Assignment 3

Part A

Meet Patel (B00899516)

Dalhousie University

Subject

CSCI 5410 (Serverless Data Processing)

Professor

Dr. Saurabh Dey

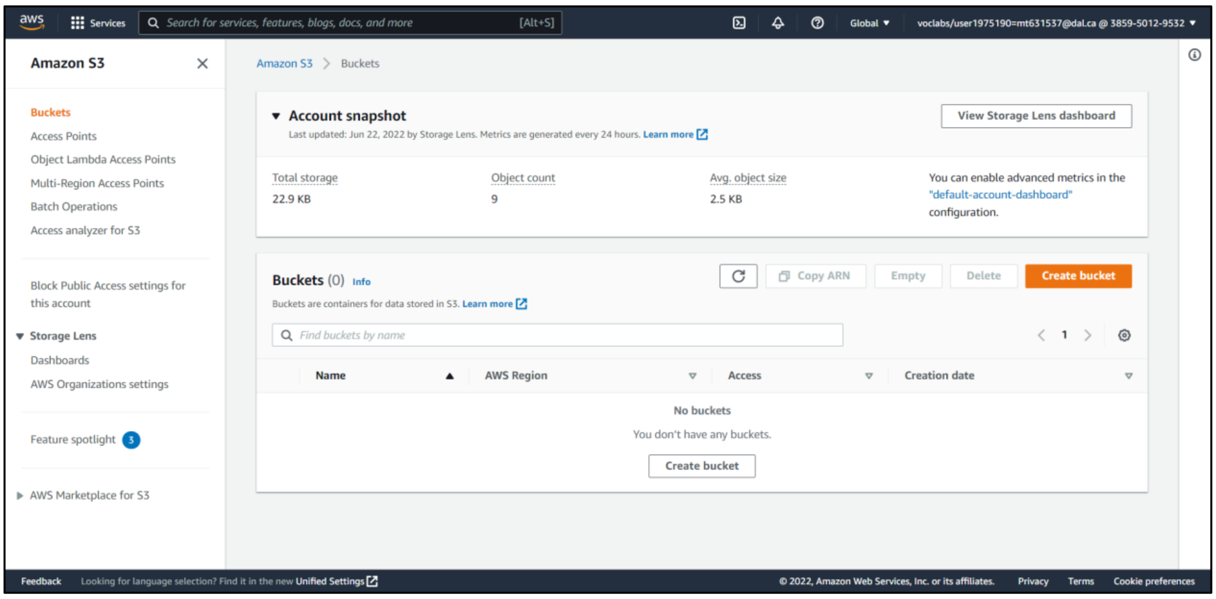
Project Git Repository

Gitlab Repository Link: https://git.cs.dal.ca/patel13/csci5410_b00899516_meet_patel.git

Serverless Application (Using Java and AWS Lambda)

A. Create your 1st S3 bucket **SourceB00xxxxxx** and 2nd S3 bucket **TagsB00xxxxxx** using AWS SDK (any programming language)

Figure 1 is responsible for representing an empty Amazon S3 dashboard. Initially there is no s3 bucket.


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global
voclabs/user1975190=mt631537@dal.ca @ 3859-5012-9532
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > Buckets
View Storage Lens dashboard
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Account snapshot

Last updated: Jun 22, 2022 by Storage Lens. Metrics are generated every 24 hours. Learn more

Total storage	Object count	Avg. object size	
22.9 KB	9	2.5 KB	You can enable advanced metrics in the "default-account-dashboard" configuration.

Buckets (0) Info

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date

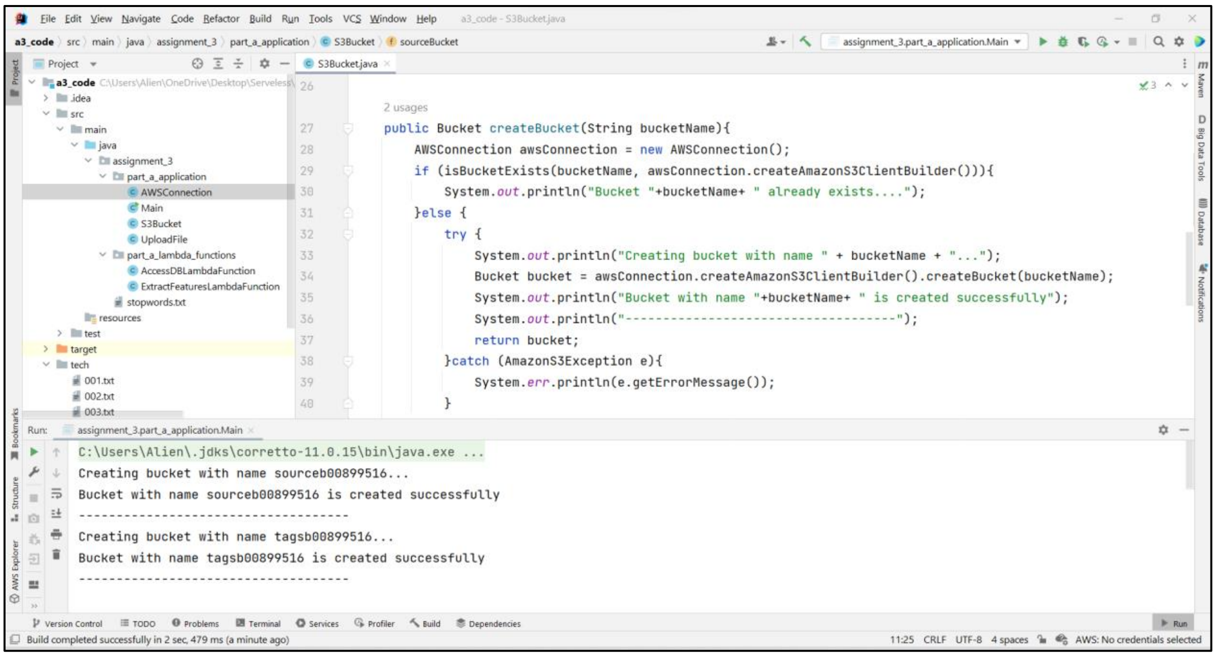
No buckets

You don't have any buckets.

Create bucket

Figure 1: Empty Amazon S3 dashboard

Figure 2 is responsible for displaying the two s3 bucket namely "sourceb00899516" and "tagsb00899516" which are created using AWS java SDK on IntelliJ IDE.

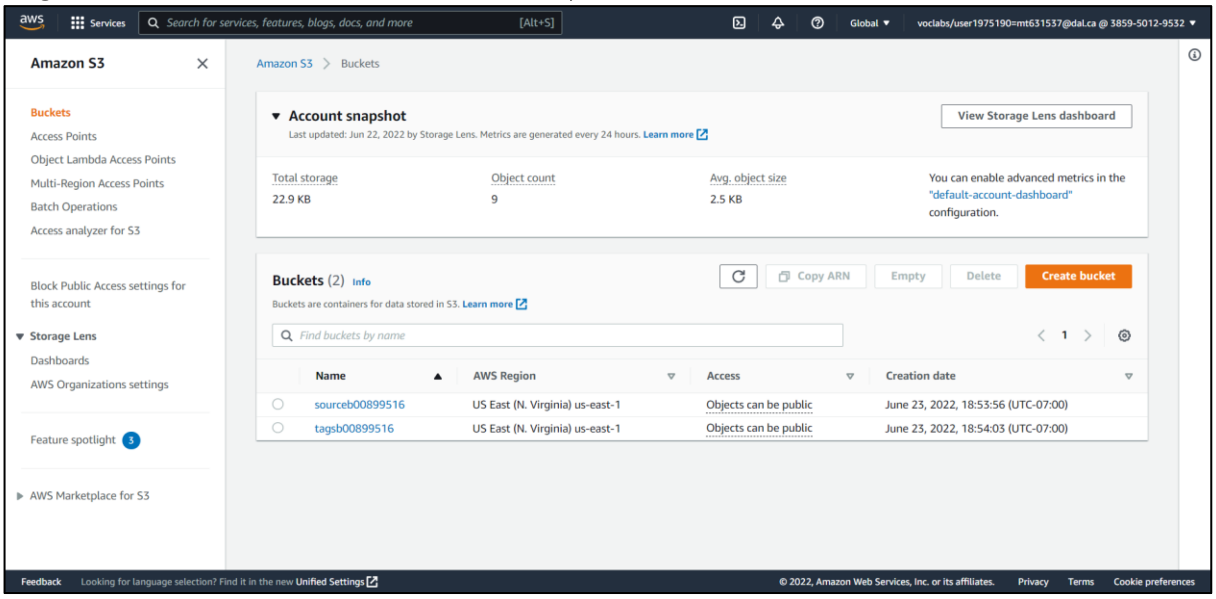


```java
2 usages
public Bucket createBucket(String bucketName){
    AWSConnection awsConnection = new AWSConnection();
    if (isBucketExists(bucketName, awsConnection.createAmazonS3ClientBuilder())){
        System.out.println("Bucket "+bucketName+ " already exists....");
    }else {
        try {
            System.out.println("Creating bucket with name " + bucketName + "...");
            Bucket bucket = awsConnection.createAmazonS3ClientBuilder().createBucket(bucketName);
            System.out.println("Bucket with name "+bucketName+ " is created successfully");
            System.out.println("------------------------------------");
            return bucket;
        }catch (AmazonS3Exception e){
            System.err.println(e.getErrorMessage());
        }
```

```
C:\Users\Alien\.jdks\corretto-11.0.15\bin\java.exe ...
Creating bucket with name sourceb00899516...
Bucket with name sourceb00899516 is created successfully
------------------------------------
Creating bucket with name tagsb00899516...
Bucket with name tagsb00899516 is created successfully
------------------------------------
```

Figure 2: Creation of two Amazon S3 Bucket namely "sourceb00899516" and "tagsb00899516" using java and AWS java SDK

Figure 3 is responsible for displaying the two Amazon S3 bucket namely "**sourceb00899516**" and "**tagsb00899516**" which are created successfully created on Amazon S3 console.



Account snapshot

Last updated: Jun 22, 2022 by Storage Lens. Metrics are generated every 24 hours. Learn more

Total storage	Object count	Avg. object size
22.9 KB	9	2.5 KB

You can enable advanced metrics in the "default-account-dashboard" configuration.

Buckets (2)

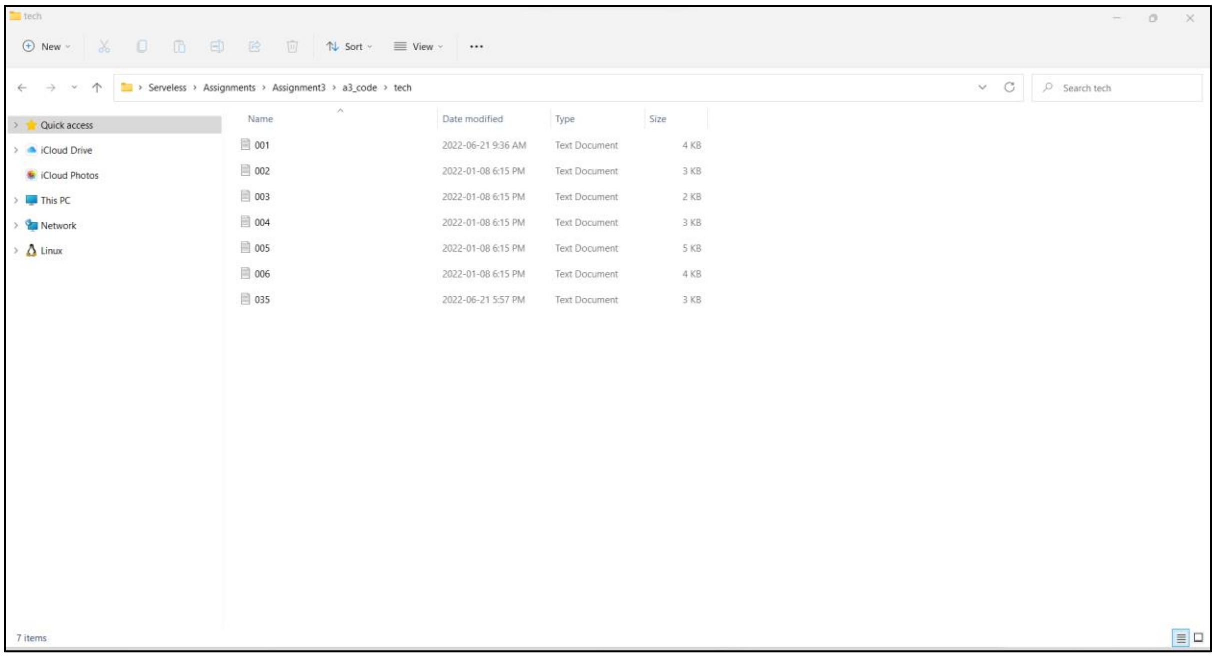
Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
sourceb00899516	US East (N. Virginia) us-east-1	Objects can be public	June 23, 2022, 18:53:56 (UTC-07:00)
tagsb00899516	US East (N. Virginia) us-east-1	Objects can be public	June 23, 2022, 18:54:03 (UTC-07:00)

Figure 3: Successful creation of two bucket namely "sourceb00899516" and "tagsb00899516" Amazon S3 console

B. Upload the files given in the **Tech folder** one at a time with a delay of **200 milliseconds** on the **1st bucket**. You need to write a script using SDK to upload the files one at a time to the S3 bucket

Figure 4 is responsible for showing the files in tech folder which need to be uploaded on "**sourceb00899516**" S3 bucket.


Serveless › Assignments › Assignment3 › a3_code › tech

Name	Date modified	Type	Size
001	2022-06-21 9:36 AM	Text Document	4 KB
002	2022-01-08 6:15 PM	Text Document	3 KB
003	2022-01-08 6:15 PM	Text Document	2 KB
004	2022-01-08 6:15 PM	Text Document	3 KB
005	2022-01-08 6:15 PM	Text Document	5 KB
006	2022-01-08 6:15 PM	Text Document	4 KB
035	2022-06-21 5:57 PM	Text Document	3 KB

7 items

Figure 4: Files in "tech" folder which are being uploaded on "sourceb00899516" Amazon S3 bucket

Figure 5 shows that all the files from the tech older has been uploaded to the "**sourceb00899516**" Amazon S3 bucket using java and AWS java SDK.

