

## Step 6-7

To trigger my lambda function, I added a S3 trigger to the *frequentWord()* Lambda function, additionally, ".txt" is added to the optional suffix box in the trigger configuration, this means the Lambda function will trigger when a text file is uploaded to the S3 bucket. The *upload.txt* is uploaded to the S3 Bucket by first specifying the access key environments, and by running the *upload\_file()* function. Once these steps are done, the Lambda function will run and parse the paragraph provided in the *upload.txt* file. The result can be found in "Monitor"->"logging"->"View logs in CloudWatch".

Execution result:

The below execution result shows the result of the *word\_count* Lambda Function by passing a JSON event object as an argument. The JSON event object contains the input paragraph to be parsed.

lambda\_function ×

Environment Var ×

Execution result: ×

+

▼ Execution results

Status: SucceededMax memory used: 40 MBTime: 20.04 ms

Test Event Name	FrequentWordEvent
Response	<pre>{   "statusCode": 200,   "body": "{ 'a': 6, 'and': 6, 'to': 5, 'chatgpt': 3, 'language': 3, 'with': 3, 'knowledge': 3, 'in': 3, 'can': 2, 'of': 2, 'natural': 2, 'processing': 2, 'looking': 2, 'the': 2, 'or': 2, 'is': 1, 'cutting-e</pre>
Function Logs	<pre>START RequestId: c7e9b77c-efd9-4a21-9ce3-1924dabfa8b3 Version: \$LATEST { 'a': 6, 'and': 6, 'to': 5, 'chatgpt': 3, 'language': 3, 'with': 3, 'knowledge': 3, 'in': 3, 'can': 2, 'of': 2, 'natural': 2, 'processing': 2, 'looking': 2, 'the': 2, 'or': 2, 'is': 1, 'cutting-edge': 1, 'm END RequestId: c7e9b77c-efd9-4a21-9ce3-1924dabfa8b3 REPORT RequestId: c7e9b77c-efd9-4a21-9ce3-1924dabfa8b3 Duration: 20.04 ms Billed Duration: 21 ms Memory Size: 128 MB Max Memory Used: 40 MB Init Duration: 119.02 ms</pre>
Request ID	c7e9b77c-efd9-4a21-9ce3-1924dabfa8b3

Log events

Q Filter events - press enter to search

Clear1m30m1h12hCustomUTC timezoneDisplay

ActionsStart tailingCreate metric filter

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

▶	Timestamp	Message
No older events at this moment. <a href="#">Retry</a>		
▶	2024-09-22T01:45:56.749Z	INIT_START Runtime Version: python:3.8.v53 Runtime Version ARN: arn:aws:lambda:ca-central-1::runtime:124ac49f727c31fa5ba440a749eef8274ca2beb4a0a8f107a0d8c69632c715a7
▶	2024-09-22T01:45:57.024Z	START RequestId: 57362c79-10bb-4967-a228-901b0ff1daa1 Version: \$LATEST
▶	2024-09-22T01:45:59.499Z	Content of the file upload.txt from bucket word-count-bucket1:
▼	2024-09-22T01:45:59.499Z	<div><div>{'a': 6, 'and': 6, 'to': 5, 'chatgpt': 3, 'language': 3, 'with': 3, 'knowledge': 3, 'in': 3, 'can': 2, 'of': 2, 'natural': 2, 'processing': 2, 'looking': 2, 'the': 2, 'or': 2, 'is': 1, ...</div><div>{'a': 6, 'and': 6, 'to': 5, 'chatgpt': 3, 'language': 3, 'with': 3, 'knowledge': 3, 'in': 3, 'can': 2, 'of': 2, 'natural': 2, 'processing': 2, 'looking': 2, 'the': 2, 'or': 2, 'is': 1, 'cutting-edge': 1, 'model': 1, 'that': 1, 'perform': 1, 'wide': 1, 'range': 1, 'tasks': 1, 'remarkable': 1, 'accuracy': 1, 'fluency': 1, 'from': 1, 'answering': 1, 'general': 1, 'questions': 1, 'assisting': 1, 'research': 1, 'generating': 1, 'human-like': 1, 'text': 1, 'even': 1, 'engaging': 1, 'creative': 1, 'writing': 1, "chatgpt's": 1, 'capabilities': 1, 'are': 1, 'truly': 1, 'impressive': 1, 'its': 1, 'vast': 1, 'base': 1, 'ability': 1, 'learn': 1, 'adapt': 1, 'over': 1, 'time': 1, 'serve': 1, 'as': 1, 'valuable': 1, 'resource': 1, 'for': 1, 'anyone': 1, 'harness': 1, 'power': 1, 'their': 1, 'work': 1, 'personal': 1, 'projects': 1, 'whether': 1, "you're": 1, 'researcher': 1, 'writer': 1, 'business': 1, 'professional': 1, 'simply': 1, 'someone': 1, 'engage': 1, 'interesting': 1, 'conversations': 1, 'has': 1, 'skills': 1, 'help': 1, 'you': 1, 'achieve': 1, 'your': 1, 'goals': 1}</div></div>
▶	2024-09-22T01:45:59.526Z	END RequestId: 57362c79-10bb-4967-a228-901b0ff1daa1
▶	2024-09-22T01:45:59.526Z	REPORT RequestId: 57362c79-10bb-4967-a228-901b0ff1daa1 Duration: 2501.90 ms Billed Duration: 2502 ms Memory Size: 128 MB Max Memory Used: 77 MB Init Duration: 273.56 ms
No newer events at this moment. <a href="#">Auto retry paused</a> . <a href="#">Resume</a>		

