AWS re:Start

CHALLENGE LAB

AWS Lambda Exercise

WEEK 9

aws re/start

Overview

Your Challenge

- Create a Lambda function to count the number of words in a text file. The general steps are as follows:
  - Use the AWS Management Console to develop a Lambda function in Python and create the function's required resources.
  - Report the word count in an email by using an SNS topic. Optionally, also send the result in an SMS (text) message.
  - Format the response message as follows: The word count in the <textFileName> file is nnn.
  - Enter the following text as the email subject: Word Count Result. Automatically invoke the function when the text file is uploaded to an S3 bucket.
- Test the function by uploading a few sample text files with different word counts to the S3 bucket.
- Forward the email that one of your tests produces and a screenshot of your Lambda function to your instructor.

Topics covered

- Create a Lambda function.
- Configure an Amazon Simple Storage Service (Amazon S3) bucket to invoke a Lambda function when a text file is uploaded to the S3 bucket.
- Create an Amazon Simple Notification Service (Amazon SNS) topic to report the word count in an email.

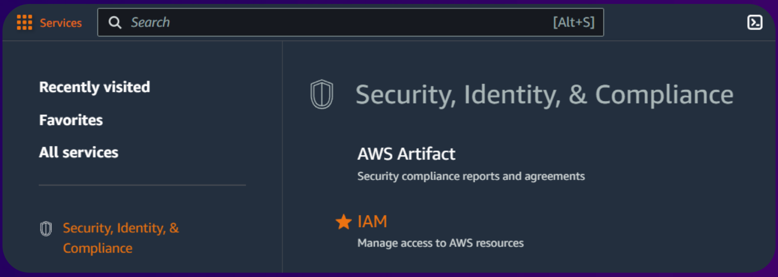
aws re/start

Task 1

Observing the IAM role settings

Step 1: Access the AWS Management Console

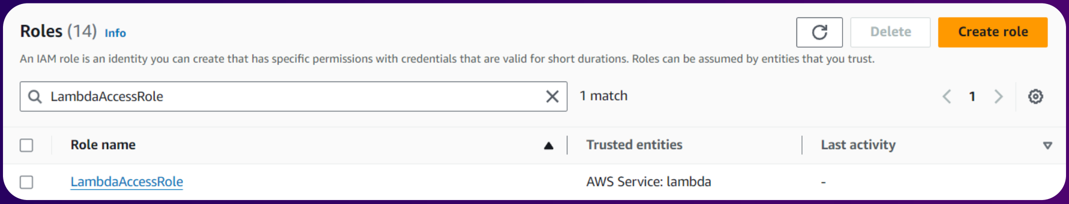
Open the AWS Management Console, and select IAM.



Services
Search
[Alt+S]
Recently visited
Favorites
All services
Security, Identity, & Compliance
Security, Identity, & Compliance
AWS Artifact
Security compliance reports and agreements
IAM
Manage access to AWS resources

Step 2: Review Role

Navigate to the **Roles** section, and review the **LambdaAccessRole** role.



An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

LambdaAccessRole — 1 match

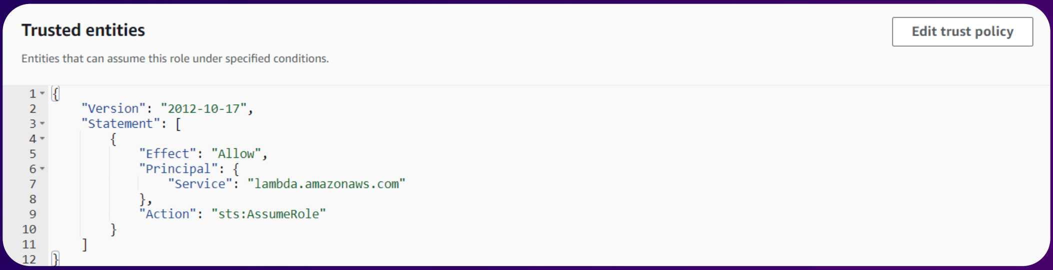
Role name	Trusted entities	Last activity
LambdaAccessRole	AWS Service: lambda	-

Task 1

Observing the IAM role settings

Step 3: Review Trusted entities

Choose the **LambdaAccessRole** role, and choose the **Trust relationships** tab, and notice that lambda.amazonaws.com is listed as a trusted entity, which means that the Lambda service can use this role.

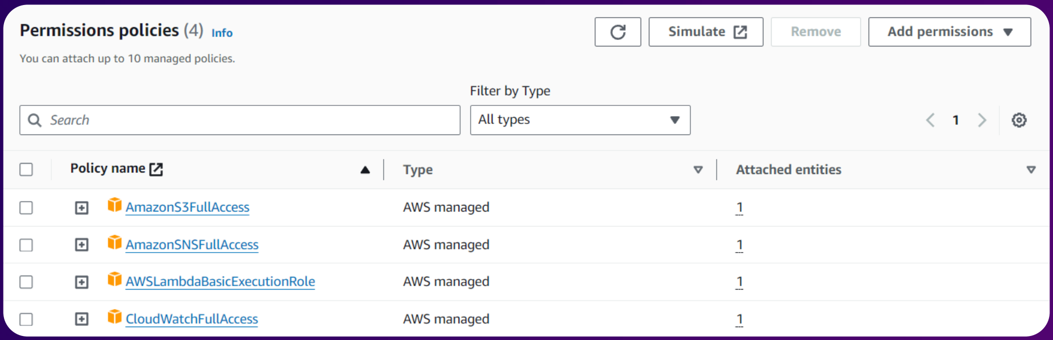


Entities that can assume this role under specified conditions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Step 4: Review Permissions policies

Choose the **Permissions** tab, and review the four permissions policies assigned to the **LambdaAccessRole** role.