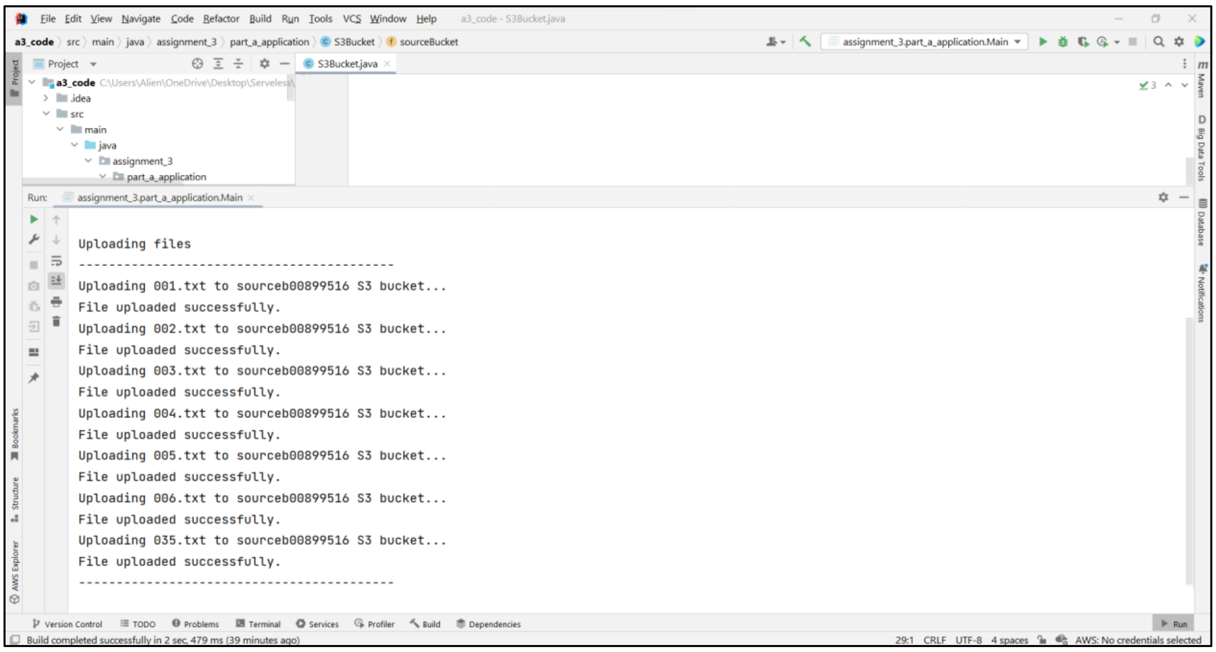
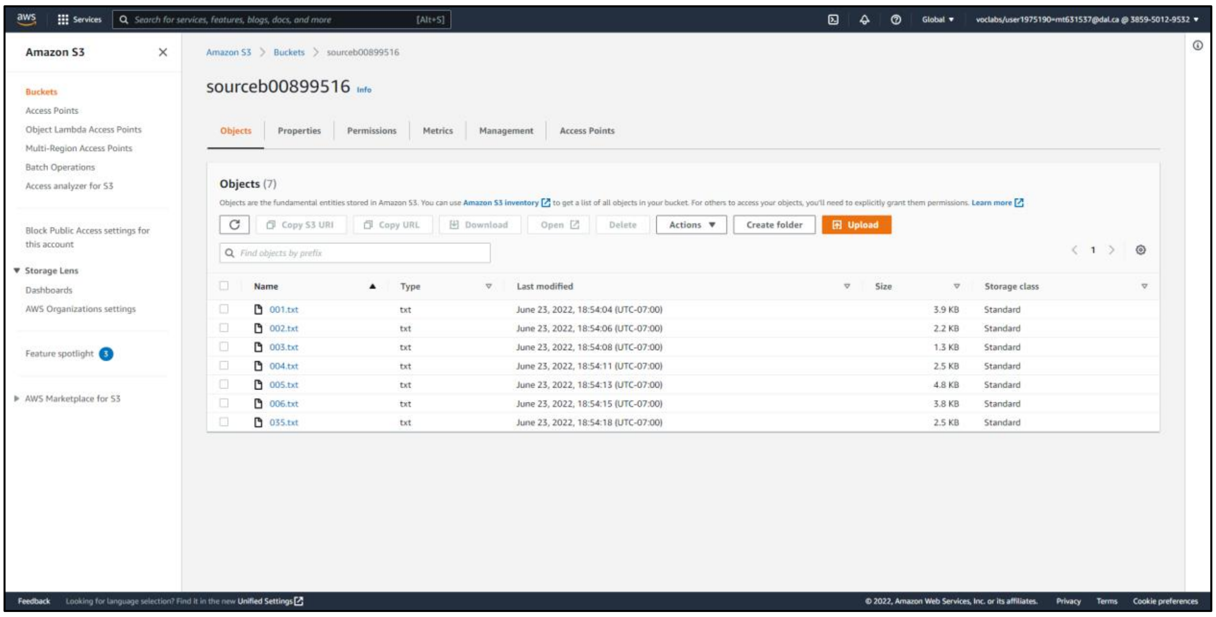

```
Uploading files
------------------------------------------
Uploading 001.txt to sourceb00899516 S3 bucket...
File uploaded successfully.
Uploading 002.txt to sourceb00899516 S3 bucket...
File uploaded successfully.
Uploading 003.txt to sourceb00899516 S3 bucket...
File uploaded successfully.
Uploading 004.txt to sourceb00899516 S3 bucket...
File uploaded successfully.
Uploading 005.txt to sourceb00899516 S3 bucket...
File uploaded successfully.
Uploading 006.txt to sourceb00899516 S3 bucket...
File uploaded successfully.
Uploading 035.txt to sourceb00899516 S3 bucket...
File uploaded successfully.
------------------------------------------
```
Build completed successfully in 2 sec, 479 ms (39 minutes ago)

Figure 5: All the files from the tech older has been uploaded to the "sourceb00899516" Amazon S3 bucket

Figure 6 shows the "**sourceb00899516**" Amazon S3 bucket console where all the files from the tech folder have been uploaded successfully.



sourceb00899516

Objects (7)

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

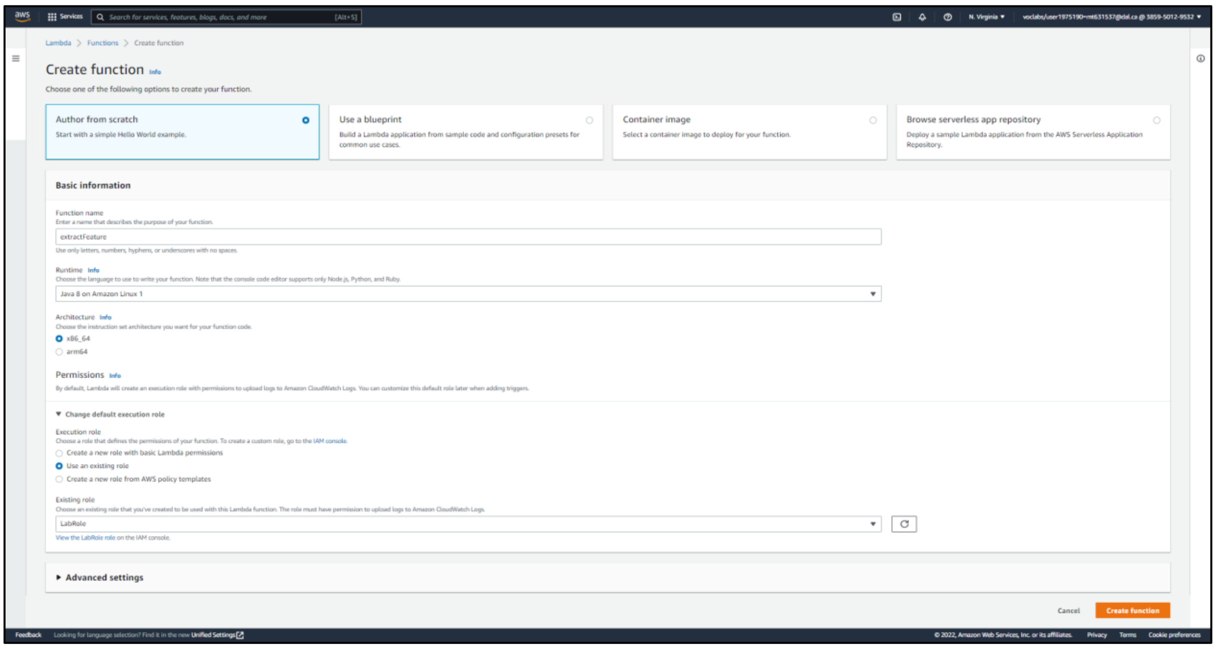
Name	Type	Last modified	Size	Storage class
001.txt	txt	June 23, 2022, 18:54:04 (UTC-07:00)	3.9 KB	Standard
002.txt	txt	June 23, 2022, 18:54:06 (UTC-07:00)	2.2 KB	Standard
003.txt	txt	June 23, 2022, 18:54:08 (UTC-07:00)	1.3 KB	Standard
004.txt	txt	June 23, 2022, 18:54:11 (UTC-07:00)	2.5 KB	Standard
005.txt	txt	June 23, 2022, 18:54:13 (UTC-07:00)	4.8 KB	Standard
006.txt	txt	June 23, 2022, 18:54:15 (UTC-07:00)	3.8 KB	Standard
035.txt	txt	June 23, 2022, 18:54:18 (UTC-07:00)	2.5 KB	Standard

*Figure 6: "**sourceb00899516**" Amazon S3 bucket console where all the files from the tech folder have been uploaded successfully*

C. If a file is available on the 1st bucket, then it triggers **extractFeatures** Lambda function, which is the 1st lambda function.

Creation Of Lambda Function

Figure 7 is responsible for the showing the creation "**extractFeatures**" lambda function and its configuration


Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1975190=mt631537@dal.ca @ 3859-5012-9532
Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

[✓ Author from scratch] Start with a simple Hello World example.

Use a blueprint: Build a Lambda application from sample code and configuration presets for common use cases.

Container image: Select a container image to deploy for your function.

Browse serverless app repository: Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name: Enter a name that describes the purpose of your function.
extractFeature
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info: Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Java 8 on Amazon Linux 1

Architecture Info: Choose the instruction set architecture you want for your function code.
[✓ x86_64] arm64

Permissions Info: By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role: Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
Create a new role with basic Lambda permissions
[✓ Use an existing role]
Create a new role from AWS policy templates

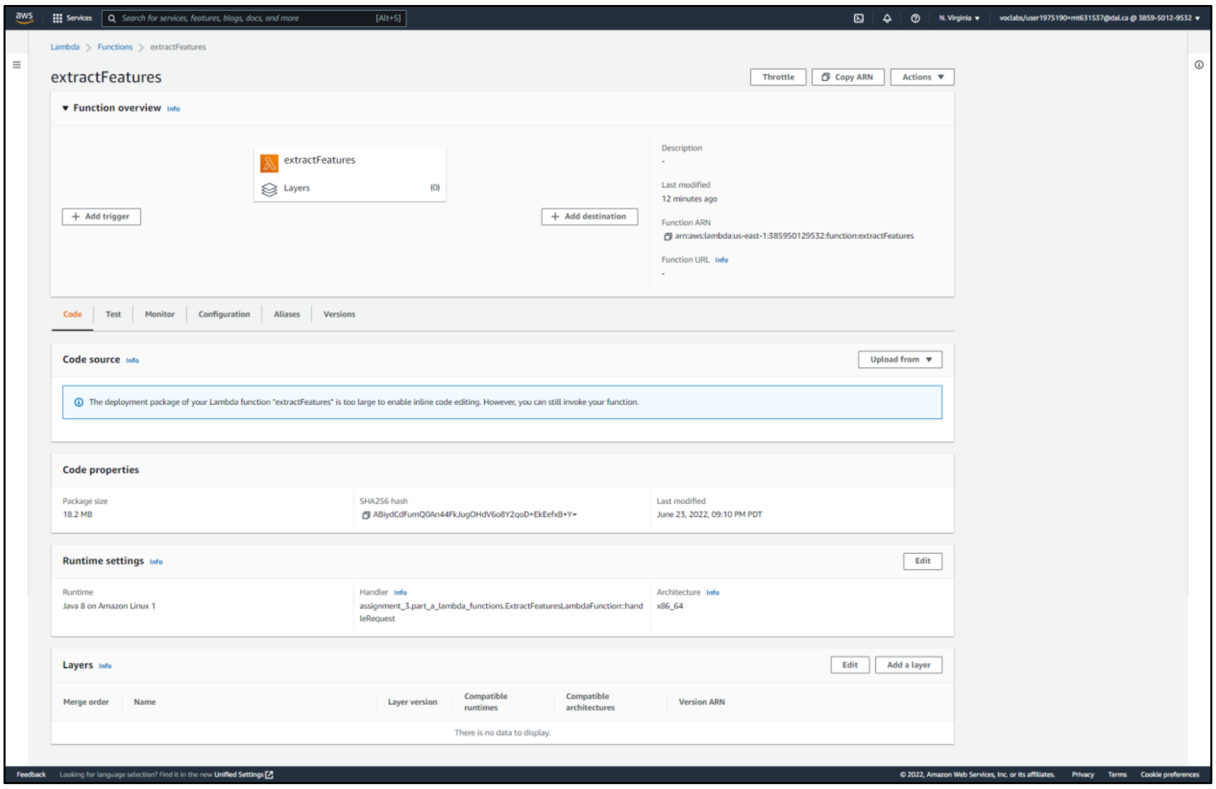
Existing role: Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
LabRole
View the LabRole role on the IAM console.

▶ Advanced settings

Cancel
Create function
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

*Figure 7: Creation of "**extractFeatures**" lambda Function*

Figure 8 shows the successfully created lambda function on Amazon Lambda named as "**extractFeatures**"



Lambda > Functions > extractFeatures

extractFeatures

Function overview

Description
-

Last modified
12 minutes ago

Function ARN
arn:aws:lambda:us-east-1:385950129532:function:extractFeatures

Function URL
-

Code source

The deployment package of your Lambda function "extractFeatures" is too large to enable inline code editing. However, you can still invoke your function.

Code properties

Package size	SHA256 hash	Last modified
18.2 MB	ABiydCdFumQ0An44FkJugOHdV6o8Y2qoD+EkEefxB+Y=	June 23, 2022, 09:10 PM PDT

Runtime settings

Runtime	Handler	Architecture
Java 8 on Amazon Linux 1	assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction::handleRequest	x86_64

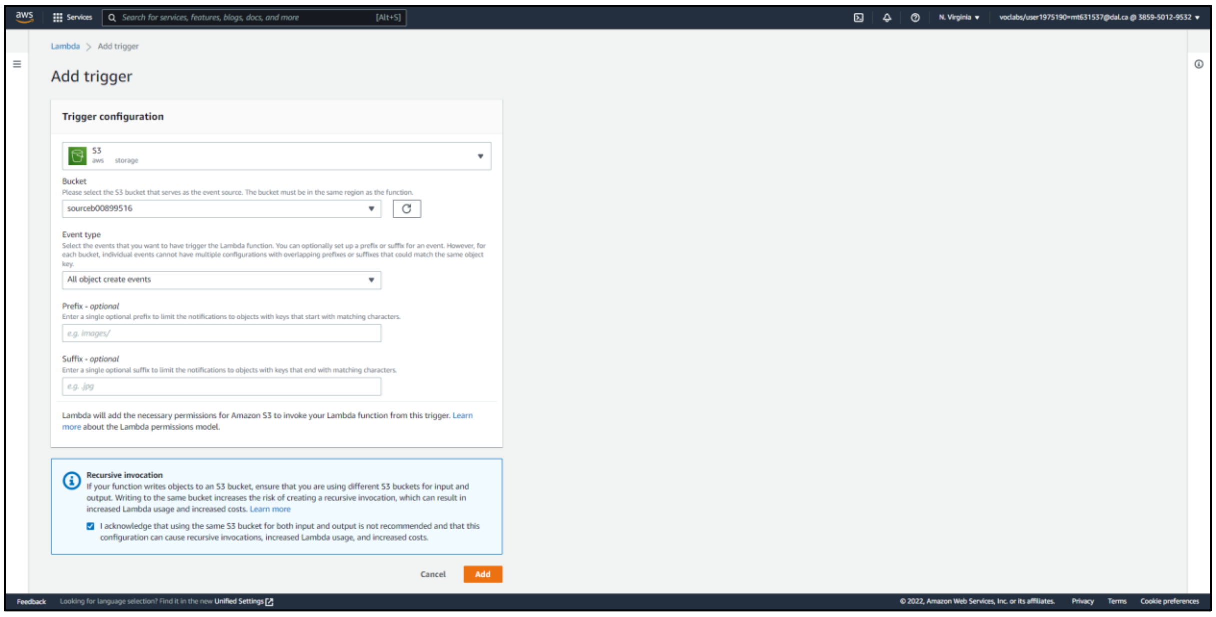
Layers

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN

There is no data to display.