Step 8

API Resources Page:

API Gateway > APIs > Resources - My API (xoqxc542i)

Resources

API actions ▼

Deploy API

Create resource

/

/test  
POST

Resource details

Update documentation

Enable CORS

Path

/

Resource ID

hlhd6iunnd

Methods (0)

Delete

Create method

Method type ▲

Integration type ▼

Authorization ▼

API key ▼

No methods

No methods defined.

API Stages Page:

[API Gateway](#) > [APIs](#) > [My API \(xoqxc542i\)](#) > Stages

Stages

Stage actions ▼

Create stage

stage\_1

Stage details

Info

Edit

Stage name	Rate	Web ACL
stage_1	-	-
Cache cluster	Burst	Client certificate
<div>Inactive</div>	-	-
Default method-level caching		
<div>Inactive</div>		

Invoke URL

https://xoqxc542i.execute-api.ca-central-1.amazonaws.com/stage\_1

Active deployment

e6eorh on September 21, 2024, 19:13 (UTC-07:00)

Logs and tracing

Info

Edit

CloudWatch logs	Detailed metrics	Data tracing
<div>Inactive</div>	<div>Inactive</div>	<div>Inactive</div>
X-Ray tracing		
<div>Inactive</div>		

stage\_1

e6e0rh on September 21, 2024, 19:13 (UTC-07:00)

### Logs and tracing [Info](#)

Edit

CloudWatch logs

⊖ Inactive

Detailed metrics

⊖ Inactive

Data tracing

⊖ Inactive

X-Ray tracing

⊖ Inactive

Custom access logging

⊖ Inactive

Stage variables

Deployment history

Documentation history

Canary

Tags

### Stage variables (0/0)

Edit

🔍 Find resources

< 1 >

Name



Value



No variables

No variables associated with the stage.

Manage variables

## Serverless Function:

The screenshot displays the AWS Lambda console interface for a function named **WordCountRestAPI**. The function is configured with **API Gateway** as the trigger (indicated by a count of 2) and has **0** layers attached. A button labeled **+ Add trigger** is visible. On the right, the function's status is shown as **Last modified 24 minutes ago**, with the **Function ARN** being `arn:aws:lambda:ca-central-1:277707110479:function:WordCountRestAPI` and a **Function URL** link.