You can attach up to 10 managed policies.

Filter by Type: All types

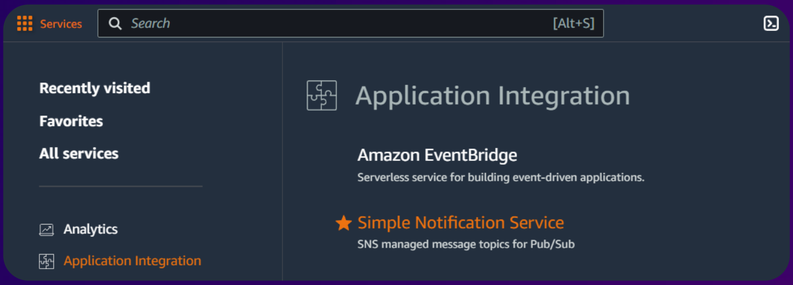
Policy name	Type	Attached entities
AmazonS3FullAccess	AWS managed	1
AmazonSNSFullAccess	AWS managed	1
AWSLambdaBasicExecutionRole	AWS managed	1
CloudWatchFullAccess	AWS managed	1

Task 2

Configuring notifications

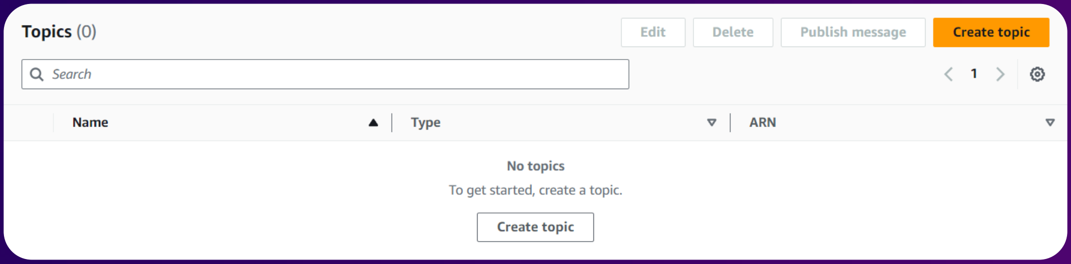
Step 1: Access the Simple Notification Service

In the AWS Management Console, select Simple Notification Service.



Step 2: Create topic

Navigate to the **Topics** section, and select Create topic.

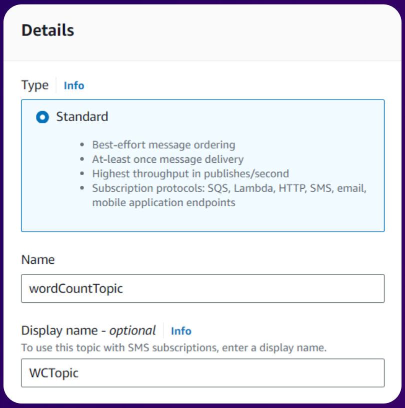


Task 2

Configuring notifications

Step 3: Topic Details

In the **Details** section, configure the following settings.



Details

Type Info

● Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

wordCountTopic

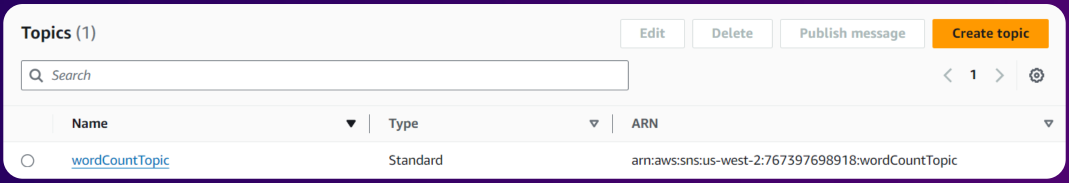
Display name - *optional* Info

To use this topic with SMS subscriptions, enter a display name.

WCTopic

Step 4: Review Topic Creation

Review the newly created **wordCountTopic**, and make note of the ARN value.



