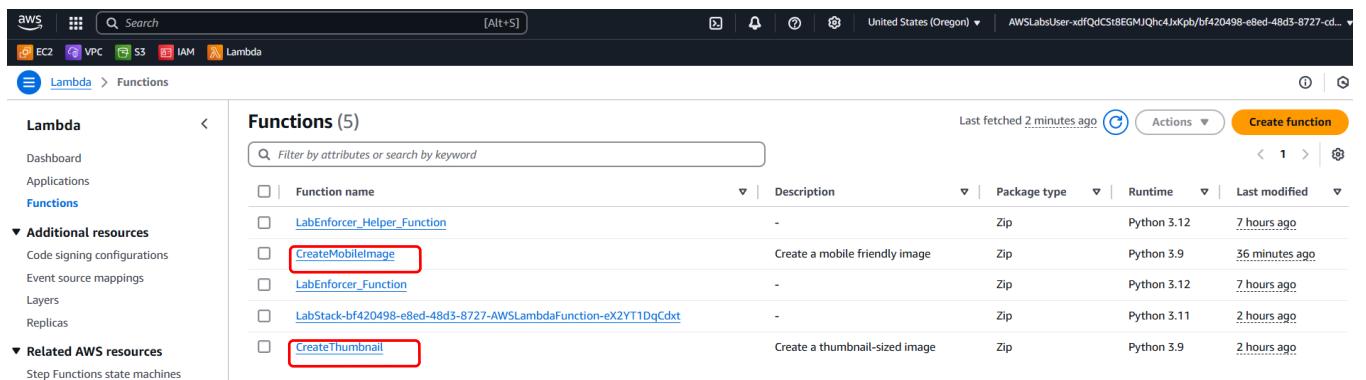
## Task 6: Validate the processed file

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.

- At the top of the AWS Management Console, in the search box, search for and choose **Lambda**.
- Choose the hyperlink for one of your **Create-** functions.



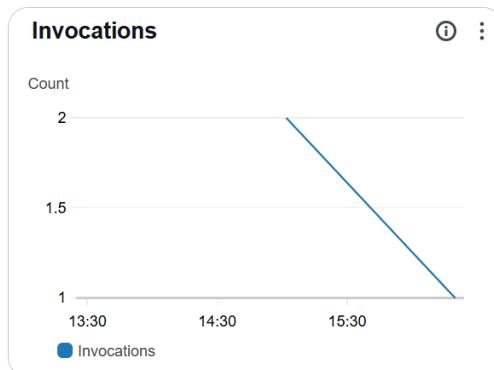
The screenshot shows the AWS Lambda Functions page. On the left, there's a sidebar with options like Dashboard, Applications, Functions (which is selected), Additional resources, and Related AWS resources. The main area is titled "Functions (5)" and lists the following functions:

Function name	Description	Package type	Runtime	Last modified
LabEnforcer_Helper_Function	-	Zip	Python 3.12	7 hours ago
CreateMobileImage	Create a mobile friendly image	Zip	Python 3.9	36 minutes ago
LabEnforcer_Function	-	Zip	Python 3.12	7 hours ago
LabStack-bf420498-e8ed-48d3-8727-AWSLambdaFunction-eX2YT1DqCdxt	-	Zip	Python 3.11	2 hours ago
CreateThumbnail	Create a thumbnail-sized image	Zip	Python 3.9	2 hours ago

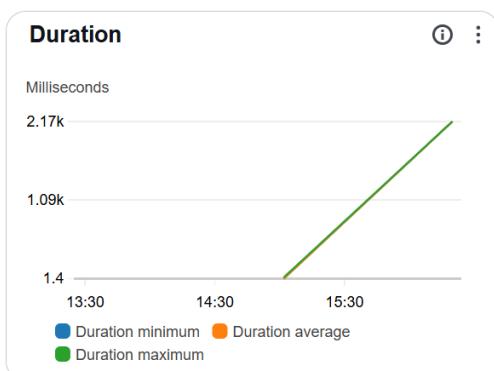
- Choose the **Monitor** tab.

The console displays graphs showing the following:

- *Invocations*: The number of times that the function was invoked.