*Figure 8: Successfully created lambda Function "**extractFeatures**"*

Figure 9 shows the creation of trigger for "**sourceb0899516**" S3 bucket, and it will be triggered when object is uploaded to the "**sourceb00899516**"



Lambda > Add trigger

Add trigger

Trigger configuration

S3
aws storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.
sourceb00899516

Event type
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

Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/

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

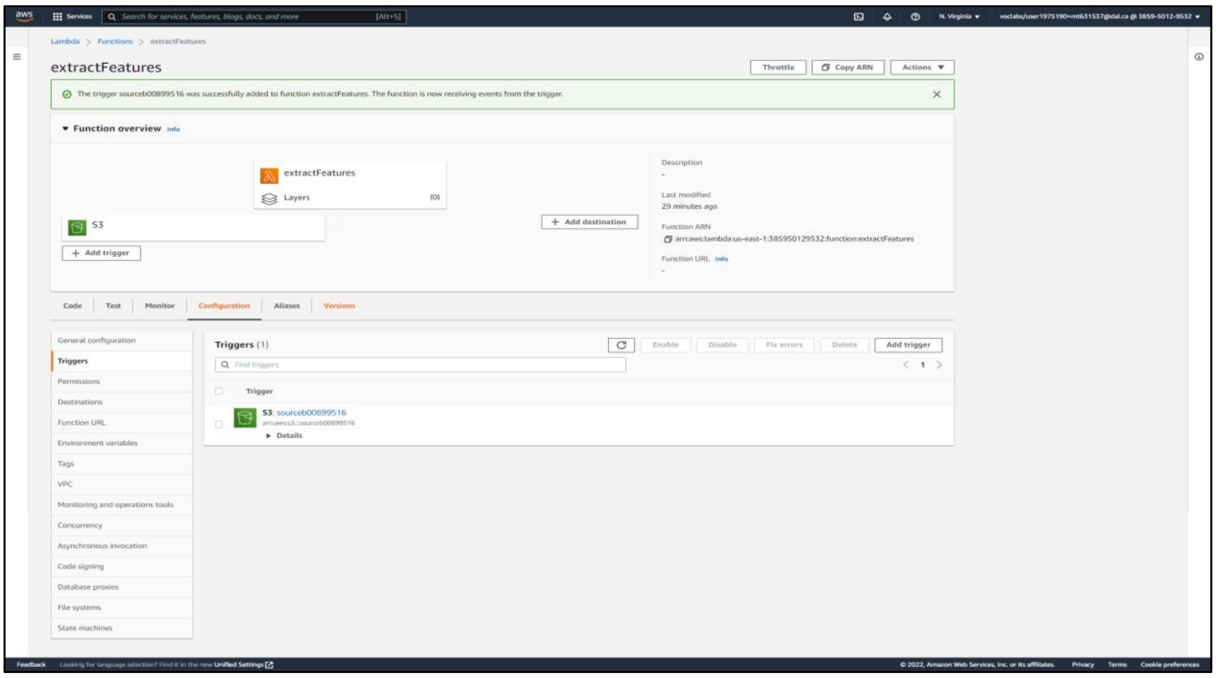
Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more

✓ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Figure 9: Creation of trigger for "sourceb0899516" S3 bucket

Figure 10 shows the successful creation of trigger for "**sourceb00899516**" S3 bucket.



extractFeatures

The trigger sourceb00899516 was successfully added to function extractFeatures. The function is now receiving events from the trigger.

Description
-

Last modified
29 minutes ago

Function ARN
arn:aws:lambda:us-east-1:385950129532:function:extractFeatures

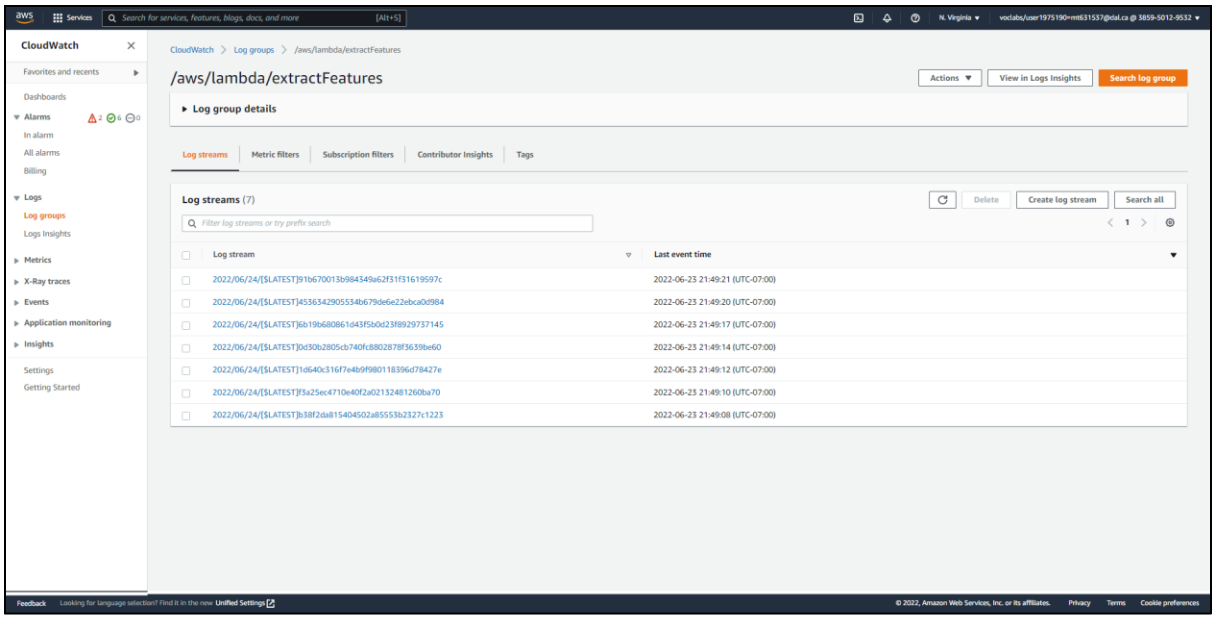
Function URL
-

Triggers (1)

Trigger
S3: sourceb00899516 arn:aws:s3:::sourceb00899516 Details

Figure 10: Successful creation of trigger for "sourceb00899516" S3 bucket

Figure 11 is shows that when object is uploaded to the "**sourceb00899516**" Amazon S3 bucket. Lambda function "**extractFeatures**" will trigger. The below figure shows the logs from CloudWatch.



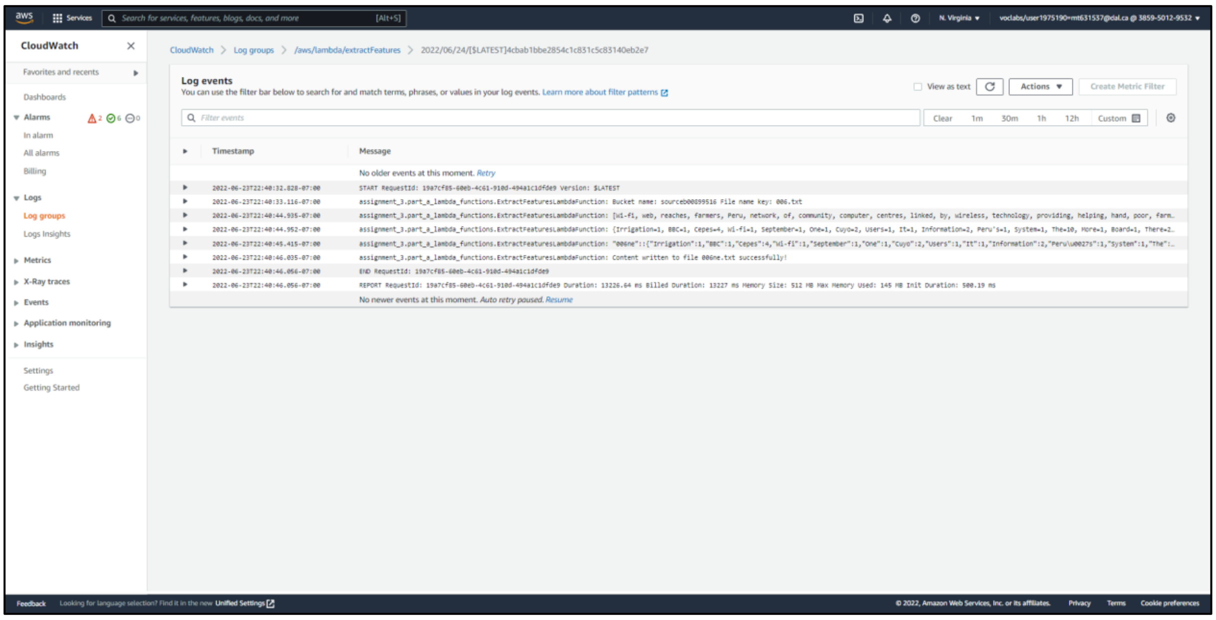
/aws/lambda/extractFeatures

Log streams (7)

Log stream	Last event time
2022/06/24/[$LATEST]91b670013b984349a62f31f31619597c	2022-06-23 21:49:21 (UTC-07:00)
2022/06/24/[$LATEST]4536342905534b679de6e22ebca0d984	2022-06-23 21:49:20 (UTC-07:00)
2022/06/24/[$LATEST]6b19b680861d43f5b0d23f8929737145	2022-06-23 21:49:17 (UTC-07:00)
2022/06/24/[$LATEST]0d30b2805cb740fc8802878f3639be60	2022-06-23 21:49:14 (UTC-07:00)
2022/06/24/[$LATEST]1d640c316f7e4b9f980118396d78427e	2022-06-23 21:49:12 (UTC-07:00)
2022/06/24/[$LATEST]f3a25ec4710e40f2a02132481260ba70	2022-06-23 21:49:10 (UTC-07:00)
2022/06/24/[$LATEST]b38f2da815404502a85553b2327c1223	2022-06-23 21:49:08 (UTC-07:00)

Figure 11: Logs of Triggered "extractFeatures" lambda function

Figure 12 shows logs in detail.

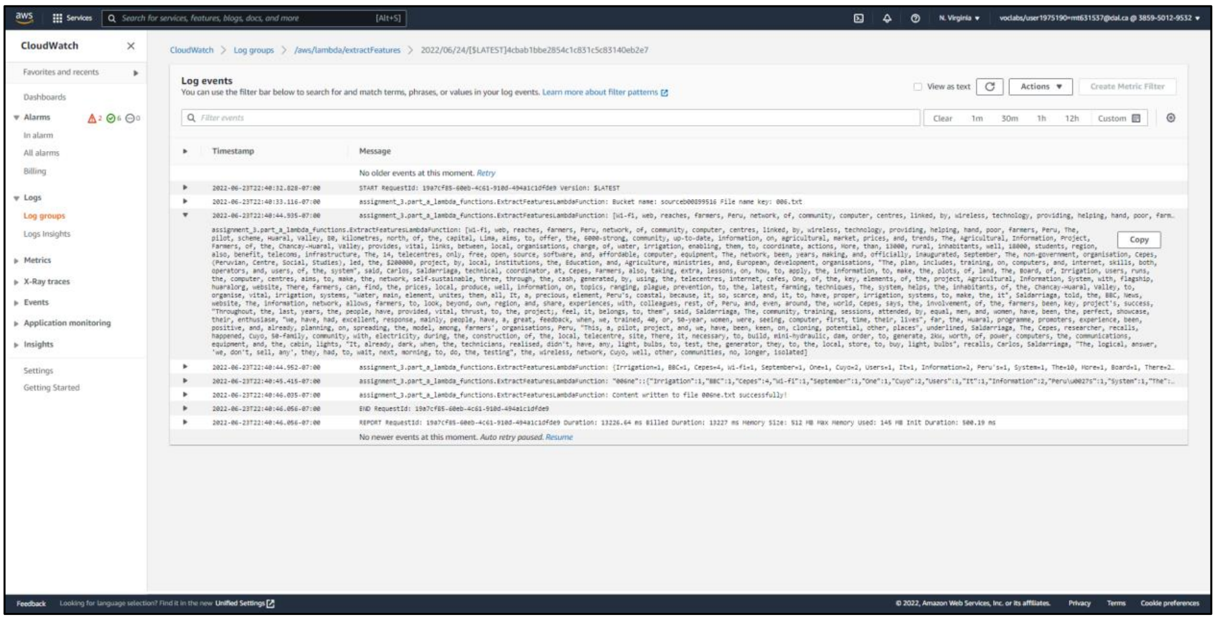

CloudWatch > Log groups > /aws/lambda/extractFeatures > 2022/06/24/[$LATEST]4cbab1bbe2854c1c831c5c83140eb2e7
Log events
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
View as text
Actions
Create Metric Filter
Filter events
Clear
1m
30m
1h
12h
Custom

```
Timestamp                        Message
                                 No older events at this moment. Retry
2022-06-23T22:40:32.828-07:00    START RequestId: 19a7cf85-60eb-4c61-910d-494a1c1dfde9 Version: $LATEST
2022-06-23T22:40:33.116-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: Bucket name: sourceb00899516 File name key: 006.txt
2022-06-23T22:40:44.935-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: [Wi-fi, web, reaches, farmers, Peru, network, of, community, computer, centres, linked, by, wireless, technology, providing, helping, hand, poor, farm…
2022-06-23T22:40:44.952-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: {Irrigation=1, BBC=1, Cepes=4, Wi-fi=1, September=1, One=1, Cuyo=2, Users=1, It=1, Information=2, Peru's=1, System=1, The=10, More=1, Board=1, There=2…
2022-06-23T22:40:45.415-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: "006ne"::{"Irrigation":1,"BBC":1,"Cepes":4,"Wi-fi":1,"September":1,"One":1,"Cuyo":2,"Users":1,"It":1,"Information":2,"Peru\u0027s":1,"System":1,"The":…
2022-06-23T22:40:46.035-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: Content written to file 006ne.txt successfully!
2022-06-23T22:40:46.056-07:00    END RequestId: 19a7cf85-60eb-4c61-910d-494a1c1dfde9
2022-06-23T22:40:46.056-07:00    REPORT RequestId: 19a7cf85-60eb-4c61-910d-494a1c1dfde9 Duration: 13226.64 ms Billed Duration: 13227 ms Memory Size: 512 MB Max Memory Used: 145 MB Init Duration: 500.19 ms
                                 No newer events at this moment. Auto retry paused. Resume
```

Figure 12: Logs in details

D. This lambda function extracts the Named entities from the file and creates a JSON array of named entities* for that file.

Figure 13 is responsible for showing the extracted list of entities from the the file content which are uploaded on “sourceb00899516”