Topics (1)

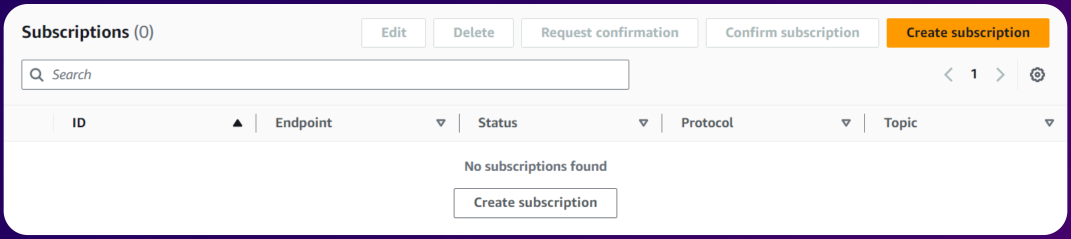
Name	Type	ARN
wordCountTopic	Standard	arn:aws:sns:us-west-2:767397698918:wordCountTopic

Task 2

Configuring notifications

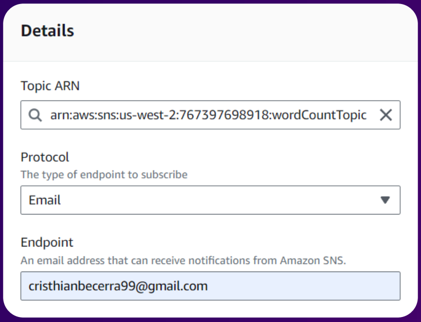
Step 5: Create subscriptions

Navigate to the **Subscriptions** section, and select Create subscription.



Step 6: Create Email Subscription

In the **Details** section, configure the following settings.



Details

Topic ARN: arn:aws:sns:us-west-2:767397698918:wordCountTopic

Protocol
The type of endpoint to subscribe
Email

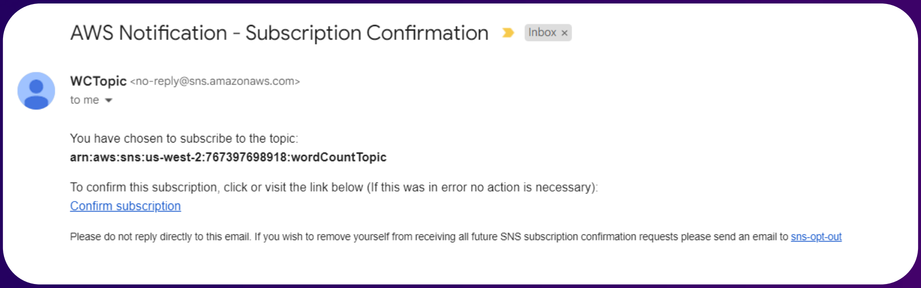
Endpoint
An email address that can receive notifications from Amazon SNS.
cristhianbecerra99@gmail.com

Task 2

Configuring notifications

Step 7: Check your email inbox

Check the inbox for the email address that you provided. You should see an email from WCTopic with the subject "AWS Notification - Subscription Confirmation."



AWS Notification - Subscription Confirmation

WCTopic <no-reply@sns.amazonaws.com>
to me

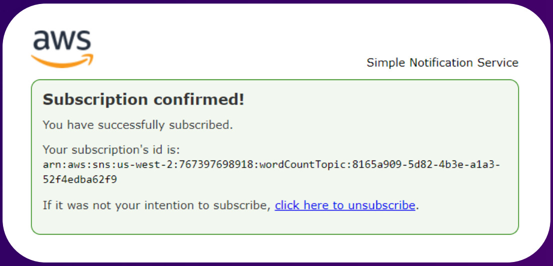
You have chosen to subscribe to the topic:
arn:aws:sns:us-west-2:767397698918:wordCountTopic

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

Step 8: Confirm Email subscription

Choose Confirm subscription. A new browser tab opens and displays a page with the message "Subscription confirmed!".



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
`arn:aws:sns:us-west-2:767397698918:wordCountTopic:8165a909-5d82-4b3e-a1a3-52f4edba62f9`

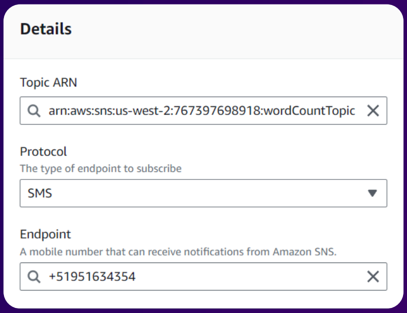
If it was not your intention to subscribe, click here to unsubscribe.

Task 2

Configuring notifications

Step 9: Create SMS subscription

In the **Subscriptions** section, select Create subscription. In the **Details** section, configure the following settings.



Details

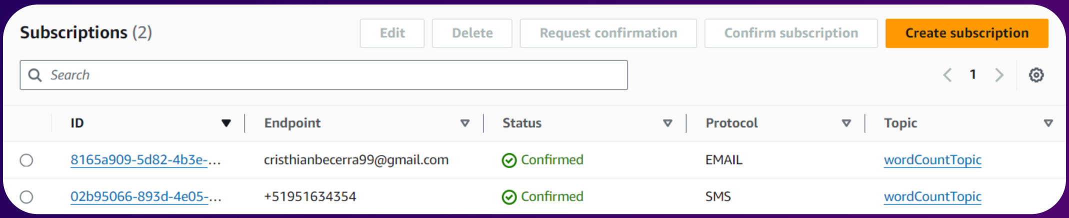
Topic ARN: arn:aws:sns:us-west-2:767397698918:wordCountTopic

Protocol
The type of endpoint to subscribe
SMS

Endpoint
A mobile number that can receive notifications from Amazon SNS.
+51951634354

Step 10: Review your Subscriptions

In the **Subscriptions** section, review your two new subscriptions.