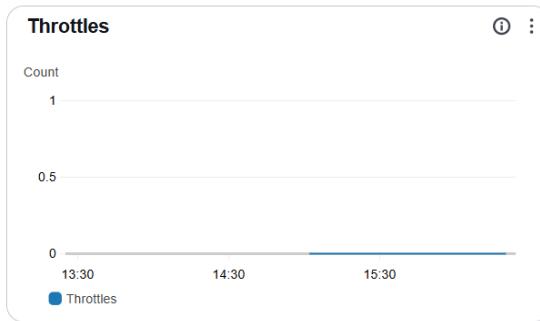
- *Duration*: The average, minimum, and maximum execution times.



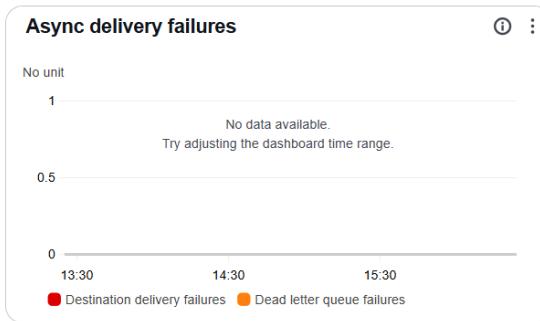
- *Error count and success rate (%)*: The number of errors and the percentage of executions that completed without error.



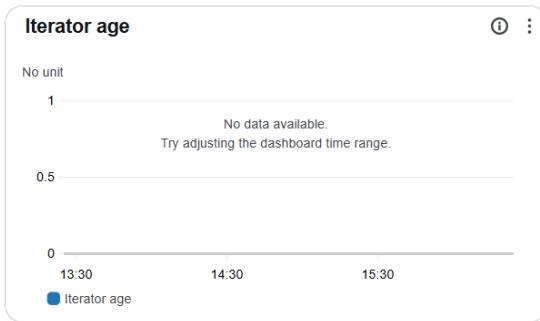
- **Throttles:** When too many functions are invoked simultaneously, they are throttled. The default is 1000 concurrent executions.



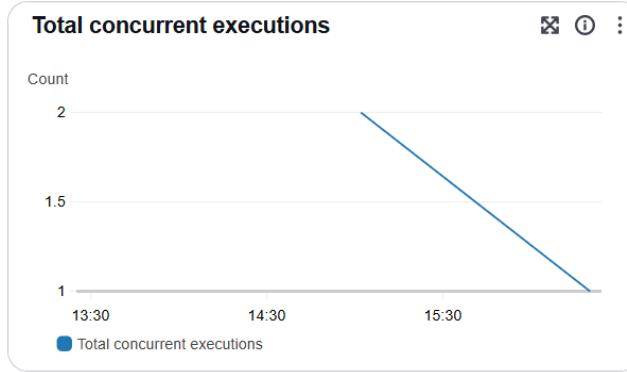
- **Async delivery failures:** The number of errors that occurred when Lambda attempted to write to a destination or dead-letter queue.



- **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*.

- Choose **View CloudWatch logs**.

The screenshot shows two consecutive screenshots of the AWS CloudWatch interface. The top part is the 'Monitor' tab under 'Lambda > Functions > CreateMobileImage'. It features a navigation bar with EC2, VPC, S3, IAM, and Lambda. Below it are tabs for Code, Test, Monitor (selected), Configuration, Aliases, and Versions. The 'Monitor' section includes links for View CloudWatch logs, View Application Signals, View X-Ray traces, View Lambda Insights, and View CodeGuru profiles. A search bar and filter metrics by Function are also present. The bottom part is the 'Log streams' page for the '/aws/lambda/CreateMobileImage' log group. It has a sidebar with CloudWatch navigation, Favorites and recent logs, AI Operations, Alarms, Metrics, X-Ray traces, Events, and Application Signals. The main area shows log group details like ARN, creation time (2 hours ago), retention (Never expire), and stored bytes. It also lists metric filters, subscription filters, Contributor Insights rules, KMS key ID, and anomaly detection. Below this are tabs for Log streams, Tags, Anomaly detection, Metric filters, Subscription filters, Contributor Insights, Data protection, and Field indexes. The 'Log streams' tab is selected, showing three entries: '2025/04/16/[LATEST]24cd565ceab5426db4f444aa276bb412' (last event time 2025-04-16 16:21:03 UTC), '2025/04/16/[LATEST]53be456683af4897a7d6804b80c70103' (2025-04-16 15:02:48 UTC), and '2025/04/16/[LATEST]b76a6fc8251f43989e91e013f217a897' (2025-04-16 15:02:48 UTC). Action buttons include Create log stream and Search all log streams.