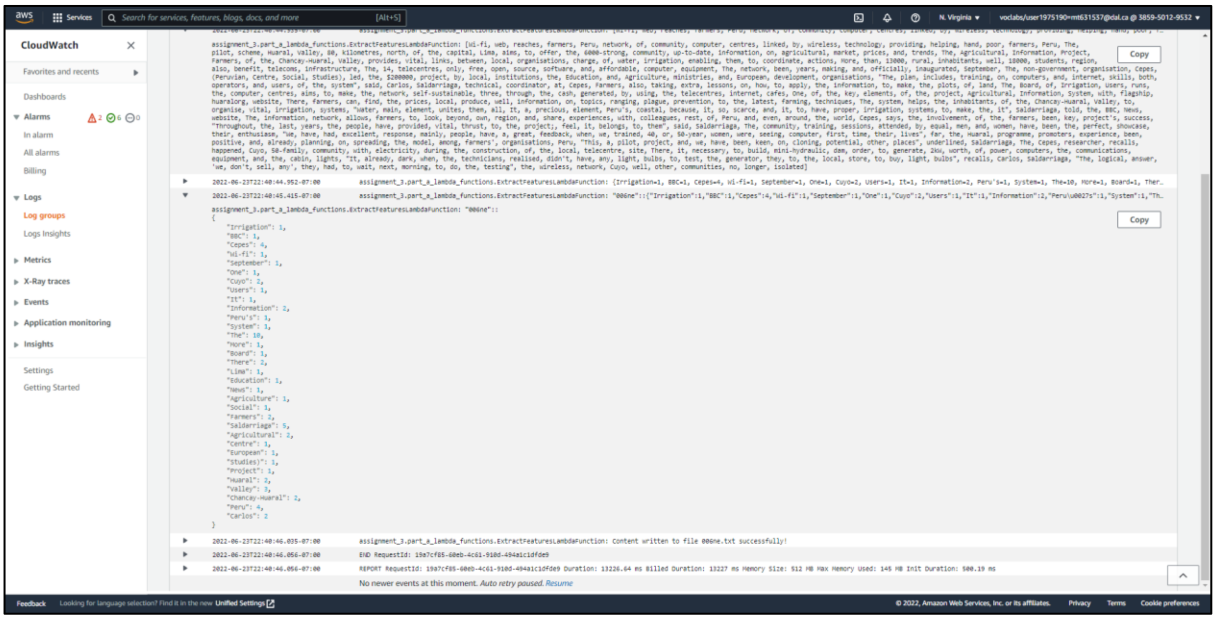

Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1975190=mt631537@dal.ca @ 3859-5012-9532
CloudWatch
Favorites and recents
Dashboards
Alarms
In alarm
All alarms
Billing
Logs
Log groups
Logs Insights
Metrics
X-Ray traces
Events
Application monitoring
Insights
Settings
Getting Started
CloudWatch > Log groups > /aws/lambda/extractFeatures > 2022/06/24/[$LATEST]4cbab1bbe2854c1c831c5c83140eb2e7
View as text
Actions
Create Metric Filter
Filter events
Clear 1m 30m 1h 12h Custom
Copy
Feedback Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Log events
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. Retry
2022-06-23T22:40:32.828-07:00    START RequestId: 19a7cf85-60eb-4c61-910d-494a1c1dfde9 Version: $LATEST
2022-06-23T22:40:33.116-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: Bucket name: sourceb00899516 File name key: 006.txt
2022-06-23T22:40:44.935-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: [Wi-fi, web, reaches, farmers, Peru, network, of, community, computer, centres, linked, by, wireless, technology, providing, helping, hand, poor, farm…
assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: [Wi-fi, web, reaches, farmers, Peru, network, of, community, computer, centres, linked, by, wireless, technology, providing, helping, hand, poor, farmers, Peru, The,
pilot, scheme, Huaral, Valley, 80, kilometres, north, of, the, capital, Lima, aims, to, offer, the, 6000-strong, community, up-to-date, information, on, agricultural, market, prices, and, trends, The, Agricultural, Information, Project,
Farmers, of, the, Chancay-Huaral, Valley, provides, vital, links, between, local, organisations, charge, of, water, irrigation, enabling, them, to, coordinate, actions, More, than, 13000, rural, inhabitants, well, 18000, students, region,
also, benefit, telecoms, infrastructure, The, 14, telecentres, only, free, open, source, software, and, affordable, computer, equipment, The, network, been, years, making, and, officially, inaugurated, September, The, non-government, organisation, Cepes,
(Peruvian, Centre, Social, Studies), led, the, $200000, project, by, local, institutions, the, Education, and, Agriculture, ministries, and, European, development, organisations, "The, plan, includes, training, on, computers, and, internet, skills, both,
operators, and, users, of, the, system", said, Carlos, Saldarriaga, technical, coordinator, at, Cepes, Farmers, also, taking, extra, lessons, on, how, to, apply, the, information, to, make, the, plots, of, land, The, Board, of, Irrigation, Users, runs,
the, computer, centres, aims, to, make, the, network, self-sustainable, three, through, the, cash, generated, by, using, the, telecentres, internet, cafes, One, of, the, key, elements, of, the, project, Agricultural, Information, System, with, flagship,
huaralorg, website, There, farmers, can, find, the, prices, local, produce, well, information, on, topics, ranging, plague, prevention, to, the, latest, farming, techniques, The, system, helps, the, inhabitants, of, the, Chancay-Huaral, Valley, to,
organise, vital, irrigation, systems, "Water, main, element, unites, them, all, It, a, precious, element, Peru's, coastal, because, it, so, scarce, and, it, to, have, proper, irrigation, systems, to, make, the, it", Saldarriaga, told, the, BBC, News,
website, The, information, network, allows, farmers, to, look, beyond, own, region, and, share, experiences, with, colleagues, rest, of, Peru, and, even, around, the, world, Cepes, says, the, involvement, of, the, farmers, been, key, project's, success,
"Throughout, the, last, years, the, people, have, provided, vital, thrust, to, the, project;, feel, it, belongs, to, them", said, Saldarriaga, The, community, training, sessions, attended, by, equal, men, and, women, have, been, the, perfect, showcase,
their, enthusiasm, "We, have, had, excellent, response, mainly, people, have, a, great, feedback, when, we, trained, 40, or, 50-year, women, were, seeing, computer, first, time, their, lives", far, the, Huaral, programme, promoters, experience, been,
positive, and, already, planning, on, spreading, the, model, among, farmers', organisations, Peru, "This, a, pilot, project, and, we, have, been, keen, on, cloning, potential, other, places", underlined, Saldarriaga, The, Cepes, researcher, recalls,
happened, Cuyo, 50-family, community, with, electricity, during, the, construction, of, the, local, telecentre, site, There, it, necessary, to, build, mini-hydraulic, dam, order, to, generate, 2kW, worth, of, power, computers, the, communications,
equipment, and, the, cabin, lights, "It, already, dark, when, the, technicians, realised, didn't, have, any, light, bulbs, to, test, the, generator, they, to, the, local, store, to, buy, light, bulbs", recalls, Carlos, Saldarriaga, "The, logical, answer,
'we, don't, sell, any', they, had, to, wait, next, morning, to, do, the, testing", the, wireless, network, Cuyo, well, other, communities, no, longer, isolated]
2022-06-23T22:40:44.952-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: {Irrigation=1, BBC=1, Cepes=4, Wi-fi=1, September=1, One=1, Cuyo=2, Users=1, It=1, Information=2, Peru's=1, System=1, The=10, More=1, Board=1, There=2…
2022-06-23T22:40:45.415-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: "006ne"::{"Irrigation":1,"BBC":1,"Cepes":4,"Wi-fi":1,"September":1,"One":1,"Cuyo":2,"Users":1,"It":1,"Information":2,"Peru\u0027s":1,"System":1,"The":…
2022-06-23T22:40:46.035-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: Content written to file 006ne.txt successfully!
2022-06-23T22:40:46.056-07:00    END RequestId: 19a7cf85-60eb-4c61-910d-494a1c1dfde9
2022-06-23T22:40:46.056-07:00    REPORT RequestId: 19a7cf85-60eb-4c61-910d-494a1c1dfde9 Duration: 13226.64 ms Billed Duration: 13227 ms Memory Size: 512 MB Max Memory Used: 145 MB Init Duration: 500.19 ms
                                 No newer events at this moment. Auto retry paused. Resume
```

Figure 13: Entities list extracted from file corrtent

Figure 14 is responsible for displaying the json array for named entities on cloud watch. The named entities were filtered out from the list of entities. Apart from that the stop words were also removed.

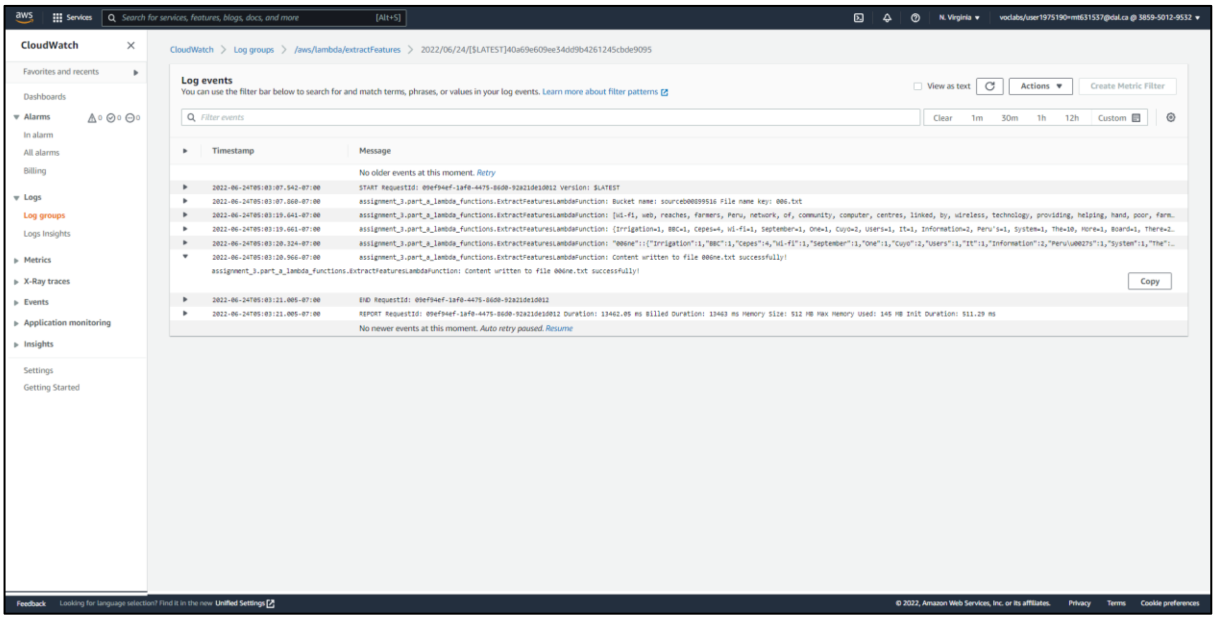

Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1975190=mt631537@dal.ca @ 3859-5012-9532
CloudWatch
Favorites and recents
Dashboards
Alarms
In alarm
All alarms
Billing
Logs
Log groups
Logs Insights
Metrics
X-Ray traces
Events
Application monitoring
Insights
Settings
Getting Started
Copy
Copy
Feedback Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: [Wi-fi, web, reaches, farmers, Peru, network, of, community, computer, centres, linked, by, wireless, technology, providing, helping, hand, poor, farmers, Peru, The,
pilot, scheme, Huaral, Valley, 80, kilometres, north, of, the, capital, Lima, aims, to, offer, the, 6000-strong, community, up-to-date, information, on, agricultural, market, prices, and, trends, The, Agricultural, Information, Project,
Farmers, of, the, Chancay-Huaral, Valley, provides, vital, links, between, local, organisations, charge, of, water, irrigation, enabling, them, to, coordinate, actions, More, than, 13000, rural, inhabitants, well, 18000, students, region,
also, benefit, telecoms, infrastructure, The, 14, telecentres, only, free, open, source, software, and, affordable, computer, equipment, The, network, been, years, making, and, officially, inaugurated, September, The, non-government, organisation, Cepes,
(Peruvian, Centre, Social, Studies), led, the, $200000, project, by, local, institutions, the, Education, and, Agriculture, ministries, and, European, development, organisations, "The, plan, includes, training, on, computers, and, internet, skills, both,
operators, and, users, of, the, system", said, Carlos, Saldarriaga, technical, coordinator, at, Cepes, Farmers, also, taking, extra, lessons, on, how, to, apply, the, information, to, make, the, plots, of, land, The, Board, of, Irrigation, Users, runs,
the, computer, centres, aims, to, make, the, network, self-sustainable, three, through, the, cash, generated, by, using, the, telecentres, internet, cafes, One, of, the, key, elements, of, the, project, Agricultural, Information, System, with, flagship,
huaralorg, website, There, farmers, can, find, the, prices, local, produce, well, information, on, topics, ranging, plague, prevention, to, the, latest, farming, techniques, The, system, helps, the, inhabitants, of, the, Chancay-Huaral, Valley, to,
organise, vital, irrigation, systems, "Water, main, element, unites, them, all, It, a, precious, element, Peru's, coastal, because, it, so, scarce, and, it, to, have, proper, irrigation, systems, to, make, the, it", Saldarriaga, told, the, BBC, News,
website, The, information, network, allows, farmers, to, look, beyond, own, region, and, share, experiences, with, colleagues, rest, of, Peru, and, even, around, the, world, Cepes, says, the, involvement, of, the, farmers, been, key, project's, success,
"Throughout, the, last, years, the, people, have, provided, vital, thrust, to, the, project;, feel, it, belongs, to, them", said, Saldarriaga, The, community, training, sessions, attended, by, equal, men, and, women, have, been, the, perfect, showcase,
their, enthusiasm, "We, have, had, excellent, response, mainly, people, have, a, great, feedback, when, we, trained, 40, or, 50-year, women, were, seeing, computer, first, time, their, lives", far, the, Huaral, programme, promoters, experience, been,
positive, and, already, planning, on, spreading, the, model, among, farmers', organisations, Peru, "This, a, pilot, project, and, we, have, been, keen, on, cloning, potential, other, places", underlined, Saldarriaga, The, Cepes, researcher, recalls,
happened, Cuyo, 50-family, community, with, electricity, during, the, construction, of, the, local, telecentre, site, There, it, necessary, to, build, mini-hydraulic, dam, order, to, generate, 2kW, worth, of, power, computers, the, communications,
equipment, and, the, cabin, lights, "It, already, dark, when, the, technicians, realised, didn't, have, any, light, bulbs, to, test, the, generator, they, to, the, local, store, to, buy, light, bulbs", recalls, Carlos, Saldarriaga, "The, logical, answer,
'we, don't, sell, any', they, had, to, wait, next, morning, to, do, the, testing", the, wireless, network, Cuyo, well, other, communities, no, longer, isolated]
2022-06-23T22:40:44.952-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: {Irrigation=1, BBC=1, Cepes=4, Wi-fi=1, September=1, One=1, Cuyo=2, Users=1, It=1, Information=2, Peru's=1, System=1, The=10, More=1, Board=1, Ther…
```

```
2022-06-23T22:40:45.415-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: "006ne"::{"Irrigation":1,"BBC":1,"Cepes":4,"Wi-fi":1,"September":1,"One":1,"Cuyo":2,"Users":1,"It":1,"Information":2,"Peru\u0027s":1,"System":1,"Th…
assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: "006ne"::
{
    "Irrigation": 1,
    "BBC": 1,
    "Cepes": 4,
    "Wi-fi": 1,
    "September": 1,
    "One": 1,
    "Cuyo": 2,
    "Users": 1,
    "It": 1,
    "Information": 2,
    "Peru's": 1,
    "System": 1,
    "The": 10,
    "More": 1,
    "Board": 1,
    "There": 2,
    "Lima": 1,
    "Education": 1,
    "News": 1,
    "Agriculture": 1,
    "Social": 1,
    "Farmers": 2,
    "Saldarriaga": 5,
    "Agricultural": 2,
    "Centre": 1,
    "European": 1,
    "Studies)": 1,
    "Project": 1,
    "Huaral": 2,
    "Valley": 3,
    "Chancay-Huaral": 2,
    "Peru": 4,
    "Carlos": 2
}
2022-06-23T22:40:46.035-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: Content written to file 006ne.txt successfully!
2022-06-23T22:40:46.056-07:00    END RequestId: 19a7cf85-60eb-4c61-910d-494a1c1dfde9
2022-06-23T22:40:46.056-07:00    REPORT RequestId: 19a7cf85-60eb-4c61-910d-494a1c1dfde9 Duration: 13226.64 ms Billed Duration: 13227 ms Memory Size: 512 MB Max Memory Used: 145 MB Init Duration: 500.19 ms
                                 No newer events at this moment. Auto retry paused. Resume
```

Figure 14: Json array of named entity without the sto words

Figure 15 shows the log for the creation of new file in “**tagsb00899516**” Amazon S3 bucket which is txt file and the content in that file is in JSON format