Subscriptions (2)

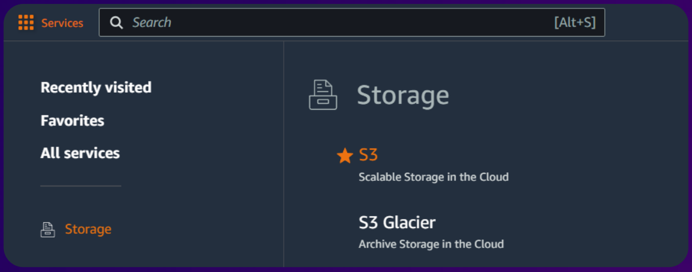
ID	Endpoint	Status	Protocol	Topic
8165a909-5d82-4b3e-...	cristhianbecerra99@gmail.com	Confirmed	EMAIL	wordCountTopic
02b95066-893d-4e05-...	+51951634354	Confirmed	SMS	wordCountTopic

Task 3

Creating the S3 bucket

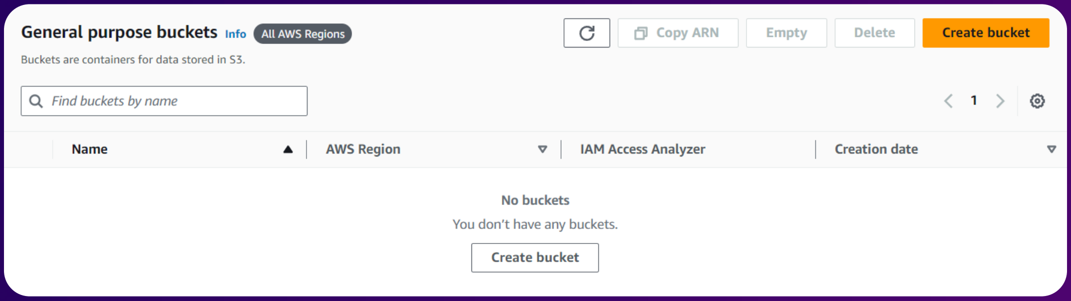
Step 1: Access the Amazon S3 Console

In the AWS Management Console, select S3.



Step 2: Create bucket

Navigate to the **Buckets** section, and select Create bucket.



Buckets are containers for data stored in S3.

No buckets

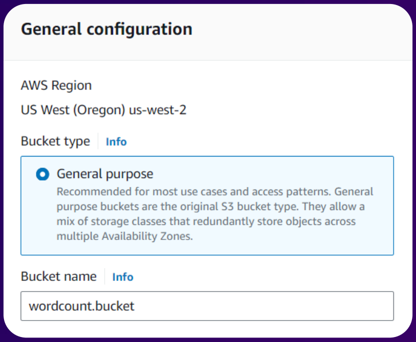
You don't have any buckets.

Task 3

Creating the S3 bucket

Step 3: General configuration

In the **General configuration** section, configure the following settings.


General configuration

AWS Region

US West (Oregon) us-west-2

Bucket type Info

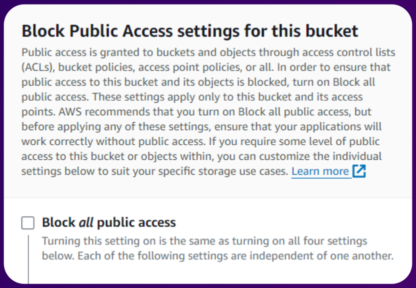
● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Bucket name Info

wordcount.bucket

Step 4: Unblock public access

In the **Block Public Access settings for this bucket** section, uncheck Block all public access.


Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

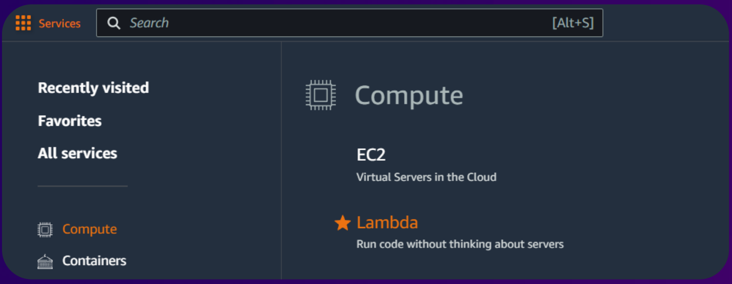
☐ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Task 4

Creating the wordCount Lambda function

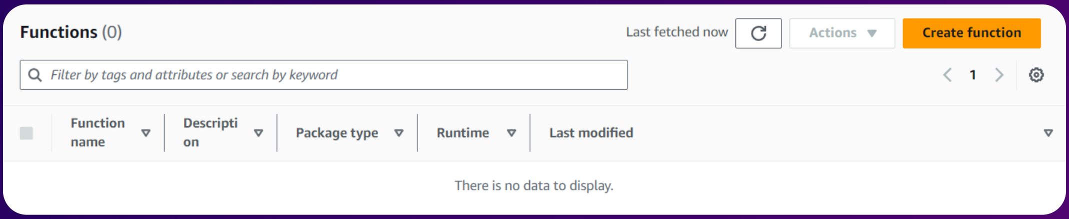
Step 1: Access the Lambda service

In the AWS Management Console, select Lambda.