The **Code source** tab is active, showing the Python code for the `lambda_handler` function. The code processes a paragraph by removing punctuation, splitting it into words, counting word occurrences using `collections.Counter`, and returning a JSON response with the sorted word counts.

```
1 import json
2 from collections import Counter
3
4 def lambda_handler(event, context):
5
6     def word_count(paragraph):
7         # Remove punctuation from input paragraph and convert to lowercase
8         paragraph = paragraph.lower().replace('.', '').replace(',', '')
9
10        # Split the paragraph into a list of words
11        words = paragraph.split()
12
13        # Count the number of occurrences of each word
14        word_count = Counter(words)
15
16        # Sort the words in descending order of frequency
17        sorted_word_count = dict(sorted(word_count.items(), key=lambda item: item[1], reverse=True))
18        print(sorted_word_count)
19        return sorted_word_count
20
21    paragraph = event['body']['paragraph']
22
23    sorted_word_count = word_count(paragraph)
24
25
26    return {
27        "statusCode": 200,
28        "body": str(sorted_word_count)
29    }
```

## Execution result:

CloudWatch > Log groups > /aws/lambda/WordCountRestAPI > 2024/09/22/[\$LATEST]f108fffb3ac74bd7af117355cce65e91

**Log events** 🔄 Actions ▼ Start tailing Create metric filter

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Clear 1m 30m 1h 12h Custom UTC timezone ▼ Display ▼ ⚙️

▶	Timestamp	Message
		No older events at this moment. <a href="#">Retry</a>
▶	2024-09-22T02:13:54.257Z	INIT_START Runtime Version: python:3.8.v53 Runtime Version ARN: arn:aws:lambda:ca-central-1::runtime:124ac49f727c31fa5ba440a749eef8274ca2beb4a0a8f107a0d8c69632c715a7
▶	2024-09-22T02:13:54.374Z	START RequestId: 9fd9f9f9-f81f-4842-9740-930ef38e4edd Version: \$LATEST
▼	2024-09-22T02:13:54.375Z	{ 'i': 18, 'you': 16, 'and': 13, 'the': 13, 'oh': 11, 'be': 10, 'to': 10, 'a': 9, 'were': 7, 'me': 7, 'romeo': 6, 'said': 6, 'love': 6, 'we': 5, 'my': 5, 'say': 5, 'it's': 5, 'see': 4, '...  { 'i': 18, 'you': 16, 'and': 13, 'the': 13, 'oh': 11, 'be': 10, 'to': 10, 'a': 9, 'were': 7, 'me': 7, 'romeo': 6, 'said': 6, 'love': 6, 'we': 5, 'my': 5, 'say': 5, 'it's': 5, 'see': 4, 'was': 4, 'don't': 4, 'alone': 4, 'i'll': 4, 'waiting': 4, 'is': 4, 'story': 4, 'baby': 4, 'just': 4, 'yes': 4, 'out': 4, 'this': 4, 'when': 3, 'on': 3, 'in': 3, 'your': 3, 'know': 3, 'juliet': 3, 'go': 3, 'all': 3, 'you'll': 3, 'so': 3, 'cause': 3, 'but': 3, 'of': 3, 'both': 2, 'young': 2, 'first': 2, 'saw': 2, 'close': 2, 'eyes': 2, 'make': 2, 'little': 2, 'daddy': 2, 'stay': 2, 'away': 2, 'from': 2, 'beggin': 2, 'please': 2, 'take': 2, 'somewhere': 2, 'can': 2, 'there's': 2, 'left': 2, 'do': 2, 'run': 2, 'prince': 2, 'princess': 2, 'keep': 2, 'if': 2, 'town': 2, 'for': 2, 'save': 2, 'never': 2, 'flashback': 1, 'starts': 1, 'i'm': 1, 'standin': 1, 'there': 1, 'balcony': 1, 'summer': 1, 'air': 1, 'lights': 1, 'party': 1, 'ball': 1, 'gowns': 1, 'way': 1, 'through': 1, 'crowd': 1, 'hello': 1, 'did': 1, 'that': 1, 'throwin': 1, 'pebbles': 1, 'cryin': 1, 'staircase': 1, 'sneak': 1, 'garden': 1, 'quiet': 1, 'we're': 1, 'dead': 1, 'they': 1, 'knew': 1, 'escape': 1, 'while': 1, 'scarlet': 1, 'letter': 1, 'everything': 1, 'they're': 1, 'tryna': 1, 'tell': 1, 'how': 1, 'feel': 1, 'difficult': 1, 'real': 1, 'afraid': 1, 'we'll': 1, 'it': 1, 'mess': 1, 'got': 1, 'tired': 1, 'wonderin': 1, 'ever': 1, 'comin': 1, 'around': 1, 'faith': 1, 'fading': 1, 'met': 1, 'outskirts': 1, 'i've': 1, 'been': 1, 'feeling': 1, 'come': 1, 'head?': 1, 'what': 1, 'think': 1, 'he': 1, 'kneelt': 1, 'ground': 1, 'pulled': 1, 'ring': 1, 'marry': 1, 'have': 1, 'that's': 1, 'really': 1, 'talked': 1, 'dad': 1, 'pick': 1, 'white': 1, 'dress': 1}
▶	2024-09-22T02:13:54.376Z	END RequestId: 9fd9f9f9-f81f-4842-9740-930ef38e4edd
▶	2024-09-22T02:13:54.376Z	REPORT RequestId: 9fd9f9f9-f81f-4842-9740-930ef38e4edd Duration: 2.23 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 39 MB Init Duration: 116.26 ms
		No newer events at this moment. <a href="#">Auto retry paused.</a> <a href="#">Resume</a>