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

The screenshot shows the AWS CloudWatch Log events interface. The left sidebar navigation includes CloudWatch, Favorites and recent, Dashboards, AI Operations, Alarms, Logs (selected), Log groups, Metrics, X-Ray traces, Events, and Application Signals. The main content area displays log events for the path /aws/lambda/CreateMobileImage on 2025/04/16. A search bar at the top allows filtering by search terms. Below it are buttons for Actions, Start tailing, and Create metric filter. A timestamp dropdown is also present. The log table has columns for Timestamp and Message. The first message is a placeholder: "No older events at this moment. Retry". Subsequent messages show the Lambda function starting, processing a labstack request, ingest/MonaLisa.jpg, and ending. The last message is a REPORT line with detailed metrics: Duration: 2169.77 ms, Billed Duration: 2170 ms, Memory Size: 128 MB, Max Memory Used: 86 MB, and Init Duration: 725.48 ms.

- 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.

**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

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

**Amazon S3**

**General purpose buckets**

**Storage Lens**

**Feature spotlight**

**Account snapshot - updated every 24 hours**

**General purpose buckets** (4) **Directory buckets**

Name	AWS Region	IAM Access Analyzer	Creation date
awslabs-resources-krxqla59sui8d-us-east-1-903564999125	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	May 28, 2022, 01:12:34 (UTC+05:30)
awslabs-resources-r5b3y6ojiszcap-us-east-1-903564999125	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	October 25, 2023, 01:35:23 (UTC+05:30)
labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj	US West (Oregon) us-west-2	<a href="#">View analyzer for us-west-2</a>	April 16, 2025, 14:44:00 (UTC+05:30)
notes-bucket-689497816	Europe (Ireland) eu-west-1	<a href="#">View analyzer for eu-west-1</a>	November 14, 2023, 02:48:27 (UTC+05:30)

- Choose the hyperlink for **xxxxxxxx-labbucket-xxxxxx** to enter the bucket.

**Amazon S3**

**Buckets** > **labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj**

**Objects** (3)

Name	Type	Last modified	Size	Storage class
ingest/	Folder	-	-	-
mobile/	Folder	-	-	-
thumbnail/	Folder	-	-	-

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

**Amazon S3**

**Buckets** > **labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqqj** > **mobile/**

**Objects** (1)

Name	Type	Last modified	Size	Storage class
MobileImage-MonaLisa.jpg	jpg	April 16, 2025, 21:51:03 (UTC+05:30)	11.0 KB	Standard

The screenshot shows the AWS Management Console interface for Amazon S3. The top navigation bar includes links for EC2, VPC, S3, IAM, and Lambda. The main navigation bar shows the path: Amazon S3 > Buckets > labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqj > thumbnail/. The left sidebar under 'Amazon S3' lists General purpose buckets, Directory buckets, Table buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. The main content area displays a table titled 'Objects (1)'. The table has columns for Name, Type, Last modified, Size, and Storage class. One object, 'Thumbnail-MonaLisa.jpg', is listed with a type of jpg, last modified on April 16, 2025, at 21:51:03 (UTC+05:50), a size of 2.8 KB, and a storage class of Standard. A 'Copy S3 URI' button is located in the top right corner of the main content area.

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

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

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

- At the top of the AWS Management Console, in the search box, search for and choose **S3**.
- On the **Buckets** page, choose the **xxxxx-labbucket-xxxxx** bucket.

The screenshot shows the AWS Management Console interface for Amazon S3. The top navigation bar includes links for EC2, VPC, S3, IAM, and Lambda. The main navigation bar shows the path: Amazon S3 > Buckets. The left sidebar under 'Amazon S3' lists General purpose buckets, Directory buckets, Table buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. The right sidebar includes sections for Block Public Access settings for this account and Storage Lens (Dashboards, Storage Lens groups, AWS Organizations settings). The main content area displays a table titled 'General purpose buckets (4)'. The table has columns for Name, AWS Region, IAM Access Analyzer, and Creation date. Four buckets are listed: 'awslabs-resources-krxqqla59sui8d-us-east-1\_90356499125' (US East (N. Virginia) us-east-1), 'awslabs-resources-r5b3y6ojjszcap-us-east-1\_90356499125' (US East (N. Virginia) us-east-1), 'labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqj' (US West (Oregon) us-west-2), and 'notes-bucket-689497816' (Europe (Ireland) eu-west-1). The fourth bucket, 'labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqj', is highlighted with a red box. A 'View Storage Lens dashboard' button is located in the top right corner of the main content area.

- Choose the **Management** tab.

The screenshot shows the AWS S3 Management console. The left sidebar has sections for General purpose buckets, Storage Lens, and replication rules. The main area is titled 'labstack-bf420498-e8ed-48d3-8727-cd8a979-labbucket-bdpda92jmqj'. The 'Management' tab is selected. Under 'Lifecycle configuration', it says 'To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.' Below this is a table for 'Lifecycle rules' with columns for Lifecycle rule name, Status, Scope, Current version ac..., Nonconcurrent versio..., Expired object del..., and Incomplete multi...'. A note says 'No lifecycle rules' and 'There are no lifecycle rules for this bucket.' A 'Create lifecycle rule' button is at the bottom.

- In the **Lifecycle rules** section, choose **Create lifecycle rule**

The screenshot shows the 'Create lifecycle rule' configuration page. It has sections for 'Lifecycle rule configuration', 'Lifecycle rule name' (with a text input field 'Enter rule name' and note 'Up to 255 characters'), 'Scope' (radio buttons for 'Limit the scope of this rule using one or more filters' (selected) and 'Apply to all objects in the bucket'), 'Filter type' (radio buttons for 'Prefix' (selected) and 'Object tags'), and 'Prefix' (text input field 'Enter prefix' with note 'Add filter to limit the scope of this rule to a single prefix.' and note 'Don't include the bucket name in the prefix. Using certain characters in key names can cause problems with some applications and protocols. Learn more [2]')).

- 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**.
  - In the **Filter type** section, configure the following:
    - Prefix:** Enter **ingest/**.

## Create lifecycle rule [Info](#)

### Lifecycle rule configuration

#### Lifecycle rule name

cleanup

Up to 255 characters

#### Choose a rule scope

- Limit the scope of this rule using one or more filters
- Apply to all objects in the bucket

#### Filter type

You can filter objects by prefix, object tags, object size, or whatever combination suits your usecase.

#### Prefix

Add filter to limit the scope of this rule to a single prefix.

ingest/

Don't include the bucket name in the prefix. Using certain characters in key names can cause problems with some applications and protocols. [Learn more](#)

- In the **Lifecycle rule actions** section, configure the following:
  - o Select **Expire current versions of objects** and **Permanently delete noncurrent versions of objects**.
  - o Enter the following values in the new boxes that open.
    - **Days after object creation:** Enter **30**.
    - **Days after objects become noncurrent:** Enter **1**.

### Lifecycle rule actions

Choose the actions you want this rule to perform.

- Transition current versions of objects between storage classes  
This action will move current versions.
- Transition noncurrent versions of objects between storage classes  
This action will move noncurrent versions.
- Expire current versions of objects
- Permanently delete noncurrent versions of objects
- Delete expired object delete markers or incomplete multipart uploads  
These actions are not supported when filtering by object tags or object size.

### Expire current versions of objects

For version-enabled buckets, Amazon S3 adds a delete marker and the current version of an object is retained as a noncurrent version. For non-versioned buckets, Amazon S3 permanently removes the object. [Learn more](#)

#### Days after object creation

30

### Permanently delete noncurrent versions of objects

Choose when Amazon S3 permanently deletes specified noncurrent versions of objects. [Learn more](#)

#### Days after objects become noncurrent

1

#### Number of newer versions to retain - Optional

Number of versions

Can be 1 to 100 versions. All other noncurrent versions will be moved.

### Review transition and expiration actions

#### Current version actions

##### Day 0

- Objects uploaded



##### Day 30

- Objects expire

#### Noncurrent versions actions

##### Day 0

- Objects become noncurrent



##### Day 1

- 0 newest noncurrent versions are retained
- All other noncurrent versions are permanently deleted

[Cancel](#) [Create rule](#)

- Choose **Create rule**

The screenshot shows the AWS Management Console interface for managing S3 buckets. The user is in the 'Lifecycle configuration' section of a specific bucket. A green success message at the top states: "The rule 'cleanup' has been successfully added and the lifecycle configuration has been updated. It may take some time for the configuration to be updated. Refresh the lifecycle rules list if changes to the configuration aren't displayed." Below this, the 'Lifecycle configuration' page displays the 'Lifecycle rules (1)' section. One rule named 'cleanup' is listed, which is enabled, filtered, and set to expire objects permanently. The 'Review transition and expiration actions' section shows that objects uploaded daily are retained for 30 days before expiring.