Step 2: Create function

Navigate to the **Functions** section, and select Create function.



Function name	Description	Package type	Runtime	Last modified

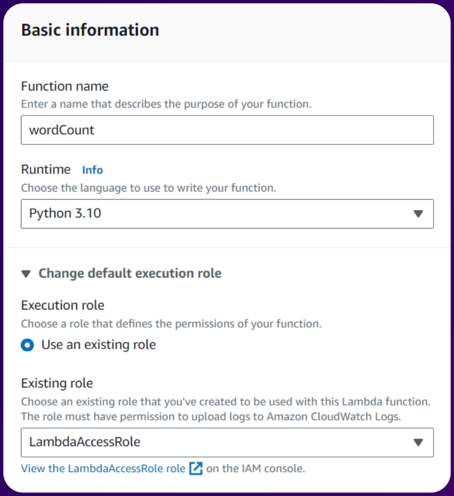
There is no data to display.

Task 4

Creating the wordCount Lambda function

Step 3: Basic information

In the **Create function** page, select Author from scratch, and configure the following settings in the **Basic information** section.



Basic information

Function name
Enter a name that describes the purpose of your function.
wordCount

Runtime Info
Choose the language to use to write your function.
Python 3.10

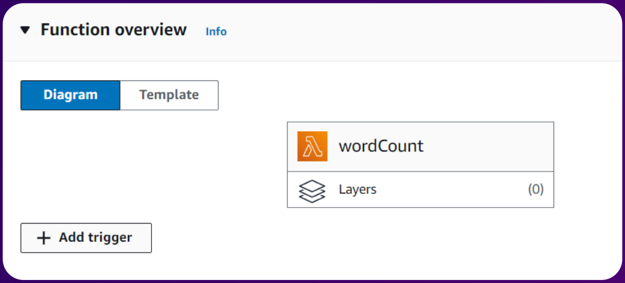
▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function.
● Use an existing role

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.
LambdaAccessRole
View the LambdaAccessRole role on the IAM console.

Step 4: Add trigger

In the **Function overview** panel, choose Add trigger.



▼ **Function overview** Info

Diagram | Template

wordCount
Layers (0)

+ Add trigger

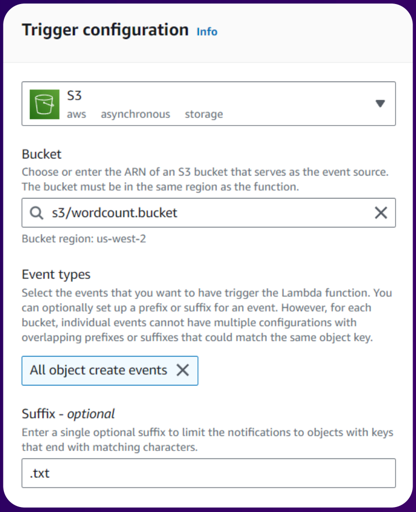
aws re/start

Task 4

Creating the wordCount Lambda function

Step 5: Trigger configuration

In the **Trigger configuration** section, configure the following settings.



Trigger configuration Info

S3
aws asynchronous storage

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

s3/wordcount.bucket

Bucket region: us-west-2

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

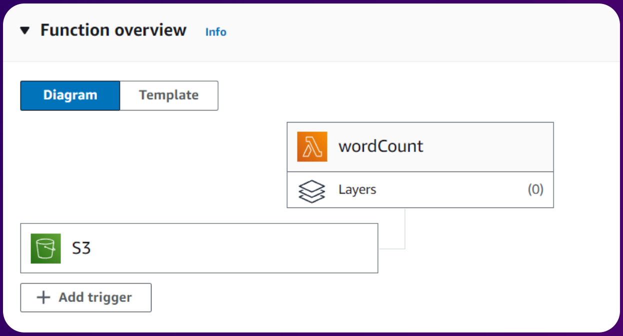
All object create events

Suffix - *optional*
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

.txt

Step 6: Review Function overview

The new trigger is created and displayed in the **Function overview** panel.

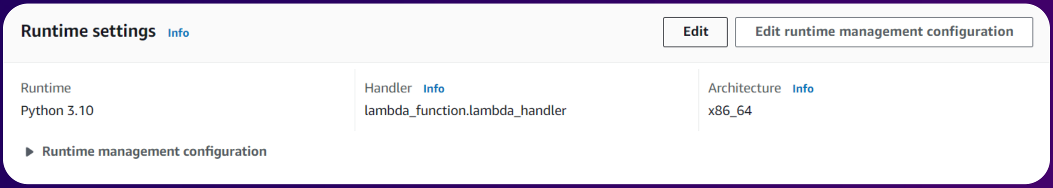


Task 4

Creating the wordCount Lambda function

Step 7: Edit Runtime settings

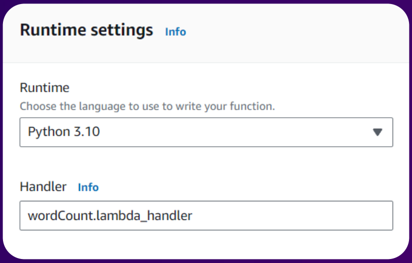
In the **Runtime settings** panel, choose Edit.