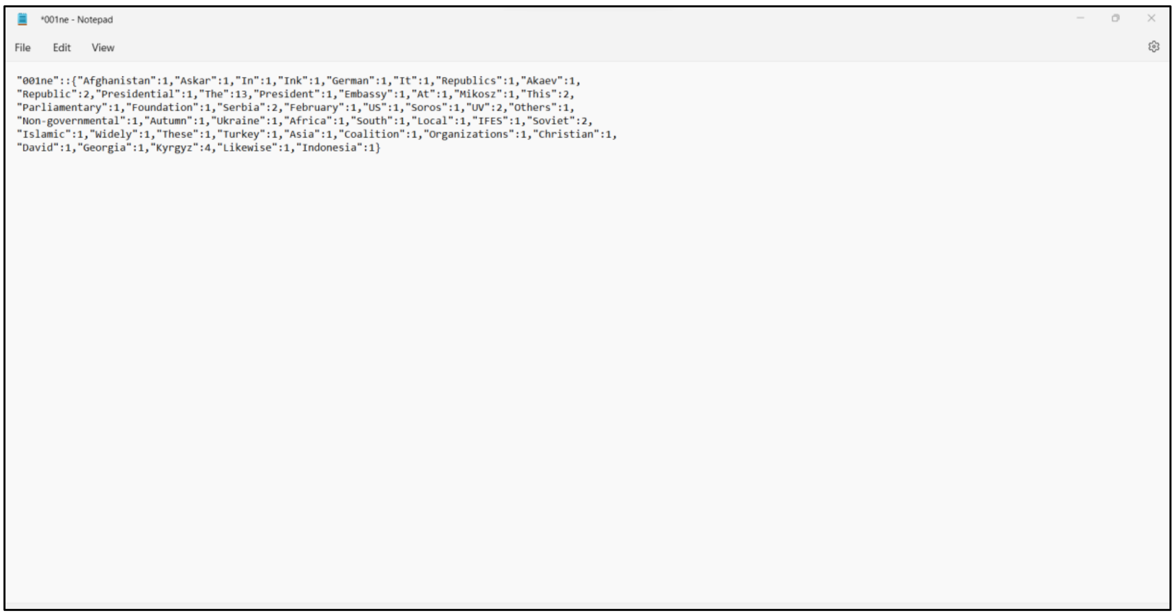

CloudWatch
Log groups
/aws/lambda/extractFeatures
2022/06/24/[$LATEST]40a69e609ee34dd9b4261245cbde9095
Log events
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
View as text
Actions
Create Metric Filter
Clear
1m
30m
1h
12h
Custom
Copy

```
Timestamp                        Message
                                 No older events at this moment. Retry
2022-06-24T05:03:07.542-07:00    START RequestId: 09ef94ef-1af0-4475-86d0-92a21de1d012 Version: $LATEST
2022-06-24T05:03:07.860-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: Bucket name: sourceb00899516 File name key: 006.txt
2022-06-24T05:03:19.641-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: [Wi-fi, web, reaches, farmers, Peru, network, of, community, computer, centres, linked, by, wireless, technology, providing, helping, hand, poor, farm…
2022-06-24T05:03:19.661-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: {Irrigation=1, BBC=1, Cepes=4, Wi-fi=1, September=1, One=1, Cuyo=2, Users=1, It=1, Information=2, Peru'S=1, System=1, The=10, More=1, Board=1, There=2…
2022-06-24T05:03:20.324-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: "006ne"::{"Irrigation":1,"BBC":1,"Cepes":4,"Wi-fi":1,"September":1,"One":1,"Cuyo":2,"Users":1,"It":1,"Information":2,"Peru\u0027s":1,"System":1,"The":…
2022-06-24T05:03:20.966-07:00    assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: Content written to file 006ne.txt successfully!
assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction: Content written to file 006ne.txt successfully!
2022-06-24T05:03:21.005-07:00    END RequestId: 09ef94ef-1af0-4475-86d0-92a21de1d012
2022-06-24T05:03:21.005-07:00    REPORT RequestId: 09ef94ef-1af0-4475-86d0-92a21de1d012 Duration: 13462.05 ms Billed Duration: 13463 ms Memory Size: 512 MB Max Memory Used: 145 MB Init Duration: 511.29 ms
                                 No newer events at this moment. Auto retry paused. Resume
```

Figure 15: Logs for the creation new file in "tagsb00899516" Amazon S3 bucket

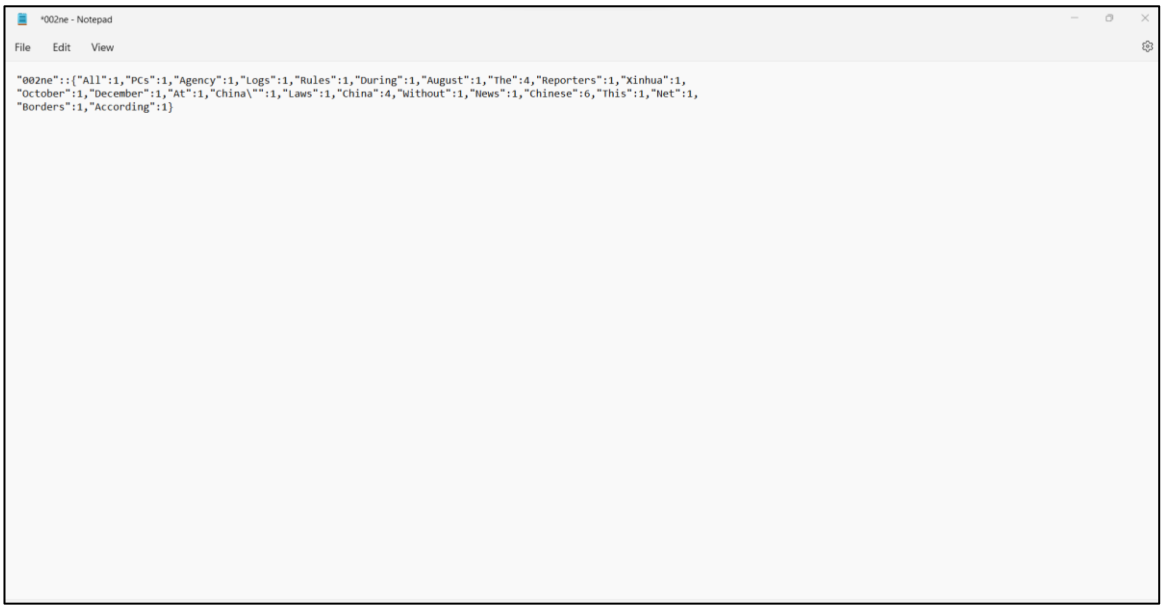
E. E.g. 001.txt contains Asia, Soviet, Serbia etc., then the JSON array created by the function should be “001ne”: {“Asia”:1, “Soviet”:1….etc.}.

Figure 16 and 17 is responsible for showing the file content which is in json format of the uploaded file on the “**tagsb00899516**” Amazon S3 bucket



```
"001ne"::{"Afghanistan":1,"Askar":1,"In":1,"Ink":1,"German":1,"It":1,"Republics":1,"Akaev":1,
"Republic":2,"Presidential":1,"The":13,"President":1,"Embassy":1,"At":1,"Mikosz":1,"This":2,
"Parliamentary":1,"Foundation":1,"Serbia":2,"February":1,"US":1,"Soros":1,"UV":2,"Others":1,
"Non-governmental":1,"Autumn":1,"Ukraine":1,"Africa":1,"South":1,"Local":1,"IFES":1,"Soviet":2,
"Islamic":1,"Widely":1,"These":1,"Turkey":1,"Asia":1,"Coalition":1,"Organizations":1,"Christian":1,
"David":1,"Georgia":1,"Kyrgyz":4,"Likewise":1,"Indonesia":1}
```

Figure 16:JSON array of Named Entities

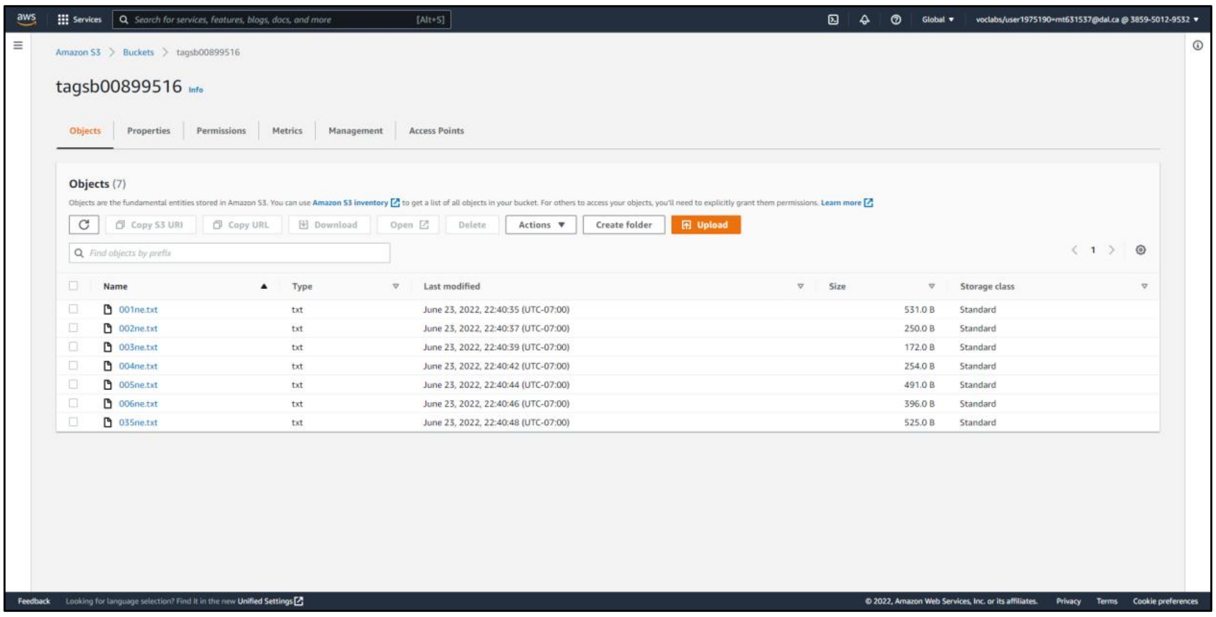


```
"002ne"::{"All":1,"PCs":1,"Agency":1,"Logs":1,"Rules":1,"During":1,"August":1,"The":4,"Reporters":1,"Xinhua":1,
"October":1,"December":1,"At":1,"China\"":1,"Laws":1,"China":4,"Without":1,"News":1,"Chinese":6,"This":1,"Net":1,
"Borders":1,"According":1}
```

Figure 17: JSON array of Named Entities

F. This file will be saved as **001ne.txt** in a new bucket - TagsB00xxxxxx.

Figure 18 is responsible for displaying all the files that are uploaded to the “tagsb00899516” Amazon S3 bucket.


Services
Search for services, features, blogs, docs, and more [Alt+S]
Global
voclabs/user1975190=mt631537@dal.ca @ 3859-5012-9532
Amazon S3 > Buckets > tagsb00899516
Objects
Properties
Permissions
Metrics
Management
Access Points
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Find objects by prefix
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

tagsb00899516 Info

Objects (7)

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

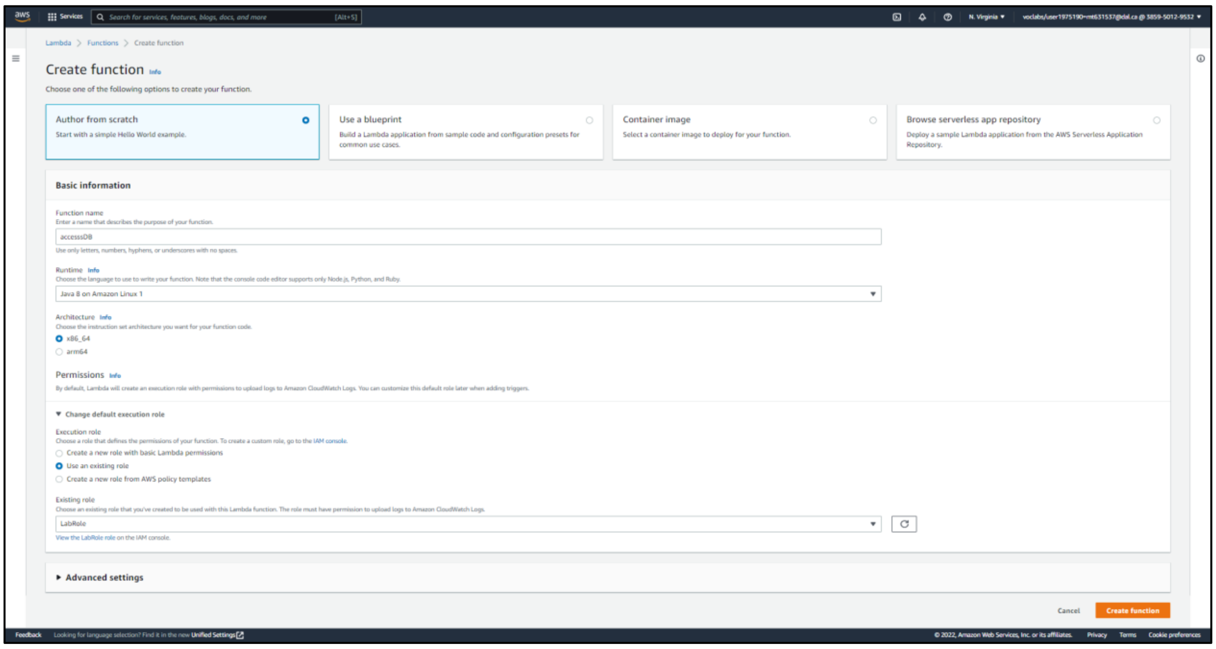
Name	Type	Last modified	Size	Storage class
001ne.txt	txt	June 23, 2022, 22:40:35 (UTC-07:00)	531.0 B	Standard
002ne.txt	txt	June 23, 2022, 22:40:37 (UTC-07:00)	250.0 B	Standard
003ne.txt	txt	June 23, 2022, 22:40:39 (UTC-07:00)	172.0 B	Standard
004ne.txt	txt	June 23, 2022, 22:40:42 (UTC-07:00)	254.0 B	Standard
005ne.txt	txt	June 23, 2022, 22:40:44 (UTC-07:00)	491.0 B	Standard
006ne.txt	txt	June 23, 2022, 22:40:46 (UTC-07:00)	396.0 B	Standard
035ne.txt	txt	June 23, 2022, 22:40:48 (UTC-07:00)	525.0 B	Standard

Figure 18: Files uploaded to the "tagsb00899516" Amazon S3 bucket

G. Once the file is available on this 2nd bucket, then **accessDB** Lambda function will automatically be triggered.

Creation Of Lambda Function ("accessDB")

Figure 19 is responsible for the showing the creation "**accessDB**" lambda function and its configuration



Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

- [✓ Author from scratch] Start with a simple Hello World example.
- Use a blueprint: Build a Lambda application from sample code and configuration presets for common use cases.
- Container image: Select a container image to deploy for your function.
- Browse serverless app repository: Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name: Enter a name that describes the purpose of your function.
accesssDB
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info: Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Java 8 on Amazon Linux 1

Architecture Info: Choose the instruction set architecture you want for your function code.
[✓ x86_64] arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role: Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
- Create a new role with basic Lambda permissions
- [✓ Use an existing role]
- Create a new role from AWS policy templates

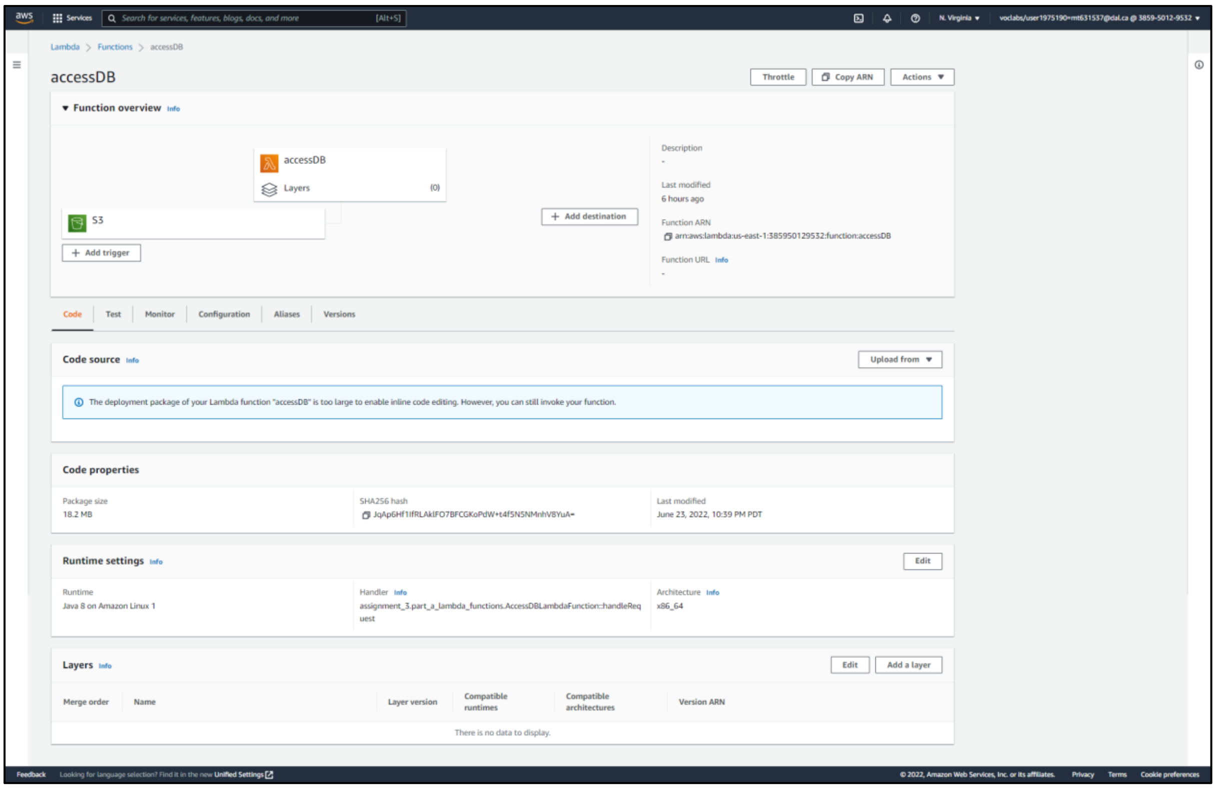
Existing role: Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
LabRole
View the LabRole role on the IAM console.

▶ Advanced settings

Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1975190=mt631537@dal.ca @ 3859-5012-9532
Cancel
Create function
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

*Figure 19: Creation "**accessDB**" lambda function and its configuration*

Figure 20 shows the successfully created lambda function on Amazon Lambda named as "**accessDB**"


Lambda > Functions > accessDB

accessDB

Function overview

Description
-

Last modified
6 hours ago

Function ARN
arn:aws:lambda:us-east-1:385950129532:function:accessDB

Function URL
-

Code source

The deployment package of your Lambda function "accessDB" is too large to enable inline code editing. However, you can still invoke your function.

Code properties

Package size	SHA256 hash	Last modified
18.2 MB	JqAp6Hf1IfRLAkIFO7BFCGKoPdW+t4f5N5NMnhV8YuA=	June 23, 2022, 10:39 PM PDT

Runtime settings

Runtime	Handler	Architecture
Java 8 on Amazon Linux 1	assignment_3.part_a_lambda_functions.AccessDBLambdaFunction::handleRequest	x86_64

Layers

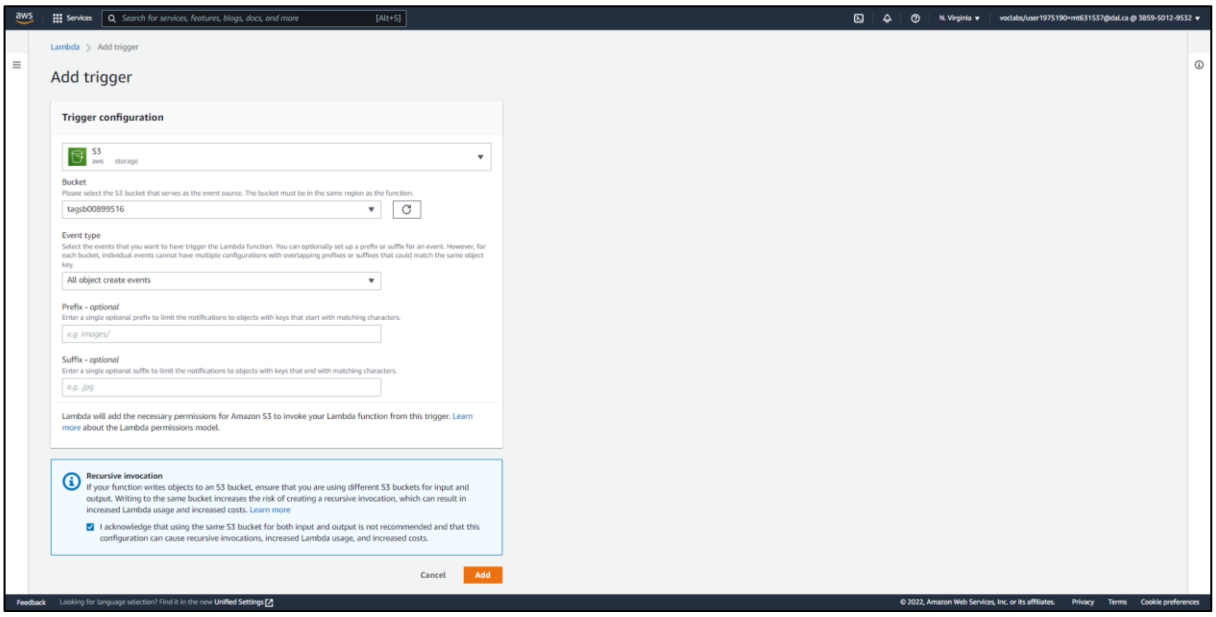
Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN

There is no data to display.

© 2022, Amazon Web Services, Inc. or its affiliates.

*Figure 20: Successful creation of Lambda Function "**accessDB**"*

Figure 21 shows the creation of trigger for "**tagsb00899516**" Amazon S3 bucket, and it will be triggered when object is uploaded to the "**tagsb00899516**"


Lambda > Add trigger

Add trigger

Trigger configuration

S3
aws storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.
tagsb00899516

Event type
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

Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/

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

Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

Recursive invocation
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. Learn more

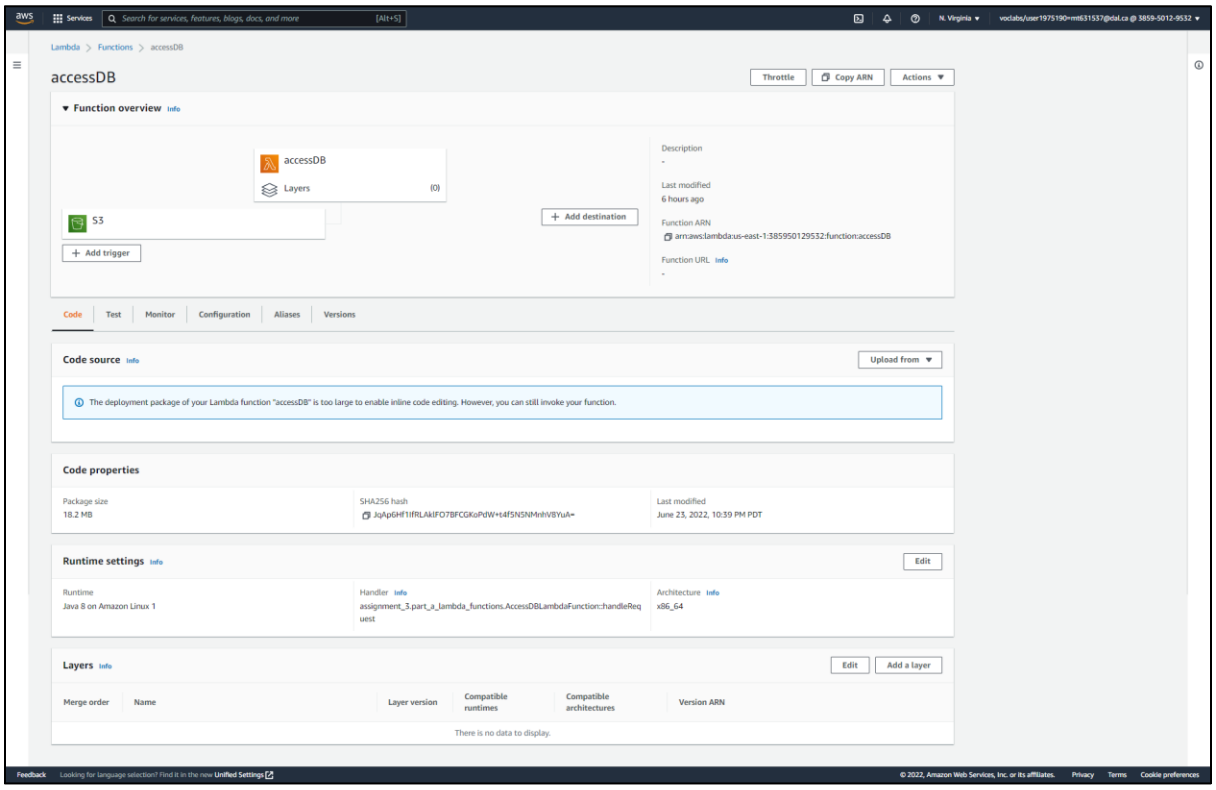
☑ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

© 2022, Amazon Web Services, Inc. or its affiliates.

Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1975190=mt631537@dal.ca @ 3859-5012-9532
Cancel
Add
Feedback
Looking for language selection? Find it in the new Unified Settings
Privacy
Terms
Cookie preferences

*Figure 21: Creation of trigger for "**tagsb00899516**" Amazon S3 bucket*

Figure 22 shows the successful creation of trigger for “**tagsb00899516**” S3 bucket.


Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1975190=mt631537@dal.ca @ 3859-5012-9532
Lambda > Functions > accessDB
Throttle
Copy ARN
Actions
Code
Test
Monitor
Configuration
Aliases
Versions
Upload from
Edit
Add a layer
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

accessDB

Function overview Info

accessDB
Layers (0)
S3
+ Add trigger
+ Add destination

Description
-
Last modified
6 hours ago
Function ARN
arn:aws:lambda:us-east-1:385950129532:function:accessDB
Function URL Info
-

Code source Info

The deployment package of your Lambda function "accessDB" is too large to enable inline code editing. However, you can still invoke your function.

Code properties

Package size	SHA256 hash	Last modified
18.2 MB	JqAp6Hf1IfRLAkIFO7BFCGKoPdW+t4f5N5NMnhV8YuA=	June 23, 2022, 10:39 PM PDT

Runtime settings Info

Runtime	Handler Info	Architecture Info
Java 8 on Amazon Linux 1	assignment_3.part_a_lambda_functions.AccessDBLambdaFunction::handleRequest	x86_64

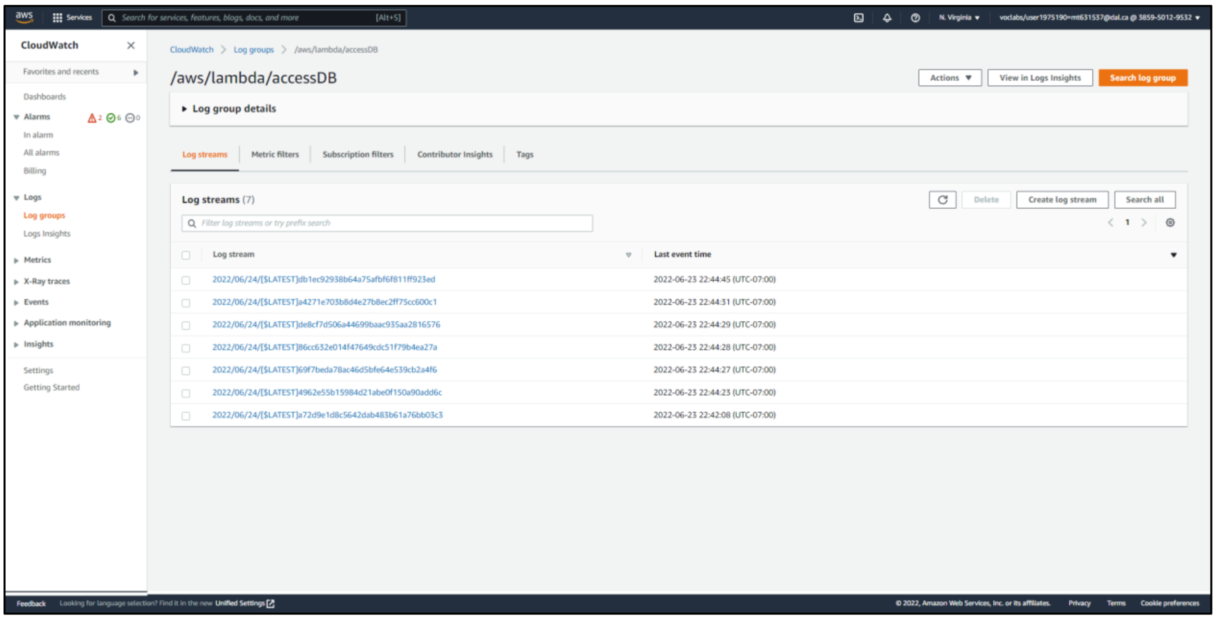
Layers Info

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN

There is no data to display.

Figure 22: Successful creation of trigger for “tagsb00899516” S3 bucket.

Figure 23 is shows that when object is uploaded to the “**tagsb00899516**” Amazon S3 bucket. Lambda function “**accessDB**” will trigger. The below figure shows the logs from CloudWatch.



/aws/lambda/accessDB

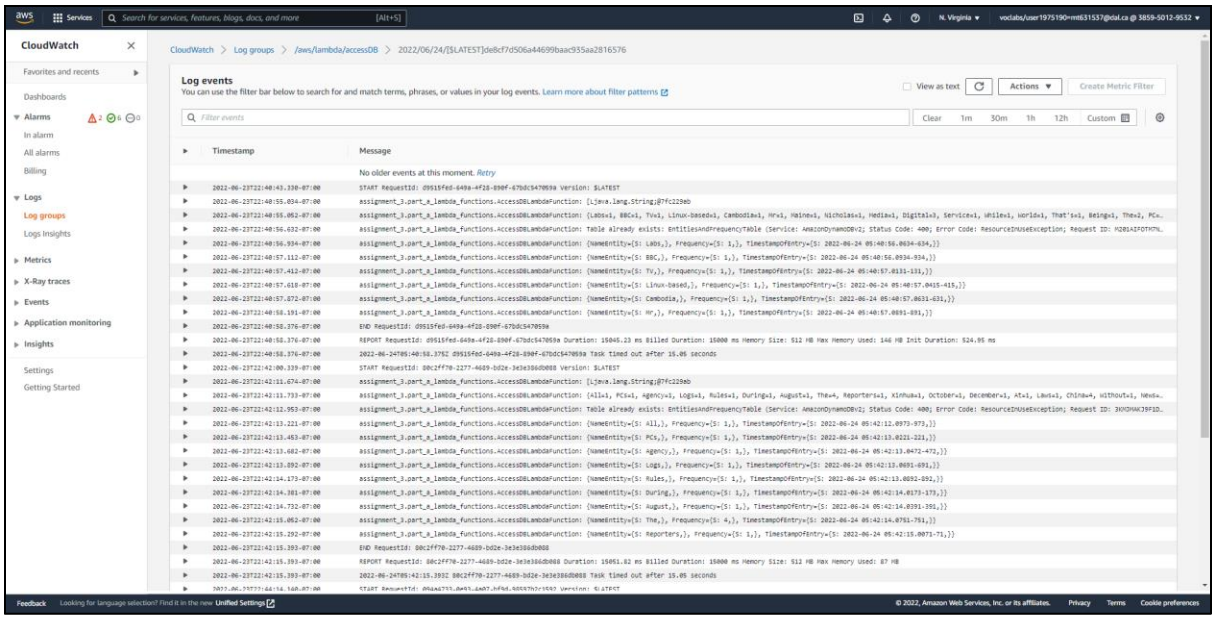
Log group details

Log streams (7)

Log stream	Last event time
2022/06/24/[$LATEST]db1ec92938b64a75afbf6f811ff923ed	2022-06-23 22:44:45 (UTC-07:00)
2022/06/24/[$LATEST]a4271e703b8d4e27b8ec2ff75cc600c1	2022-06-23 22:44:31 (UTC-07:00)
2022/06/24/[$LATEST]de8cf7d506a44699baac935aa2816576	2022-06-23 22:44:29 (UTC-07:00)
2022/06/24/[$LATEST]86cc632e014f47649cdc51f79b4ea27a	2022-06-23 22:44:28 (UTC-07:00)
2022/06/24/[$LATEST]69f7beda78ac46d5bfe64e539cb2a4f6	2022-06-23 22:44:27 (UTC-07:00)
2022/06/24/[$LATEST]4962e55b15984d21abe0f150a90add6c	2022-06-23 22:44:23 (UTC-07:00)
2022/06/24/[$LATEST]a72d9e1d8c5642dab483b61a76bb03c3	2022-06-23 22:42:08 (UTC-07:00)

Figure 23: Logs for the accessDb lambda function

Figure 24 shows logs for accessDB in detail.

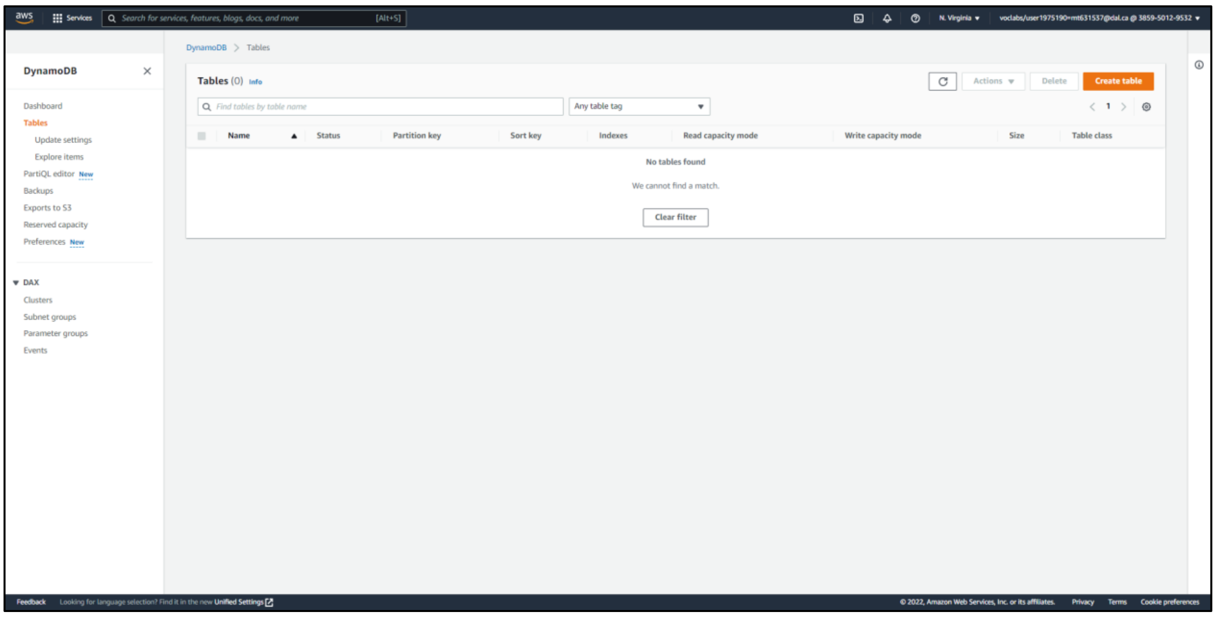

aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user1975190=mt631537@dal.ca @ 3859-5012-9532
CloudWatch
Favorites and recents
Dashboards
Alarms 2 6 0
In alarm
All alarms
Billing
Logs
Log groups
Logs Insights
Metrics
X-Ray traces
Events
Application monitoring
Insights
Settings
Getting Started
CloudWatch > Log groups > /aws/lambda/accessDB > 2022/06/24/[$LATEST]de8cf7d506a44699baac935aa2816576
Log events
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
View as text
Actions
Create Metric Filter
Filter events
Clear 1m 30m 1h 12h Custom
Timestamp
Message
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

```
No older events at this moment. Retry
2022-06-23T22:40:43.330-07:00  START RequestId: d9515fed-649a-4f28-890f-67bdc547059a Version: $LATEST
2022-06-23T22:40:55.034-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: [Ljava.lang.String;@7fc229eb
2022-06-23T22:40:55.052-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {Labs=1, BBC=1, TV=1, Linux-based=1, Cambodia=1, Mr=1, Maine=1, Nicholas=1, Media=1, Digital=3, Service=1, While=1, World=1, That's=1, Being=1, The=2, PC=…
2022-06-23T22:40:56.632-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: Table already exists: EntitiesAndFrequencyTable (Service: AmazonDynamoDBv2; Status Code: 400; Error Code: ResourceInUseException; Request ID: M201AIFOTM7N…
2022-06-23T22:40:56.934-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: Labs,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:40:56.0634-634,}}
2022-06-23T22:40:57.112-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: BBC,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:40:56.0934-934,}}
2022-06-23T22:40:57.412-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: TV,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:40:57.0131-131,}}
2022-06-23T22:40:57.618-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: Linux-based,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:40:57.0415-415,}}
2022-06-23T22:40:57.872-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: Cambodia,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:40:57.0631-631,}}
2022-06-23T22:40:58.191-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: Mr,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:40:57.0891-891,}}
2022-06-23T22:40:58.376-07:00  END RequestId: d9515fed-649a-4f28-890f-67bdc547059a
2022-06-23T22:40:58.376-07:00  REPORT RequestId: d9515fed-649a-4f28-890f-67bdc547059a Duration: 15045.23 ms Billed Duration: 15000 ms Memory Size: 512 MB Max Memory Used: 146 MB Init Duration: 524.95 ms
2022-06-23T22:40:58.376-07:00  2022-06-24T05:40:58.375Z d9515fed-649a-4f28-890f-67bdc547059a Task timed out after 15.05 seconds
2022-06-23T22:42:00.339-07:00  START RequestId: 80c2ff70-2277-4689-bd2e-3e3e386db088 Version: $LATEST
2022-06-23T22:42:11.674-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: [Ljava.lang.String;@7fc229eb
2022-06-23T22:42:11.733-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {All=1, PCs=1, Agency=1, Logs=1, Rules=1, During=1, August=1, The=4, Reporters=1, Xinhua=1, October=1, December=1, At=1, Laws=1, China=4, Without=1, News=…
2022-06-23T22:42:12.953-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: Table already exists: EntitiesAndFrequencyTable (Service: AmazonDynamoDBv2; Status Code: 400; Error Code: ResourceInUseException; Request ID: 3KMJMAKJ9F1D…
2022-06-23T22:42:13.221-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: All,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:42:12.0973-973,}}
2022-06-23T22:42:13.453-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: PCs,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:42:13.0221-221,}}
2022-06-23T22:42:13.682-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: Agency,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:42:13.0472-472,}}
2022-06-23T22:42:13.892-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: Logs,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:42:13.0691-691,}}
2022-06-23T22:42:14.173-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: Rules,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:42:13.0892-892,}}
2022-06-23T22:42:14.381-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: During,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:42:14.0173-173,}}
2022-06-23T22:42:14.732-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: August,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:42:14.0391-391,}}
2022-06-23T22:42:15.052-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: The,}, Frequency={S: 4,}, TimestampOfEntry={S: 2022-06-24 05:42:14.0751-751,}}
2022-06-23T22:42:15.292-07:00  assignment_3.part_a_lambda_functions.AccessDBLambdaFunction: {NameEntity={S: Reporters,}, Frequency={S: 1,}, TimestampOfEntry={S: 2022-06-24 05:42:15.0071-71,}}
2022-06-23T22:42:15.393-07:00  END RequestId: 80c2ff70-2277-4689-bd2e-3e3e386db088
2022-06-23T22:42:15.393-07:00  REPORT RequestId: 80c2ff70-2277-4689-bd2e-3e3e386db088 Duration: 15051.82 ms Billed Duration: 15000 ms Memory Size: 512 MB Max Memory Used: 87 MB
2022-06-23T22:42:15.393-07:00  2022-06-24T05:42:15.393Z 80c2ff70-2277-4689-bd2e-3e3e386db088 Task timed out after 15.05 seconds
2022-06-23T22:44:14.140-07:00  START RequestId: 094a4733-0e93-4a07-bf9d-98597b2c1592 Version: $LATEST
```

Figure 24: Logs in detail

H. accessDB is your 2nd Lambda function. This Lambda function reads each named entity JSON file and updates the DynamoDb database table (three entries/array - NameEntity, Frequency, TimeStamp of Entry).

Figure 25 and 26 shows the table entry in the **Amazon DynamoDB** when the lambda function **accessDB** triggered. The name of the table entry is "**EntitiesAndFrequencyTable**"



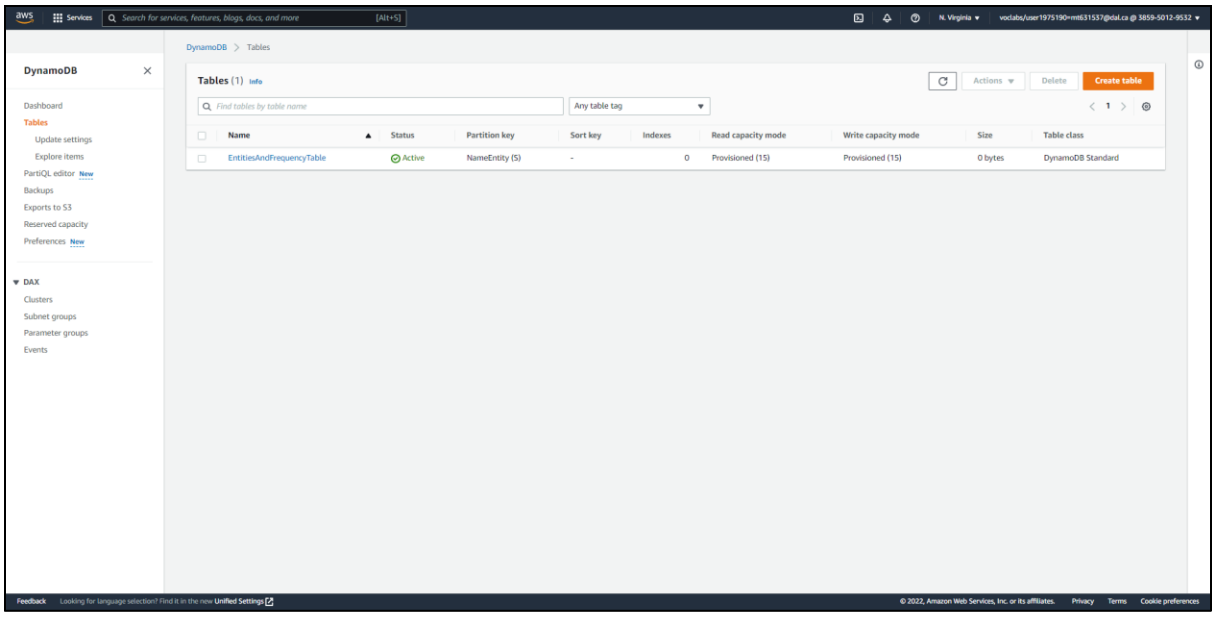
Tables (0)

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode	Size	Table class

No tables found

We cannot find a match.

Figure 25: Empty DynamoDB database

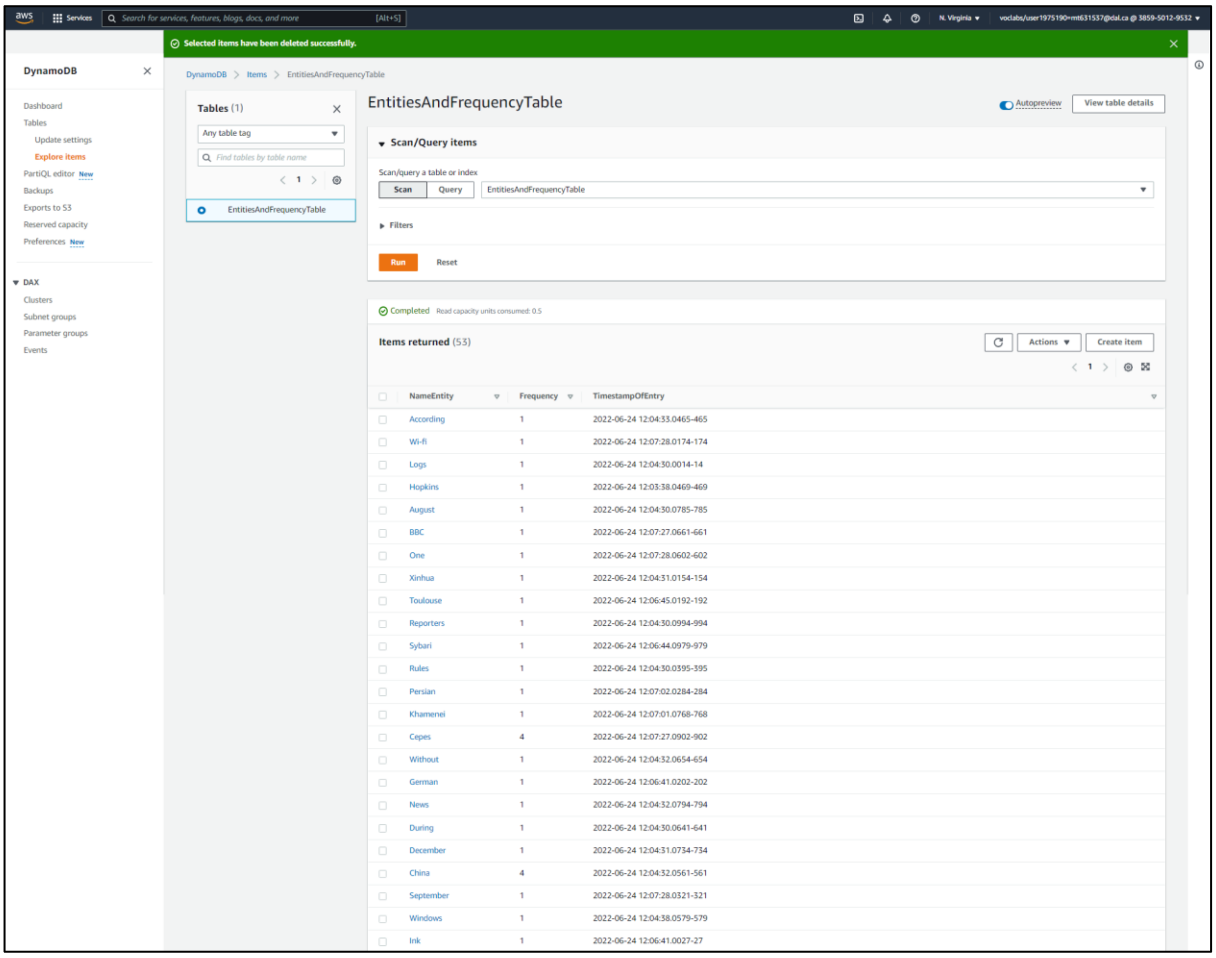


Tables (1)

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacity mode	Size	Table class
EntitiesAndFrequencyTable	Active	NameEntity (S)	-	0	Provisioned (15)	Provisioned (15)	0 bytes	DynamoDB Standard

Figure 26: Table entry in the DynamoDB database

Figure 27 shows the "**EntitiesAndFrequencyTable**" with attributes NameEntity, Frequency, and TimestampOfEntry



Selected items have been deleted successfully.

EntitiesAndFrequencyTable

Scan/query a table or index: Scan — EntitiesAndFrequencyTable

Completed Read capacity units consumed: 0.5

Items returned (53)

NameEntity	Frequency	TimestampOfEntry
According	1	2022-06-24 12:04:33.0465-465
Wi-fi	1	2022-06-24 12:07:28.0174-174
Logs	1	2022-06-24 12:04:30.0014-14
Hopkins	1	2022-06-24 12:03:38.0469-469
August	1	2022-06-24 12:04:30.0785-785
BBC	1	2022-06-24 12:07:27.0661-661
One	1	2022-06-24 12:07:28.0602-602
Xinhua	1	2022-06-24 12:04:31.0154-154
Toulouse	1	2022-06-24 12:06:45.0192-192
Reporters	1	2022-06-24 12:04:30.0994-994
Sybari	1	2022-06-24 12:06:44.0979-979
Rules	1	2022-06-24 12:04:30.0395-395
Persian	1	2022-06-24 12:07:02.0284-284
Khamenei	1	2022-06-24 12:07:01.0768-768
Cepes	4	2022-06-24 12:07:27.0902-902
Without	1	2022-06-24 12:04:32.0654-654
German	1	2022-06-24 12:06:41.0202-202
News	1	2022-06-24 12:04:32.0794-794
During	1	2022-06-24 12:04:30.0641-641
December	1	2022-06-24 12:04:31.0734-734
China	4	2022-06-24 12:04:32.0561-561
September	1	2022-06-24 12:07:28.0321-321
Windows	1	2022-06-24 12:04:38.0579-579
Ink	1	2022-06-24 12:06:41.0027-27

Figure 27: "EntitiesAndFrequencyTable" with attributes NameEntity, Frequency,

Program/Scripts

Part A Serverless Application

AWS Connection.java

```
public class AWSConnection {
    private static final String AWS_ACCESS_KEY_ID =
"ASIAVTXDLTV6KQROAMCN";
    private static final String AWS_SECRET_ACCESS_KEY =
"CZFd7kNhFHhElIzb5zdAqe9jizBl5SObNn/I38OA";

    private static final String AWS_SESSION_TOKEN =
"FwoGZXIvYXdzEIX//////////wEaDO7rUh9qgArwGtlLbSLAATsjzoWPaBNNi" +

"GSOEhr9b9Xjr+P1CMf6ItDGY3643cUh9oeyJ/FkN3WoqjxkCCczAbrc6Y90F9ka3tBmB3a9jm
/rHO" +

"7fCCzAmESniUELZqaxLNOM2U9lhDmS9k3sF1K0VPlnjFvJVMZT3apEgxGOZnCJD7DA80uk4TF
4Ft" +

"TUYeYhpVb3kS9B1nb7csUjvh26JHoN33pUcvil/KPEvuOEe+cIhJMFZ/wjRnxMZXl+Dnmzjuk
c7" +

"n8b4HN9dgyYv+YxGyiuvtaVBjItamLu54vhiqtEKR6ABboTl8JfRSDuD1mjbrfJWYBNewNkWJ
12" +
            "gphFztYVxmJf";

    public AmazonS3 createAmazonS3ClientBuilder() {
        BasicSessionCredentials basicSessionCredentials = new
BasicSessionCredentials(AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY,
AWS_SESSION_TOKEN);

        AmazonS3 amazonS3object = AmazonS3ClientBuilder.standard()
                .withCredentials(new
AWSStaticCredentialsProvider(basicSessionCredentials))
                .withRegion(Regions.US_EAST_1)
                .build();

        return amazonS3object;
    }
}
```

S3 Bucket.java

```
public class S3Bucket {

    Bucket sourceBucket;
    Bucket tagsBucket;
    public S3Bucket() {
    }

    public boolean isBucketExists(String bucketName, AmazonS3
amazonS3ClientBuilder){
        List<Bucket> bucketList = amazonS3ClientBuilder.listBuckets();
        for (Bucket bucket: bucketList){
            if(bucket.getName().equals(bucketName)){
                return true;
            }
        }
        return false;
    }

    public Bucket createBucket(String bucketName){
        AWSConnection awsConnection = new AWSConnection();
        if (isBucketExists(bucketName,
awsConnection.createAmazonS3ClientBuilder())){
            System.out.println("Bucket "+bucketName+ " already
exists....");
        }else {
            try {
                System.out.println("Creating bucket with name " +
bucketName + "...");
                Bucket bucket =
awsConnection.createAmazonS3ClientBuilder().createBucket(bucketName);
                System.out.println("Bucket with name "+bucketName+ " is
created successfully");
                System.out.println("------------------------------------
");
                return bucket;
            }catch (AmazonS3Exception e){
                System.err.println(e.getErrorMessage());
            }
        }
        return null;
    }

    public void createbucketWithName(){
        AWSConnection awsConnection = new AWSConnection();
        try{
            final String bucketName1 = "sourceb00899516";
            final String bucketName2 = "tagsb00899516";
            if (awsConnection.createAmazonS3ClientBuilder() != null){
                sourceBucket = createBucket(bucketName1);
                tagsBucket = createBucket(bucketName2);
            }
        }catch (Exception e){
            e.printStackTrace();
        }
    }
}
```

UploadFile.java

```
public class UploadFile {
    AWSConnection awsConnection = new AWSConnection();
    S3Bucket s3Bucket = new S3Bucket();
    public void uploadSingleFile(String bucketName, File file) {
        final String fileName = file.getName();
        System.out.println("Uploading "+fileName+" to "+ bucketName +" S3 
bucket...");
        try {

awsConnection.createAmazonS3ClientBuilder().putObject(bucketName, 
fileName, file);
            System.out.println("File uploaded successfully.");
        } catch (final AmazonServiceException e) {
            e.printStackTrace();
        }
    }
    public void uploadFiles(String folderName){
        final File[] allFiles = new File(folderName).listFiles();
        if (allFiles != null) {
            System.out.println("\n");
            System.out.println("Uploading files");
            System.out.println("------------------------------------------
");
            for (final File file : allFiles) {
                uploadSingleFile("sourceb00899516", file);
                try {
                    Thread.sleep(200);
                } catch (final InterruptedException e) {
                    e.printStackTrace();
                }
            }
            System.out.println("------------------------------------------
");
        } else {
            System.out.println("File upload Failed");
        }
    }
}
```

Main.java

```
public class Main {
    public static void main(String[] args) {
        S3Bucket s3Bucket = new S3Bucket();
        s3Bucket.createbucketWithName();
        UploadFile uploadFile = new UploadFile();
        uploadFile.uploadFiles("tech");
    }
}
```

Part A lambda Function

extractFeatures Lambda Function

```
public class ExtractFeaturesLambdaFunction implements RequestHandler<S3Event, String> {
    final static String LOG =
"assignment_3.part_a_lambda_functions.ExtractFeaturesLambdaFunction";

    public ExtractFeaturesLambdaFunction() throws IOException {
    }
    @Override
    public String handleRequest(final S3Event s3Event,
                                final Context context) {
        String bucketName = null;
        String fileNameKey = null;
        String fileNameEnding=  "ne \":";
        String fileExtention = "ne" + ".txt";
        try {
            // Source Link: https://www.techie-knowledge.co.in/2017/02/removing-stop-words-
from-text-using-java.html
            String[] stopWords = {  "a", "about", "above", "across", "after", "again",
                    "against", "all", "almost", "alone", "along", "already", "also",
                    "although", "always", "among", "an", "and", "another", "any",
                    "anybody", "anyone", "anything", "anywhere", "are", "area",
                    "areas", "around", "as", "ask", "asked", "asking", "asks", "at",
                    "away", "b", "back", "backed", "backing", "backs", "be", "became",
                    "behind", "being", "beings", "best", "better", "between", "big",
                    "both", "but", "by", "c", "f", "face", "faces",
                    "fact", "facts", "far", "felt", "few", "find", "finds", "first",
                    "for", "four", "from", "full", "fully", "further", "furthered",
                    "furthering", "furthers", "g", "gave", "general", "generally",
                    "get", "gets", "give", "given", "gives", "go", "going", "good",
                    "goods", "got", "great", "greater", "greatest", "group", "grouped",
                    "grouping", "groups", "h", "had", "has", "have", "having", "he",
                    "her", "here", "herself", "high", "high", "high", "higher",
                    "highest", "him", "himself", "his", "how", "however", "i", "if",
                    "important", "in", "interest", "interested", "interesting",
                    "interests", "into", "is", "it", "its", "itself", "j", "just", "k",
                    "keep", "keeps", "kind", "knew", "know", "known", "knows", "l",
                    "large", "largely", "last", "later", "latest", "least", "less",
                    "let", "lets", "like", "likely", "long", "longer", "longest", "m",
                    "made", "make", "making", "man", "many", "may", "me", "member",
                    "members", "men", "might", "more", "most", "mostly", "mr", "mrs",
                    "much", "must", "my", "myself", "n", "necessary", "need", "needed",
                    "needing", "needs", "never", "new", "new", "newer", "newest",
                    "next", "no", "nobody", "non", "noone", "not", "nothing", "now",
                    "nowhere", "number", "numbers", "o", "of", "off", "often", "old",
                    "older", "oldest", "on", "once", "one", "only", "open", "opened",
                    "opening", "opens", "or", "order", "ordered", "ordering", "orders",
                    "other", "others", "our", "out", "over", "p", "part", "parted",
                    "parting", "parts", "per", "perhaps", "place", "places", "point",
                    "pointed", "pointing", "points", "possible", "present",
                    "presented", "presenting", "presents", "problem", "problems",
                    "put", "puts", "q", "quite", "r", "rather", "really", "right",
                    "right", "room", "rooms", "s", "said", "same", "saw", "say",
                    "says", "second", "seconds", "see", "seem", "seemed", "seeming",
                    "seems", "sees", "several", "shall", "she", "should", "show",
                    "showed", "showing", "shows", "side", "sides", "since", "small",
                    "smaller", "smallest", "so", "some", "somebody", "someone",
                    "something", "somewhere", "state", "states", "still", "still",
                    "such", "sure", "t", "take", "taken", "than", "that", "the",
                    "their", "them", "then", "there", "therefore", "these", "they",
                    "thing", "things", "think", "thinks", "this", "those", "though",
                    "thought", "thoughts", "three", "through", "thus", "to", "today",
                    "together", "too", "took", "toward", "turn", "turned", "turning",
                    "turns", "two", "u", "under", "until", "up", "upon", "us", "use",
                    "used", "uses", "v", "very", "w", "want", "wanted", "wanting",
                    "wants", "was", "way", "ways", "we", "well", "wells", "went",
                    "were", "what", "when", "where", "whether", "which", "while",
                    "who", "whole", "whose", "why", "will", "with", "within",
                    "without", "work", "worked", "working", "works", "would", "x", "y",
                    "year", "years", "yet", "you", "young", "younger", "youngest",
                    "your", "yours", "z", "the" };
            Map<String, Integer> entities = new HashMap<>();
            bucketName = s3Event.getRecords().get(0).getS3().getBucket().getName();
            fileNameKey =
URLDecoder.decode(s3Event.getRecords().get(0).getS3().getObject().getKey().replace('+', ' '),
"UTF-8");
            context.getLogger().log(LOG + ": BucketName: " + bucketName + " FileName key: " +
fileNameKey);

            String fileContent =
AmazonS3ClientBuilder.standard().withRegion(Regions.US_EAST_1).build().getObjectAsString(buck
etName, fileNameKey);

            String[] listOfWords = fileContent
                    .replaceAll("\\.", "")
                    .replaceAll(",", "")
                    .replaceAll("\\s+(.*?)", " ")
                    .split(" ");

            ArrayList<String> listOfWordsWithoutStopwords = new ArrayList<>();

            for (int i=0; i< listOfWords.length; i++){
                for (int j=0; j<stopWords.length; j++){
                    if (listOfWords[i].equalsIgnoreCase(stopWords[j])){
                        i++;
                    } else {
                        j++;
                    }
                }
                listOfWordsWithoutStopwords.add(listOfWords[i]);
            }

            context.getLogger().log(LOG + ": " + listOfWordsWithoutStopwords);

            getWordFrequency(entities, listOfWordsWithoutStopwords);

            context.getLogger().log(LOG + ": " + entities);

            String fileName = "\"" + fileNameKey.split("\\.")[0] + fileNameEnding;

            Gson gsonObj = new Gson();
            String jsonStr = fileName + gsonObj.toJson(entities);

            context.getLogger().log(LOG + ": " + jsonStr);

            String newFileName = fileNameKey.split("\\.")[0] + fileExtention;

AmazonS3ClientBuilder.standard().withRegion(Regions.US_EAST_1).build().putObject("tagsb008995
16", newFileName, jsonStr);
            context.getLogger().log(LOG + ": " + "Content written to file " + newFileName + "
successfully!");
            return "correct";
        } catch (UnsupportedEncodingException e) {
            context.getLogger().log(LOG + ": " + e.getMessage());
            return "error";
        }
    }
    private void getWordFrequency(Map<String, Integer> entities, ArrayList<String>
listOfWords) {
        for (String word : listOfWords) {
            if (Character.isUpperCase(word.charAt(0))) {
                if (entities.containsKey(word)) {
                    entities.put(word, entities.get(word) + 1);
                } else {
                    entities.put(word, 1);
                }
            }
        }
    }
`
```

accessDB lambda Function

```
public class AccessDBLambdaFunction implements RequestHandler<S3Event, String> {
    String LOG = "assignment_3.part_a_lambda_functions.AccessDBLambdaFunction";

    @Override
    public String handleRequest(S3Event s3Event, Context context) {
        String bucketName = null;
        String fileNameKey = null;
        String ENTITIES_AND_FREQUENCY_TABLE = "EntitiesAndFrequencyTable";
        String NAME_ENTITY_COLUMN = "NameEntity";
        String FREQUENCY_COLUMN = "Frequency";
        String TIMESTAMP_OF_ENTRY = "TimestampOfEntry";
        Map<String, Integer> entityAndFrequencyMap = new HashMap<>();
        try {
            bucketName = s3Event.getRecords().get(0).getS3().getBucket().getName();
            fileNameKey =
URLDecoder.decode(s3Event.getRecords().get(0).getS3().getObject().getKey().replace('+', ' '),
"UTF-8");

            String fileContent =
AmazonS3ClientBuilder.standard().withRegion(Regions.US_EAST_1).build().getObjectAsString(buck
etName, fileNameKey);
            String[] entities = fileContent.split("\\{")[1].replaceAll("}", "").split(",");

            context.getLogger().log(LOG + ": " + entities);

            for (String entity : entities) {
                String entityName = entity.split(":")[0].replaceAll("\"",
"").replaceAll("\\\\u0027", "'");
                Integer entityFrequency = Integer.parseInt(entity.split(":")[1]);
                entityAndFrequencyMap.put(entityName, entityFrequency);
            }

            context.getLogger().log(LOG + ": " + entityAndFrequencyMap);

            try {
                final CreateTableResult createTableResult = AmazonDynamoDBClientBuilder
                        .standard().withRegion(Regions.US_EAST_1)
                        .build().createTable(new CreateTableRequest()
                        .withAttributeDefinitions(new AttributeDefinition(NAME_ENTITY_COLUMN,
ScalarAttributeType.S))
                        .withKeySchema(new KeySchemaElement(NAME_ENTITY_COLUMN,
KeyType.HASH))
                        .withProvisionedThroughput(new ProvisionedThroughput(15L, 15L))
                        .withTableName(ENTITIES_AND_FREQUENCY_TABLE));
            } catch (AmazonServiceException e) {
                context.getLogger().log(LOG + ": " + e.getMessage());
            }

            for (Map.Entry<String, Integer> entity : entityAndFrequencyMap.entrySet()) {
                Map<String, AttributeValue> columnItem = new HashMap<>();
                columnItem.put(NAME_ENTITY_COLUMN, new AttributeValue(entity.getKey()));
                columnItem.put(FREQUENCY_COLUMN, new
AttributeValue(String.valueOf(entity.getValue())));
                columnItem.put(TIMESTAMP_OF_ENTRY, new AttributeValue(new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSSS-SS")
                        .format(new Date())));

                AmazonDynamoDBClientBuilder
                        .standard().withRegion(Regions.US_EAST_1)
                        .build().putItem(ENTITIES_AND_FREQUENCY_TABLE, columnItem);

                context.getLogger().log(LOG + ": " + columnItem);
            }
            return "correct";
        } catch (UnsupportedEncodingException e) {
            return "error";
        }
    }
}
```

References

[1] AWS, "Amazon S3," Amazon, [Online]. Available: https://aws.amazon.com/s3/ . [Accessed 24 June 2022].

[2] AWS, "AWS Identity and Access Management (IAM)," Amazon, [Online]. Available: https://aws.amazon.com/iam/ . [Accessed 24 June 2022].

[3] AWS, "AWS Lambda," Amazon, [Online]. Available: https://aws.amazon.com/lambda/. [Accessed 24 June 2022].

[4] AWS, "Amazon CloudWatch," Amazon, [Online]. Available: https://aws.amazon.com/cloudwatch/ . [Accessed 24 June 2022].

[5] AWS, "Amazon DynamoDB," Amazon, [Online]. Available: https://aws.amazon.com/dynamodb/ . [Accessed 24 June 2022].

[6] "Removing stop words from Text using java," Techie-knowledge.co.in. [Online]. Available: https://www.techie-knowledge.co.in/2017/02/removing-stop-words-from-text-using-java.html . [Accessed 24 June 2022].