URL:

invoke\_url\_and\_resource\_path = "[https://xoqxcn542i.execute-api.ca-central-1.amazonaws.com/stage\\_1/test](https://xoqxcn542i.execute-api.ca-central-1.amazonaws.com/stage_1/test)"

## Step 9

Similar to Step 7, I created a S3 bucket to trigger my *goat\_latin()* Lambda Function. The trigger code is the *upload\_file()* provided in the assignment, with some minor changes to the environment variables and bucket name. Once *upload\_file()* is ran, the *upload.txt* file containing a sentence will be uploaded to the S3 bucket, thereby triggering the *goat\_latin()* Lambda function, which will parse the sentence provided in the *upload.txt* file into goat latin.

Serverless Function (*goat\_latin*):

Goat Latin rules:

You are given a string sentence that consist of words separated by spaces. Each word consists of lowercase and uppercase letters only.

We would like to convert the sentence to "Goat Latin" (a made-up language similar to Pig Latin.) The rules of Goat Latin are as follows:

If a word begins with a vowel ('a', 'e', 'i', 'o', or 'u'), append "ma" to the end of the word.

- For example, the word "apple" becomes "applema".

If a word begins with a consonant (i.e., not a vowel), remove the first letter and append it to the end, then add "ma".

- For example, the word "goat" becomes "oatgma".

Add one letter 'a' to the end of each word per its word index in the sentence, starting with 1.

- For example, the first word gets "a" added to the end, the second word gets "aa" added to the end, and so on.



## Code source [Info](#)

File Edit Find View Go Tools Window

Test

Deploy

Changes not deployed



Go to Anything (Ctrl-P)



lambda\_function x

Environment Var x

Execution results x



Environment

goatLatin - /

lambda\_function.py

```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     def goat_latin(sentence):
6         res = ""
7         vowel="aeiouAEIOU"
8         for idx, word in enumerate(sentence.split()):
9             if word[0] not in vowel:
10                 word = word[1:] + word[0]
11                 word += "ma" + "a"*(idx+1)
12                 res += word + " "
13
14         res = res.strip()
15         print(res)
16         return res
17
18     s3 = boto3.client('s3')
19
20     # Get bucket name and file key from the S3 event
21     bucket_name = event['Records'][0]['s3']['bucket']['name']
22     file_key = event['Records'][0]['s3']['object']['key']
23
24     # Get the file object from S3
25     file_obj = s3.get_object(Bucket=bucket_name, Key=file_key)
26
27     # Read the content of the file
28     file_content = file_obj['Body'].read().decode('utf-8')
29
30     print(f'Content of the file {file_key} from bucket {bucket_name}:')
31
32     res = goat_latin(file_content)
33
34     return {
35         'statusCode': 200,
36         'body': str(res)
37     }
```

Execution result

ToolsWindow

TestDeploy

lambda\_function x

Environment Var x

Execution result x

+

Execution results

Status: SucceededMax memory used: 40 MBTime: 2.00 ms

A function update is still in progress so the invocation went to the previously deployed code and configuration.