Runtime settings Info
Edit
Edit runtime management configuration
Runtime management configuration

Runtime: Python 3.10
Handler: lambda_function.lambda_handler
Architecture: x86_64

Step 8: Handler

In the **Runtime settings** section, configure the following handler.



Runtime
Choose the language to use to write your function.
Python 3.10

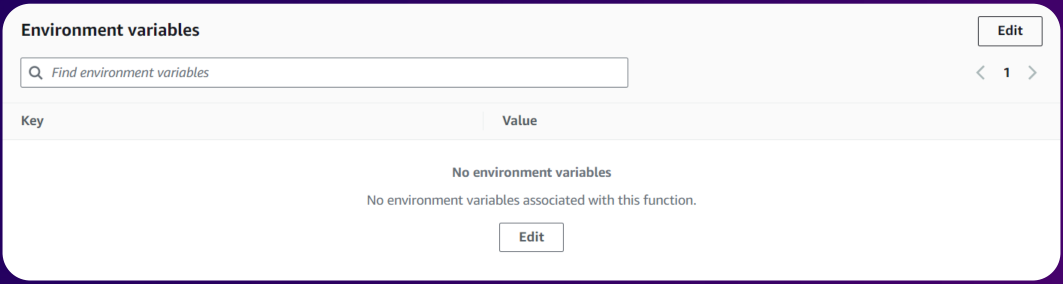
Handler: wordCount.lambda_handler

Task 4

Creating the wordCount Lambda function

Step 9: Edit Environment variables

Choose the **Configuration** tab, and choose **Environment variables**. Choose Edit.

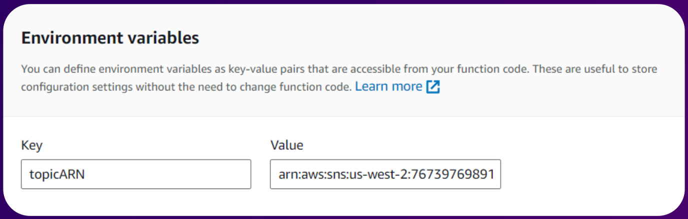


No environment variables

No environment variables associated with this function.

Step 10: Environment variables

In the **Environment variables** section, configure the following options.



You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more

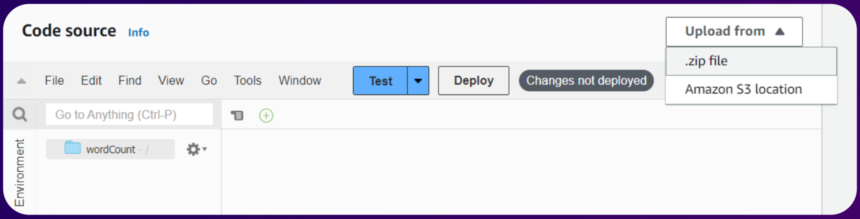
Key	Value
topicARN	arn:aws:sns:us-west-2:76739769891

Task 4

Creating the wordCount Lambda function

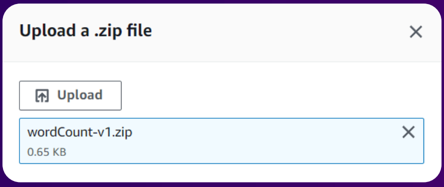
Step 11: Upload Code source

In the **Code source** panel, choose Upload from, and select .zip file.



Step 12: Upload a .zip file

Upload the .zip file containing the .py file with your Python code.