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

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

The screenshot shows the AWS SNS Dashboard. On the left, there's a navigation pane with options like Dashboard, Topics, Subscriptions, and Mobile. A blue banner at the top right says "New Feature" with the text "Amazon SNS now supports High Throughput FIFO topics. Learn more". Below the banner, the dashboard shows "Resources for us-west-2": 1 Topic, 0 Platform applications, and 2 Subscriptions.

- In the left navigation pane, choose **Subscriptions**

The screenshot shows the AWS SNS Subscriptions page. It lists two confirmed subscriptions under the "us-west-2" region. The first subscription, "2de22532-1132-406e-9c36-2...", has an endpoint of "arn:aws:sqs:us-west-2:903564..." and is associated with the topic "resize-image-topic-1992". The second subscription, "eb00d873-72e1-42c9-bae0-f...", also has an endpoint of "arn:aws:sqs:us-west-2:903564..." and is associated with the same topic.

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

The screenshot shows the "Create subscription" page. In the "Details" section, the "Topic ARN" field contains "MyTopic". The "Protocol" dropdown is set to "Email". A note below the protocol field says "After your subscription is created, you must confirm it." In the "Subscription filter policy - optional" section, it says "This policy filters the messages that a subscriber receives." In the "Redrive policy (dead-letter queue) - optional" section, it says "Send undeliverable messages to a dead-letter queue." At the bottom right, there are "Cancel" and "Create subscription" buttons.

- 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.

**Details**

**Topic ARN**  
arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992

**Protocol**  
The type of endpoint to subscribe  
Email

**Endpoint**  
An email address that can receive notifications from Amazon SNS.  
vivek.velturi@gmail.com

ⓘ After your subscription is created, you must confirm it. [Info](#)

► **Subscription filter policy - optional** [Info](#)  
This policy filters the messages that a subscriber receives.

► **Redrive policy (dead-letter queue) - optional** [Info](#)  
Send undeliverable messages to a dead-letter queue.

[Cancel](#) [Create subscription](#)

- Choose **Create subscription**.

**Subscription to resize-image-topic-1992 created successfully.**  
The ARN of the subscription is arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992:a7740ee3-e257-4979-9ac3-6e22d09298d2.

**Subscription: a7740ee3-e257-4979-9ac3-6e22d09298d2**

**Details**

**ARN**  
arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992:a7740ee3-e257-4979-9ac3-6e22d09298d2

**Endpoint**  
vivek.velturi@gmail.com

**Topic**  
resize-image-topic-1992

**Subscription Principal**  
arn:aws:iam::903564999125:role/AWSLabsUser-xdfQdCSt8EGMJQhc4JxKpb

**Status**  
Pending confirmation

**Protocol**  
EMAIL

[Edit](#) [Delete](#)

[Subscription filter policy](#) [Redrive policy \(dead-letter queue\)](#)

**Subscription filter policy** [Info](#)  
This policy filters the messages that a subscriber receives.

No filter policy configured for this subscription.  
To apply a filter policy, edit this subscription.

[Edit](#)

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.

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



## AWS Notification - Subscription Confirmation



AWS Notifications <no-reply@sns.amazonaws.com>  
to me ▾

10:45 PM (1 minute ago)



You have chosen to subscribe to the topic:

**arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992**

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):

[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

[Reply](#)

[Forward](#)



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



Simple Notification Service

### Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992:a7740ee3-e257-4979-9ac3-6e22d09298d2

If it was not your intention to subscribe, [click here to unsubscribe](#).

A screenshot of the AWS SNS console. The left sidebar shows "Amazon SNS" with "Subscriptions" selected. The main panel shows a subscription with the ARN: "arn:aws:sns:us-west-2:903564999125:resize-image-topic-1992:a7740ee3-e257-4979-9ac3-6e22d09298d2". The status is "Confirmed" (highlighted with a red box). The protocol is "EMAIL". The "Subscription filter policy" section is expanded, showing a note: "This policy filters the messages that a subscriber receives." and "No filter policy configured for this subscription. To apply a filter policy, edit this subscription." An "Edit" button is visible at the bottom of this section.