Test Event Name

(unsaved) test event

Response

{  
 "statusCode": 200,  
 "body": "heTmaa uickqmaa rownbmaaaa oxfmaaaa umpedjmaaaaa overmaaaaaa hetmaaaaaaa ayzlmaaaaaaaa ogdmaaaaaaaa"  
}

Function Logs

START RequestId: 26e07bca-a884-4c97-86f4-5de57ab26dca Version: \$LATEST  
END RequestId: 26e07bca-a884-4c97-86f4-5de57ab26dca  
REPORT RequestId: 26e07bca-a884-4c97-86f4-5de57ab26dca Duration: 2.00 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 40 MB Init Duration: 113.03 ms

Request ID

26e07bca-a884-4c97-86f4-5de57ab26dca

CloudWatch Log:

CloudWatch > Log groups > /aws/lambda/goatLatin > 2024/09/24/[\$LATEST]1805dedccd9244acacc29285ef3152bc

Log events

ActionsStart tailingCreate metric filter

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Filter events - press enter to search

Clear1m30m1h12hCustomUTC timezoneDisplay

Timestamp	Message
	No older events at this moment. <a href="#">Retry</a>
2024-09-24T00:40:05.199Z	INIT_START Runtime Version: python:3.8.v53 Runtime Version ARN: arn:aws:lambda:ca-central-1::runtime:124ac49f727c31fa5ba440a749eef8274ca2beb4a0a8f107a0d8c69632c715a7
2024-09-24T00:40:05.487Z	START RequestId: b27bbdce-2738-457f-8cc9-43ad5907172b Version: \$LATEST
2024-09-24T00:40:07.931Z	Content of the file upload.txt from bucket goatlatinbucket:
2024-09-24T00:40:07.931Z	heTmaa uickqmaa rownbmaaaa oxfmaaaa umpedjmaaaaa overmaaaaaa hetmaaaaaaa ayzlmaaaaaaaa ogdmaaaaaaaa
2024-09-24T00:40:07.978Z	END RequestId: b27bbdce-2738-457f-8cc9-43ad5907172b
2024-09-24T00:40:07.978Z	REPORT RequestId: b27bbdce-2738-457f-8cc9-43ad5907172b Duration: 2490.68 ms Billed Duration: 2491 ms Memory Size: 128 MB Max Memory Used: 77 MB Init Duration: 286.80 ms
	No newer events at this moment. Auto retry paused. <a href="#">Resume</a>

## Trigger Event:

```
import logging
import boto3
from botocore.exceptions import ClientError
import os
from dotenv import load_dotenv

load_dotenv(".env")

for key, value in os.environ.items():
    print(f"{key}: {value}")

def upload_file(file_name, bucket, object_name=None):
    # Verify env file is setup correctly
    print("AWS Access Key:", os.getenv("AWS_ACCESS_KEY_ID"))

    """Upload a file to an S3 bucket

    :param file_name: File to upload
    :param bucket: Bucket to upload to
    :param object_name: S3 object name. If not specified then file_name is used
    :return: True if file was uploaded, else False
    """

    # If S3 object_name was not specified, use file_name
    if object_name is None:
        object_name = os.path.basename(file_name)
    # Upload the file
    s3_client = boto3.client('s3', aws_access_key_id=os.getenv("AWS_ACCESS_KEY_ID"), aws_secret_access_key=os.getenv("AWS_SECRET_ACCESS_KEY"), aws_session_token=os.getenv("AWS_SESSION_TOKEN"))
    try:
        response = s3_client.upload_file(file_name, bucket, object_name)
        print(response)
    except ClientError as e:
        logging.error(e)
        return False
    return True

upload_file('upload.txt', "goatlatinbucket", 'upload.txt')
```

# goatlatinbucket Info

- Objects
- Properties
- Permissions
- Metrics
- Management
- Access Points

Objects (1) Info

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

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](#)

Find objects by prefix

< 1 >

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	<a href="#">upload.txt</a>	txt	September 23, 2024, 17:40:05 (UTC-07:00)	44.0 B	Standard