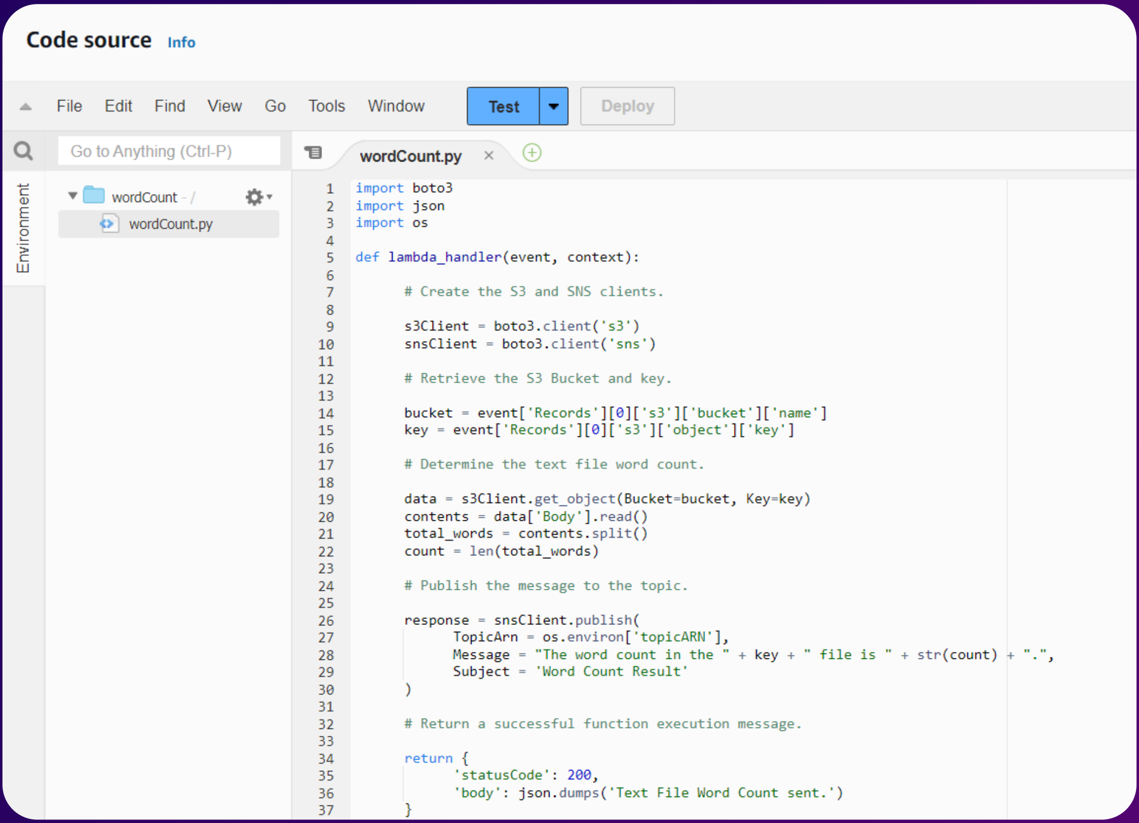
aws re/start

Task 4

Creating the wordCount Lambda function

Step 13: Review Code source

The Lambda function code is imported and displayed in the **Code source** panel. Review the Python code that implements the function. Read the comments included in the code to understand its logic flow.



```python
import boto3
import json
import os

def lambda_handler(event, context):

      # Create the S3 and SNS clients.

      s3Client = boto3.client('s3')
      snsClient = boto3.client('sns')

      # Retrieve the S3 Bucket and key.

      bucket = event['Records'][0]['s3']['bucket']['name']
      key = event['Records'][0]['s3']['object']['key']

      # Determine the text file word count.

      data = s3Client.get_object(Bucket=bucket, Key=key)
      contents = data['Body'].read()
      total_words = contents.split()
      count = len(total_words)

      # Publish the message to the topic.

      response = snsClient.publish(
            TopicArn = os.environ['topicARN'],
            Message = "The word count in the " + key + " file is " + str(count) + ".",
            Subject = 'Word Count Result'
      )

      # Return a successful function execution message.

      return {
            'statusCode': 200,
            'body': json.dumps('Text File Word Count sent.')
      }
```

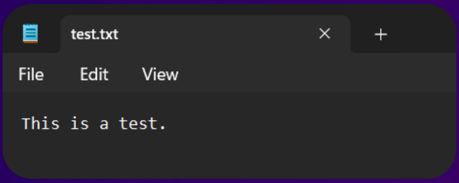
aws re/start

Task 5

Testing the wordCount Lambda function

Step 1: Create a sample .txt file

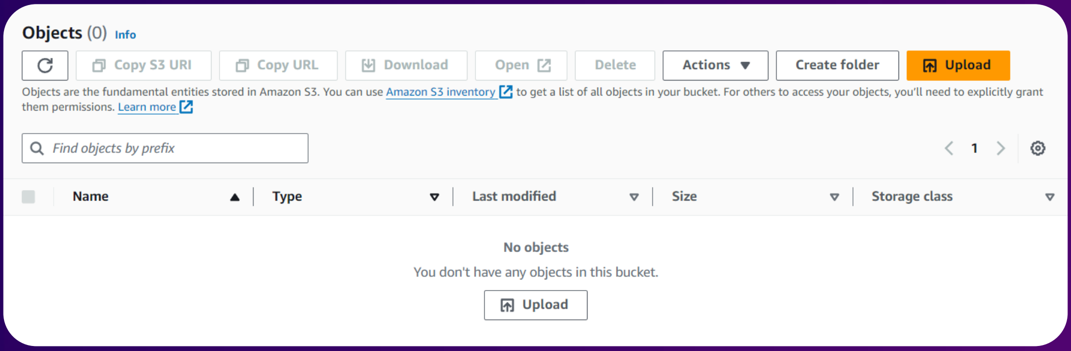
Create a sample .txt file. For this test, the sample file named test.txt contains 4 words.



```
This is a test.
```

Step 2: Upload objects

In the S3 Console, navigate to the **Buckets** section, choose the **wordcount.bucket**, and in the **Objects** panel, select Upload.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.

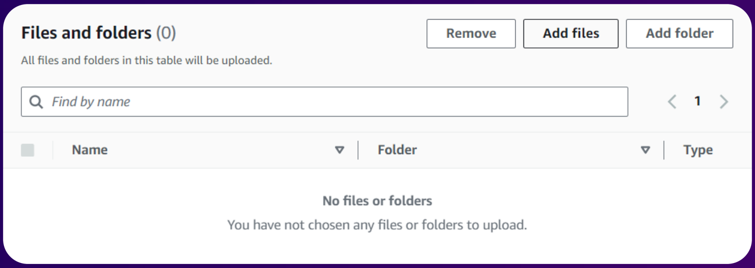
Upload

Task 5

Testing the wordCount Lambda function

Step 3: Add files to upload

In the **Files and folders** section, select Add files, and choose the test.txt file.

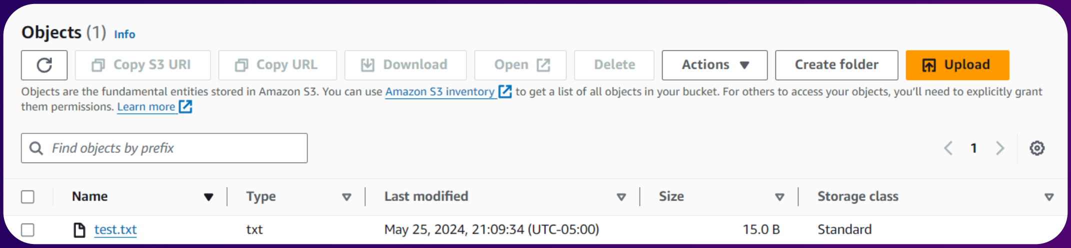


All files and folders in this table will be uploaded.

No files or folders
You have not chosen any files or folders to upload.

Step 4: Review uploaded objects

Review the uploaded text file in the **Objects** panel.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

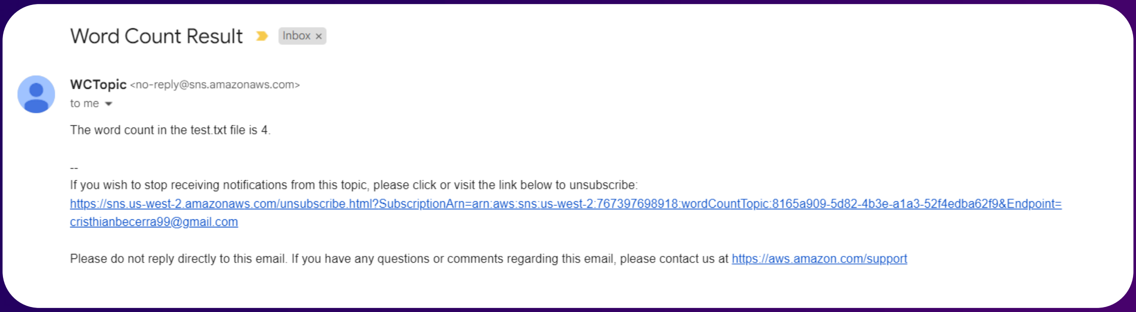
Name	Type	Last modified	Size	Storage class
test.txt	txt	May 25, 2024, 21:09:34 (UTC-05:00)	15.0 B	Standard

Task 5

Testing the wordCount Lambda function

Step 5: Check your Email inbox

If there were no errors, you should see a new email from AWS Notifications with the subject "Word Count Result."



Word Count Result Inbox ×

WCTopic <no-reply@sns.amazonaws.com>
to me

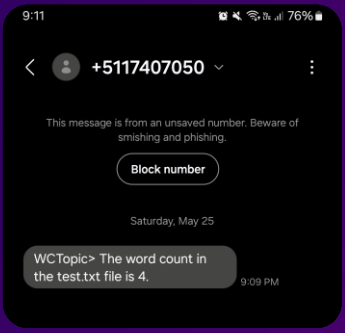
The word count in the test.txt file is 4.

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-west-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-west-2:767397698918:wordCountTopic:8165a909-5d82-4b3e-a1a3-52f4edba62f9&Endpoint=cristhianbecerra99@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

Step 6: Check your SMS messages

If there were no errors, you should see a new SMS message from AWS Notifications starting with "WCTopic>".



9:11 76%

+5117407050

This message is from an unsaved number. Beware of smishing and phishing.

Block number

Saturday, May 25

WCTopic> The word count in the test.txt file is 4. 9:09 PM

aws re/start

Conclusions

AWS Lambda
AWS Lambda allows you to run code without provisioning or managing servers, enabling scalable and cost-effective computing with automatic scaling and high availability.

Lambda Function
A Lambda function is a self-contained piece of code written in a supported language, executed in response to specific triggers, events, or conditions.

Runtime
The runtime provides the execution environment for Lambda functions, including the necessary libraries, dependencies, and runtime languages like Node.js, Python, or Java.

Execution Role
The execution role in AWS Lambda is an IAM role that grants the function permissions to interact with other AWS services, ensuring secure and controlled access to resources during function execution.

Triggers
Triggers are event sources that invoke Lambda functions, such as changes in data state in DynamoDB, updates in an S3 bucket, or messages in an SQS queue, enabling event-driven architecture.

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru