CSCI 5410

Assignment 3 – Part A

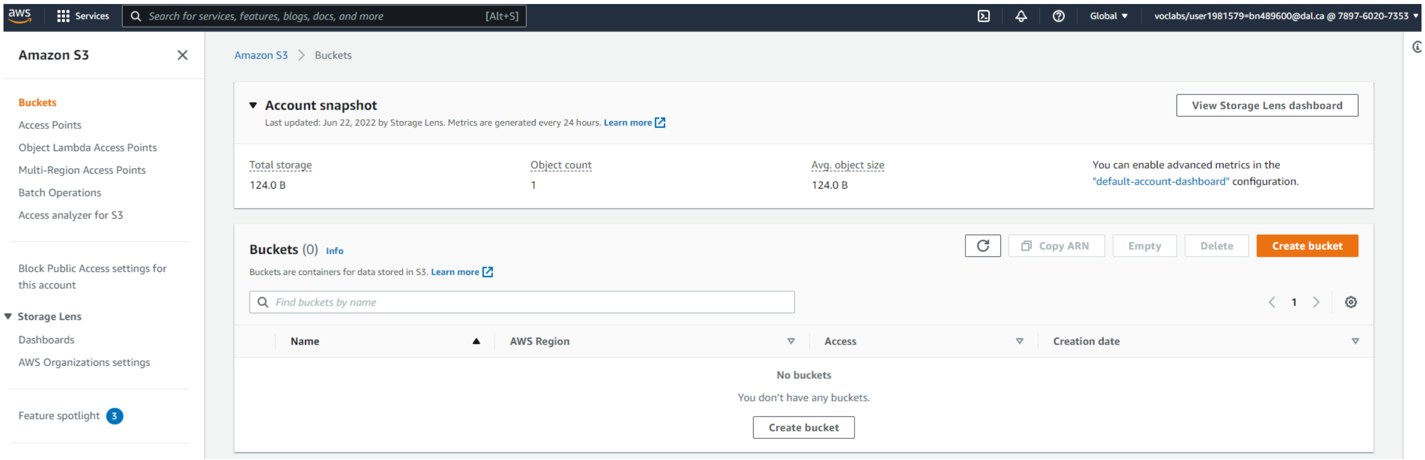
Name: Benny Daniel Tharigopala

Banner ID: B00899629

GitLab URL: https://git.cs.dal.ca/benny/csci5410_B00899629_Benny_Tharigopala

Event-driven Serverless application with AWS Lambda

S3Buckets Operations:


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global ▾
voclabs/user1981579=bn489600@dal.ca @ 7897-6020-7353 ▾
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
Amazon S3 > Buckets
View Storage Lens dashboard
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
< 1 >

▼ 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
124.0 B	1	124.0 B

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

Buckets (0) Info

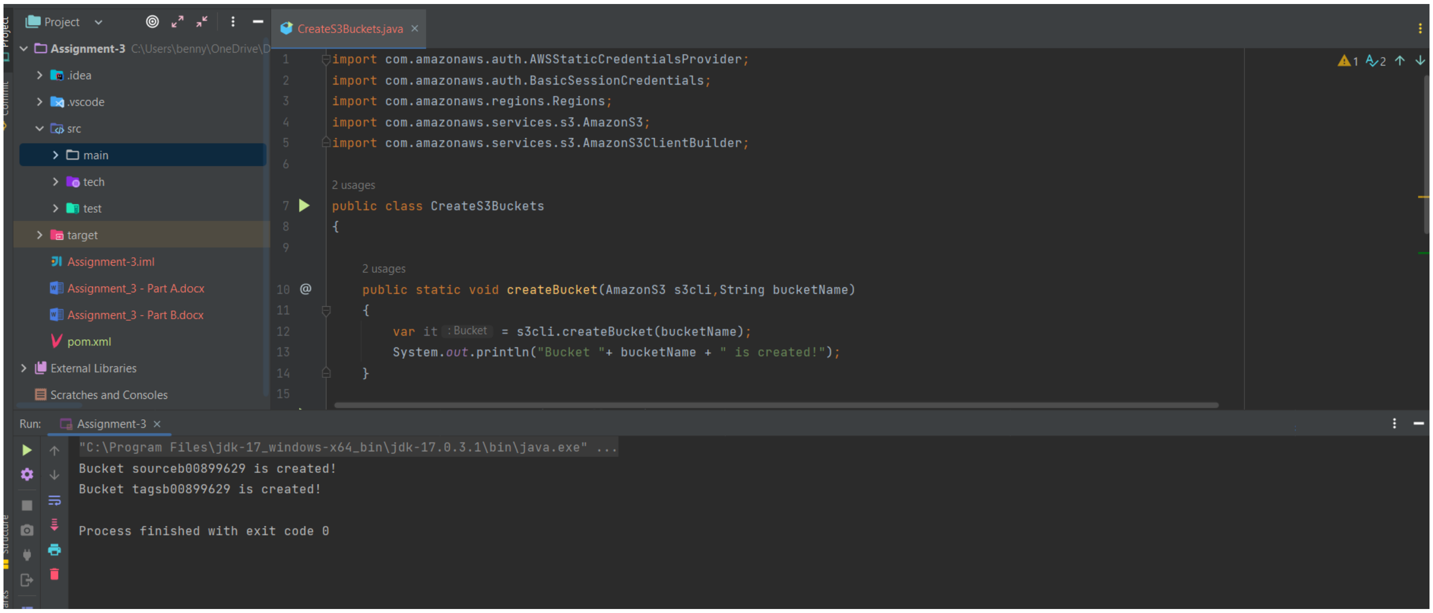
Buckets are containers for data stored in S3. Learn more

Name ▲	AWS Region ▽	Access ▽	Creation date ▽

No buckets

You don't have any buckets.

Create bucket



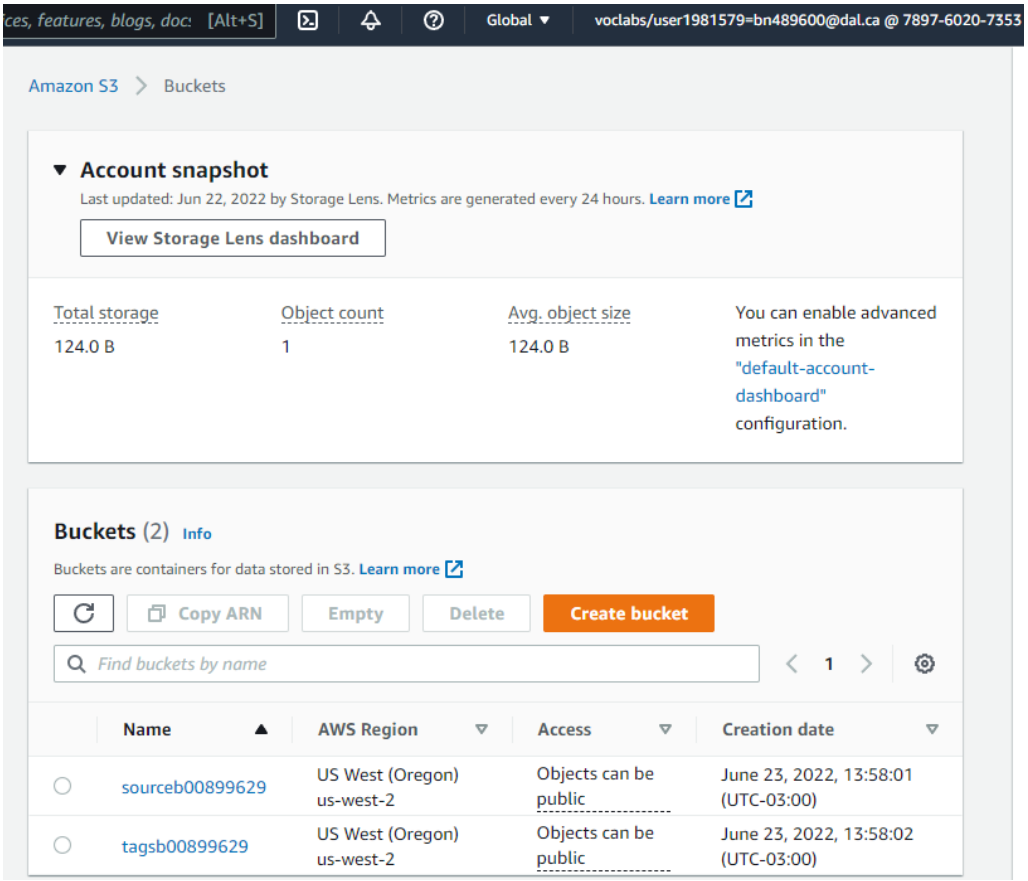
```java
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;

2 usages
public class CreateS3Buckets
{

    2 usages
    public static void createBucket(AmazonS3 s3cli,String bucketName)
    {
        var it : Bucket = s3cli.createBucket(bucketName);
        System.out.println("Bucket "+ bucketName + " is created!");
    }
```

```
"C:\Program Files\jdk-17_windows-x64_bin\jdk-17.0.3.1\bin\java.exe" ...
Bucket sourceb00899629 is created!
Bucket tagsb00899629 is created!

Process finished with exit code 0
```


ces, features, blogs, doc: [Alt+S]
Global ▼
voclabs/user1981579=bn489600@dal.ca @ 7897-6020-7353
Amazon S3 > Buckets
View Storage Lens dashboard
Info
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
< 1 >

▼ 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	You can enable advanced metrics in the "default-account-dashboard" configuration.
124.0 B	1	124.0 B	

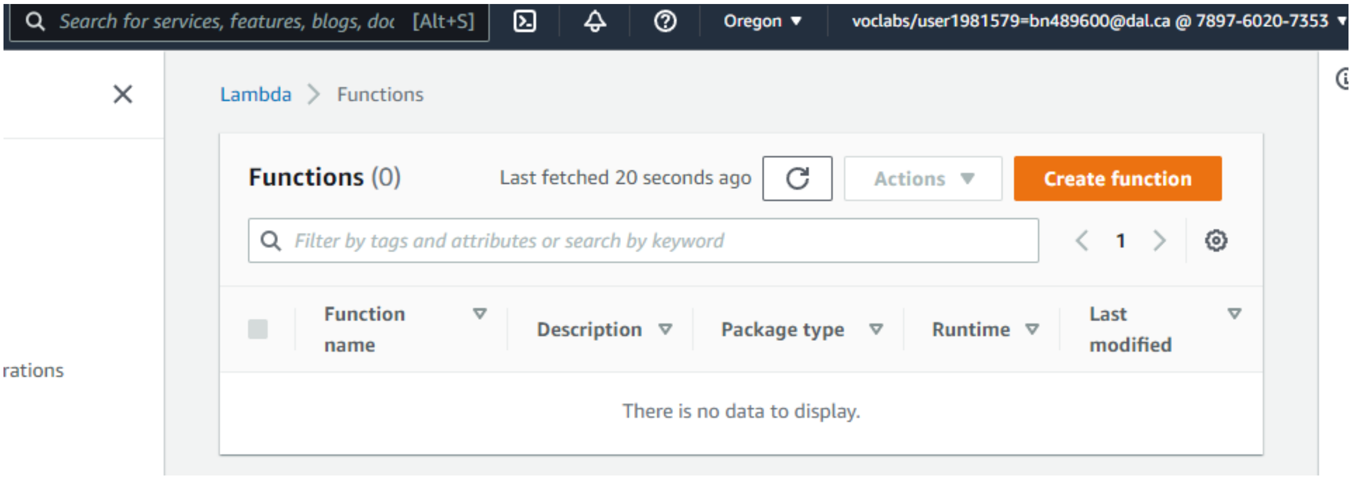
Buckets (2)

Buckets are containers for data stored in S3. Learn more

	Name ▲	AWS Region	Access	Creation date
○	sourceb00899629	US West (Oregon) us-west-2	Objects can be public	June 23, 2022, 13:58:01 (UTC-03:00)
○	tagsb00899629	US West (Oregon) us-west-2	Objects can be public	June 23, 2022, 13:58:02 (UTC-03:00)

Lambda Functions:

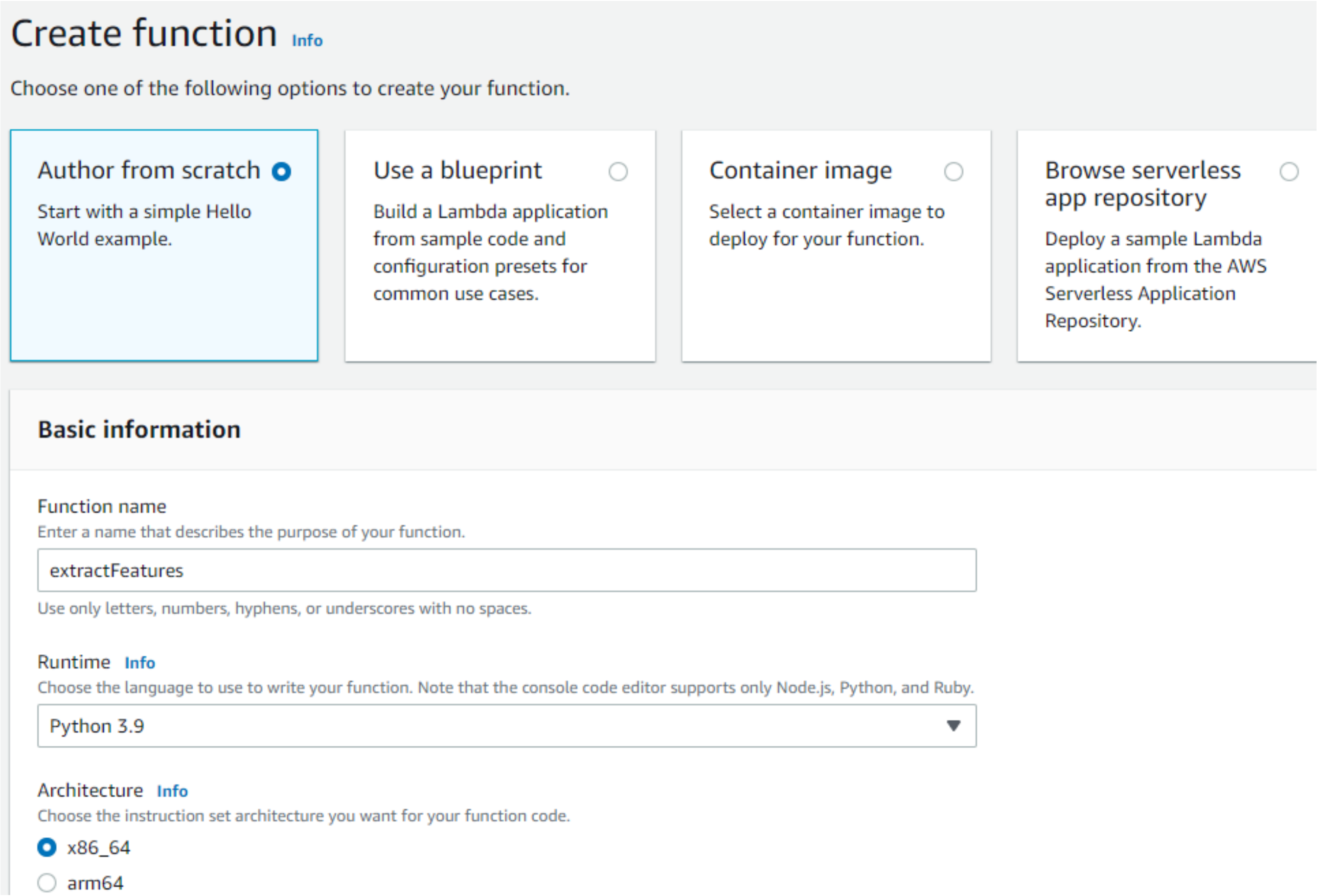
Function – “extractFeatures”:


Search for services, features, blogs, doc [Alt+S]
Oregon ▼
voclabs/user1981579=bn489600@dal.ca @ 7897-6020-7353 ▼
Lambda > Functions
Actions ▼
Create function
Filter by tags and attributes or search by keyword
< 1 >
rations

Functions (0) Last fetched 20 seconds ago

□	Function name	Description	Package type	Runtime	Last modified

There is no data to display.



Create function Info

Choose one of the following options to create your function.

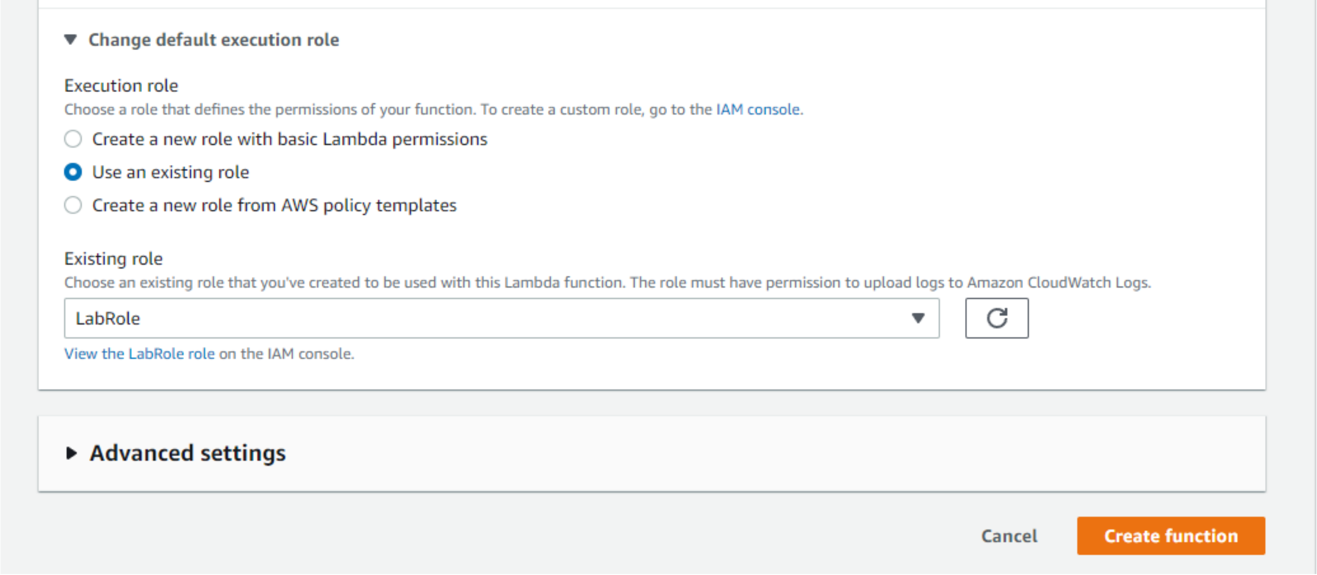
- Author from scratch ● — Start with a simple Hello World example.
- Use a blueprint ○ — Build a Lambda application from sample code and configuration presets for common use cases.
- Container image ○ — Select a container image to deploy for your function.
- Browse serverless app repository ○ — Deploy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

Basic information

Function name
Enter a name that describes the purpose of your function.
extractFeatures
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.
Python 3.9 ▼

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64



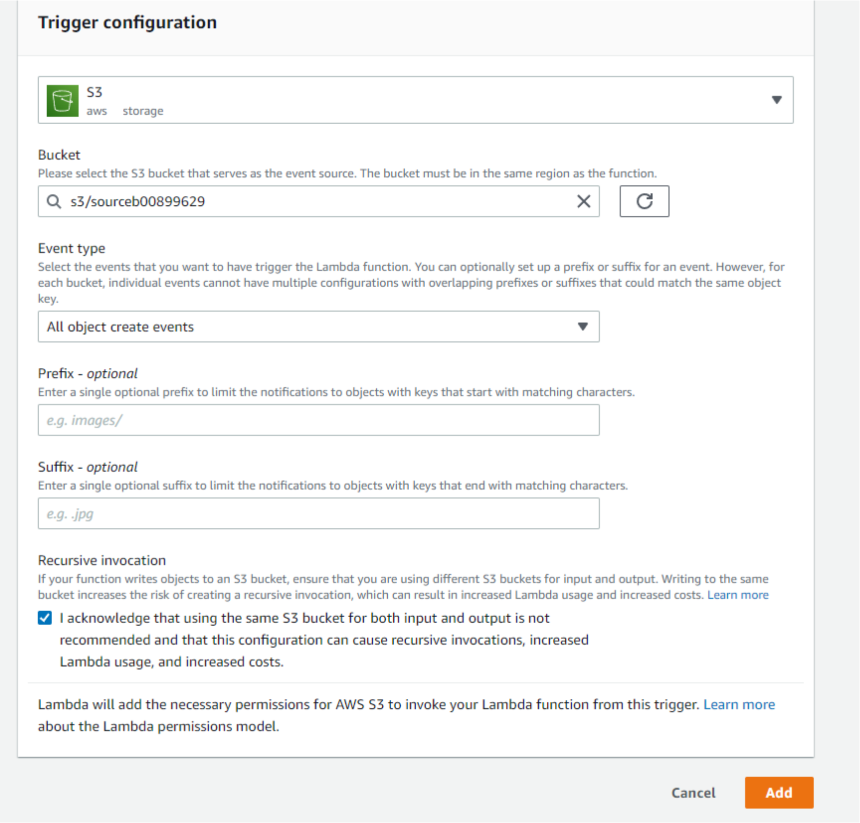
▼ **Change default execution role**

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



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.
s3/sourceb00899629

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

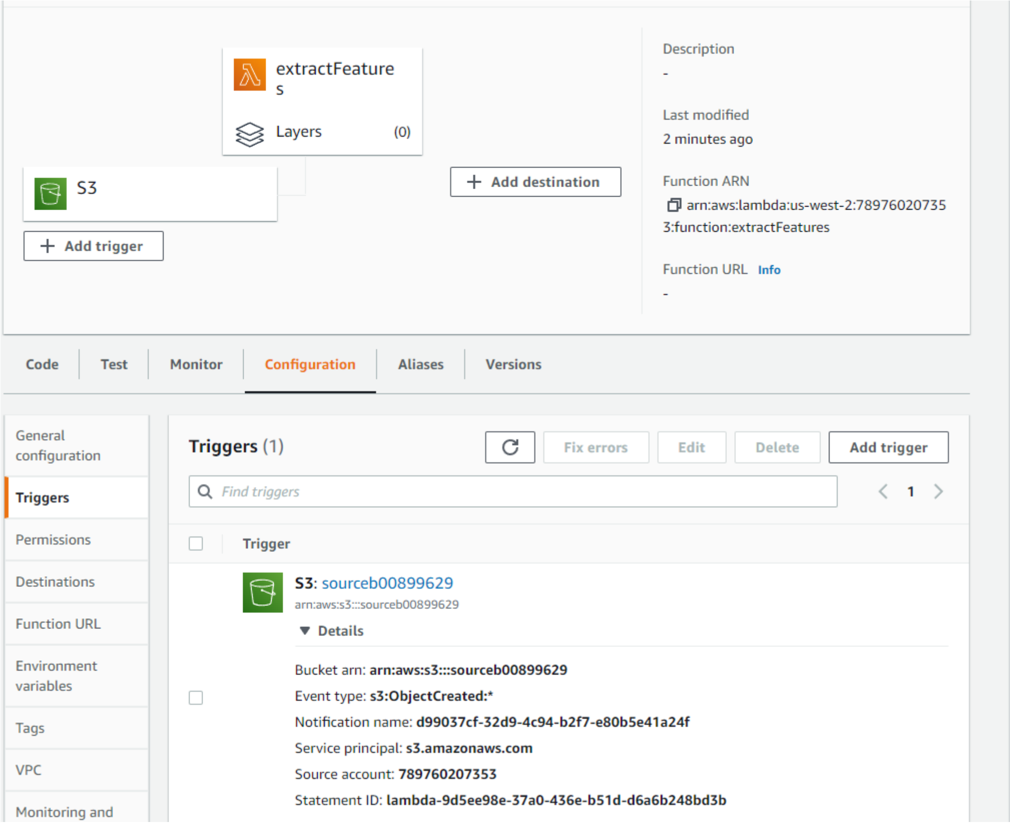
Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/

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

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.

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



Description
-

Last modified
2 minutes ago

Function ARN
arn:aws:lambda:us-west-2:789760207353:function:extractFeatures

Function URL Info
-

Triggers (1)

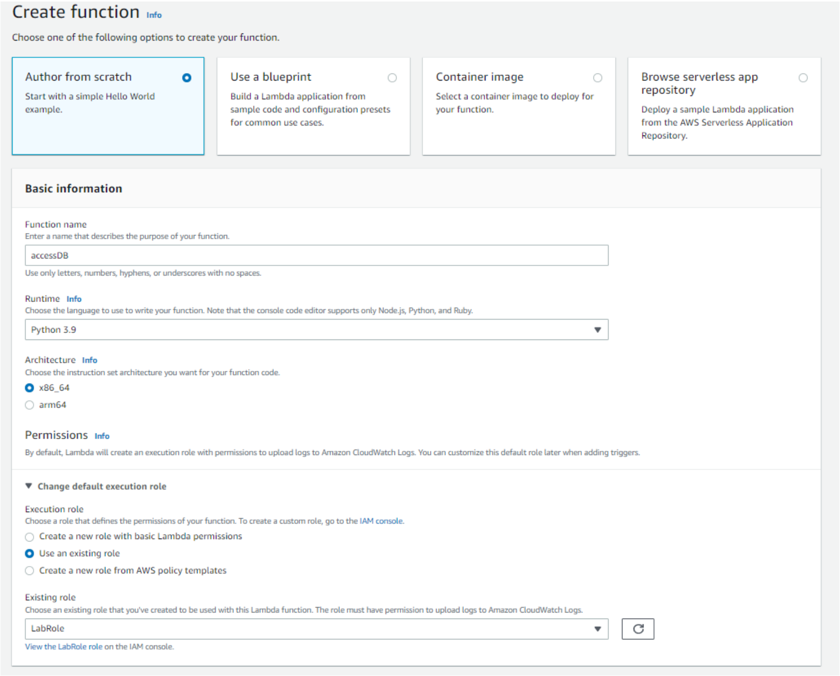
Trigger

S3: sourceb00899629
arn:aws:s3:::sourceb00899629

▼ Details

Bucket arn: **arn:aws:s3:::sourceb00899629**
Event type: **s3:ObjectCreated:***
Notification name: **d99037cf-32d9-4c94-b2f7-e80b5e41a24f**
Service principal: **s3.amazonaws.com**
Source account: **789760207353**
Statement ID: **lambda-9d5ee98e-37a0-436e-b51d-d6a6b248bd3b**

Function – "accessDB":



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.
accessDB
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.
Python 3.9

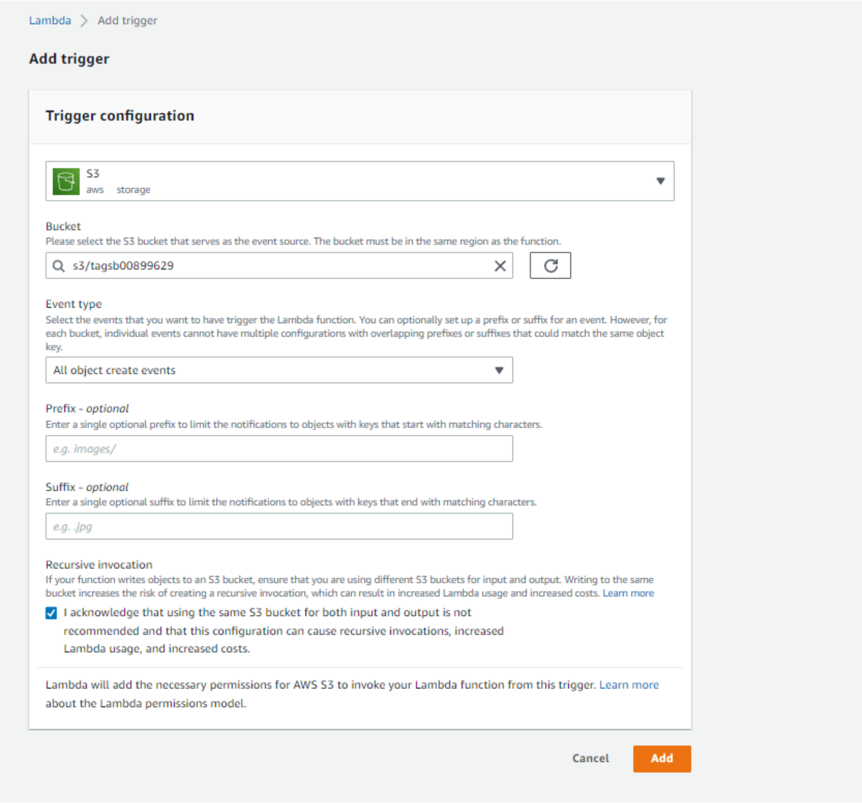
Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

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.



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.
s3/tagsb00899629

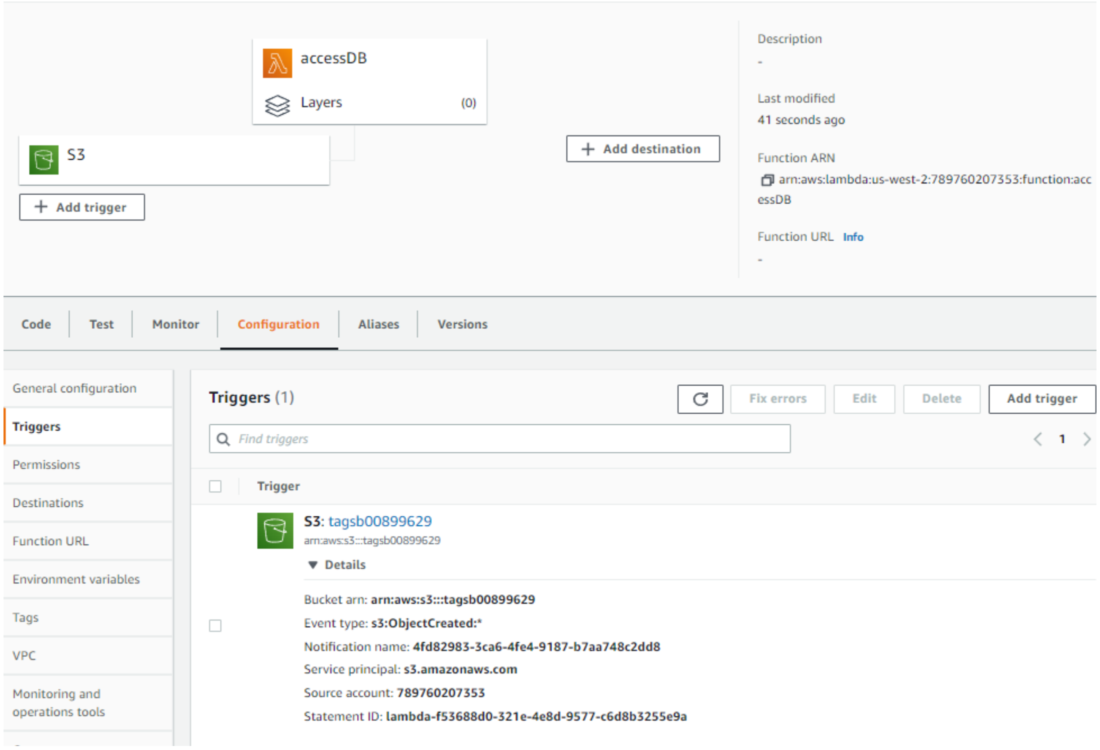
Event type
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
All object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

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.

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



Description
-

Last modified
41 seconds ago

Function ARN
arn:aws:lambda:us-west-2:789760207353:function:accessDB

Function URL Info
-

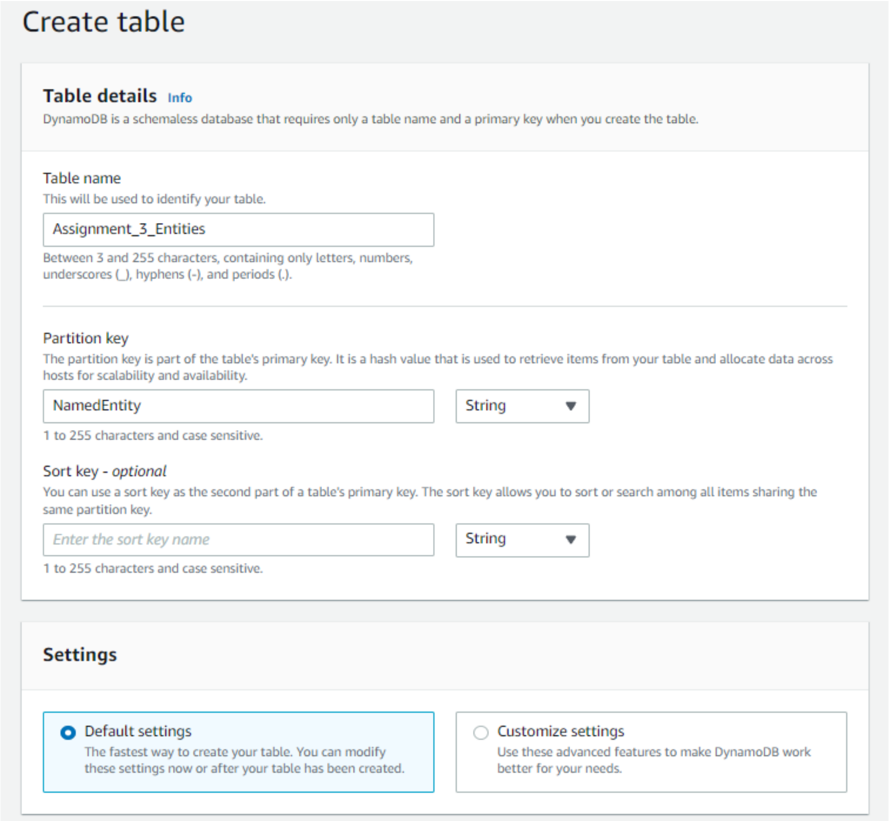
Triggers (1)

□ Trigger

□ S3: tagsb00899629
arn:aws:s3:::tagsb00899629
▼ Details

Bucket arn: **arn:aws:s3:::tagsb00899629**
Event type: **s3:ObjectCreated:***
Notification name: **4fd82983-3ca6-4fe4-9187-b7aa748c2dd8**
Service principal: **s3.amazonaws.com**
Source account: **789760207353**
Statement ID: **lambda-f53688d0-321e-4e8d-9577-c6d8b3255e9a**

DynamoDB Table:



Create table

Table details Info
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.
Assignment_3_Entities
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

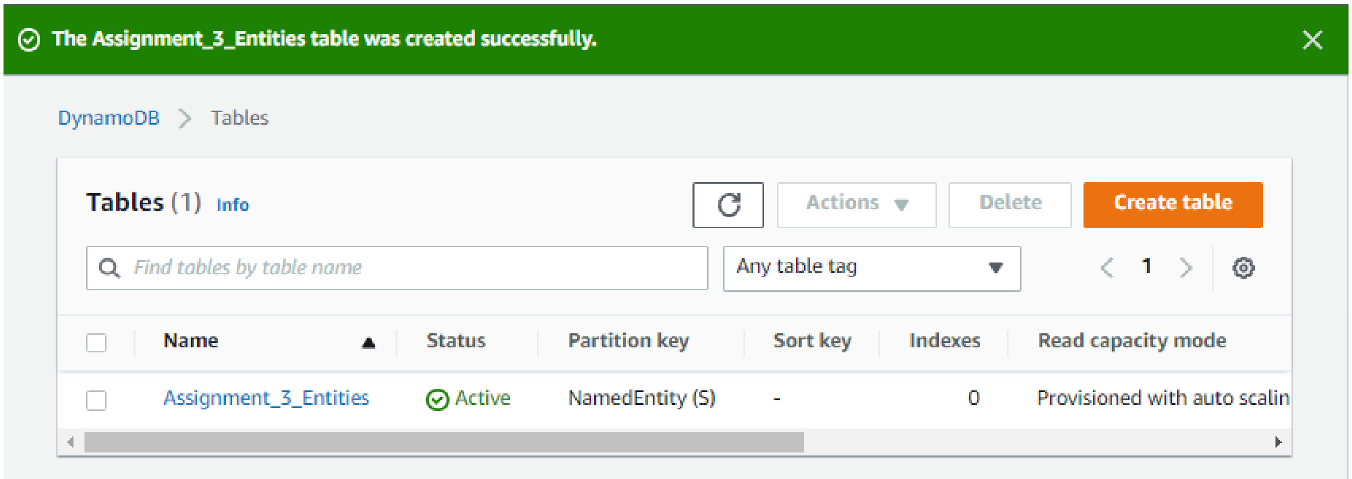
Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
NamedEntity String ▼
1 to 255 characters and case sensitive.

Sort key - *optional*
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name String ▼
1 to 255 characters and case sensitive.

Settings

● Default settings
The fastest way to create your table. You can modify these settings now or after your table has been created.

○ Customize settings
Use these advanced features to make DynamoDB work better for your needs.



The Assignment_3_Entities table was created successfully.

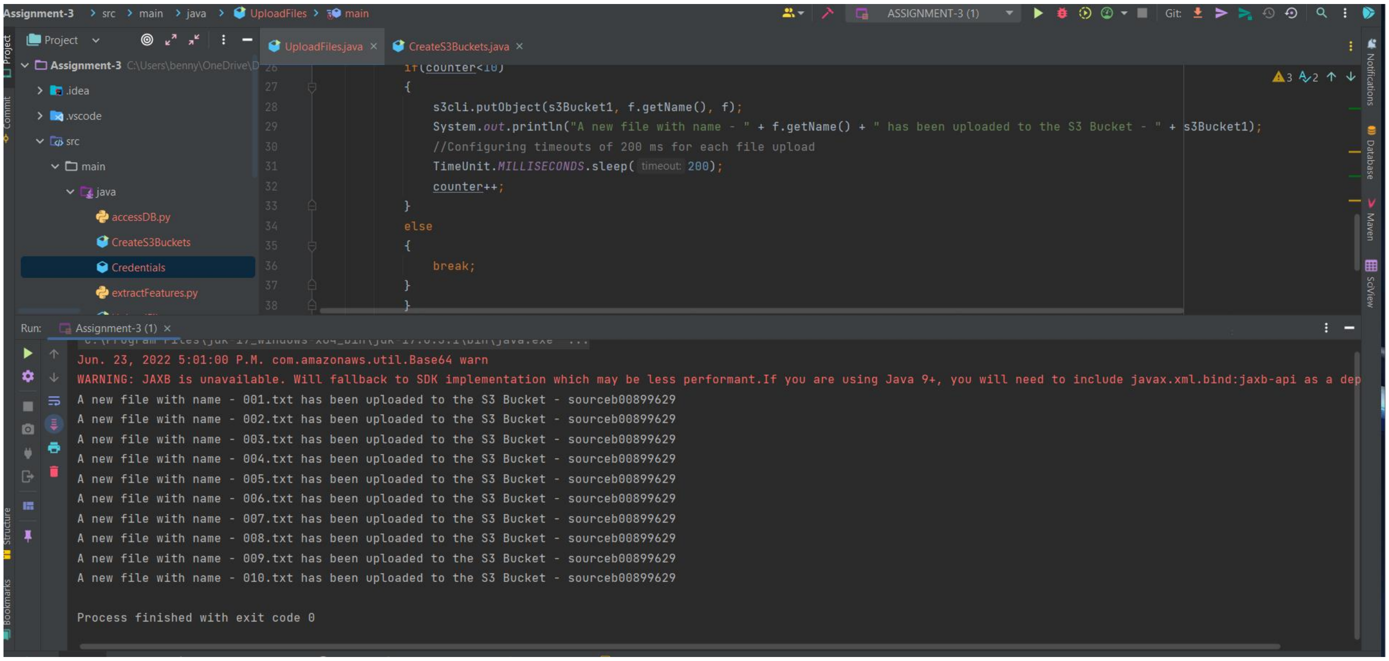
DynamoDB > Tables

Tables (1) Info

☐	Name ▲	Status	Partition key	Sort key	Indexes	Read capacity mode
☐	Assignment_3_Entities	✓ Active	NamedEntity (S)	-	0	Provisioned with auto scalin

Snips of Operations in Chronological Order:

1. Upload 10 files to the Source S3 Bucket

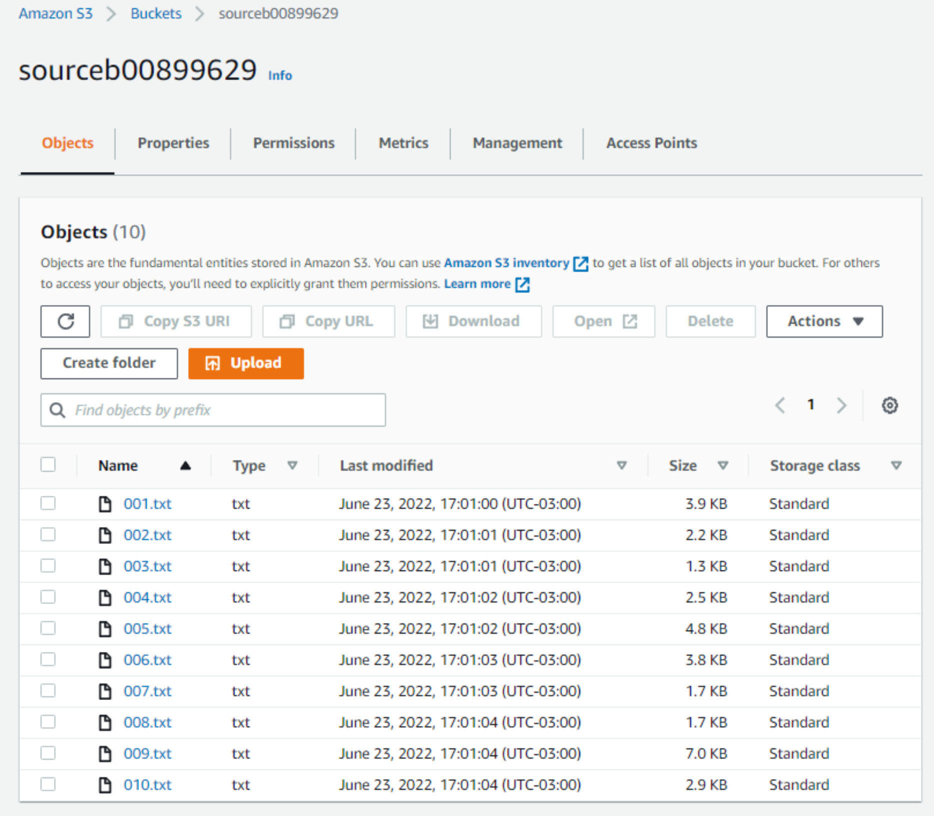


```java
if(counter<10)
{
    s3cli.putObject(s3Bucket1, f.getName(), f);
    System.out.println("A new file with name - " + f.getName() + " has been uploaded to the S3 Bucket - " + s3Bucket1);
    //Configuring timeouts of 200 ms for each file upload
    TimeUnit.MILLISECONDS.sleep( timeout: 200);
    counter++;
}
else
{
    break;
}
}
```

```
Jun. 23, 2022 5:01:00 P.M. com.amazonaws.util.Base64 warn
WARNING: JAXB is unavailable. Will fallback to SDK implementation which may be less performant.If you are using Java 9+, you will need to include javax.xml.bind:jaxb-api as a dep
A new file with name - 001.txt has been uploaded to the S3 Bucket - sourceb00899629
A new file with name - 002.txt has been uploaded to the S3 Bucket - sourceb00899629
A new file with name - 003.txt has been uploaded to the S3 Bucket - sourceb00899629
A new file with name - 004.txt has been uploaded to the S3 Bucket - sourceb00899629
A new file with name - 005.txt has been uploaded to the S3 Bucket - sourceb00899629
A new file with name - 006.txt has been uploaded to the S3 Bucket - sourceb00899629
A new file with name - 007.txt has been uploaded to the S3 Bucket - sourceb00899629
A new file with name - 008.txt has been uploaded to the S3 Bucket - sourceb00899629
A new file with name - 009.txt has been uploaded to the S3 Bucket - sourceb00899629
A new file with name - 010.txt has been uploaded to the S3 Bucket - sourceb00899629

Process finished with exit code 0
```

2. Text Files in the Source S3 Bucket

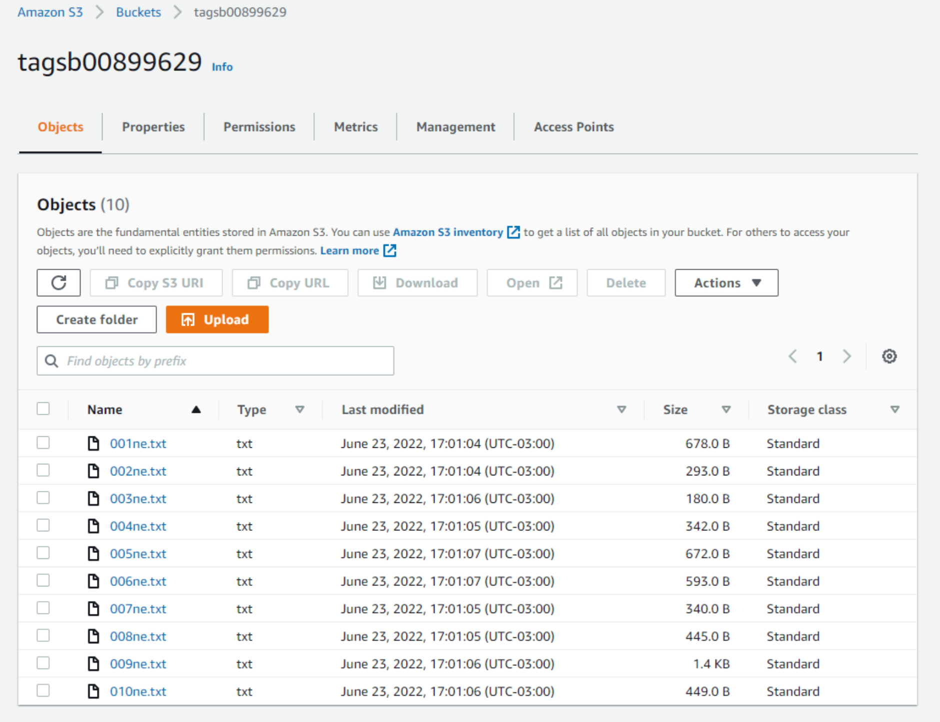


Objects (10)

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, 17:01:00 (UTC-03:00)	3.9 KB	Standard
□	002.txt	txt	June 23, 2022, 17:01:01 (UTC-03:00)	2.2 KB	Standard
□	003.txt	txt	June 23, 2022, 17:01:01 (UTC-03:00)	1.3 KB	Standard
□	004.txt	txt	June 23, 2022, 17:01:02 (UTC-03:00)	2.5 KB	Standard
□	005.txt	txt	June 23, 2022, 17:01:02 (UTC-03:00)	4.8 KB	Standard
□	006.txt	txt	June 23, 2022, 17:01:03 (UTC-03:00)	3.8 KB	Standard
□	007.txt	txt	June 23, 2022, 17:01:03 (UTC-03:00)	1.7 KB	Standard
□	008.txt	txt	June 23, 2022, 17:01:04 (UTC-03:00)	1.7 KB	Standard
□	009.txt	txt	June 23, 2022, 17:01:04 (UTC-03:00)	7.0 KB	Standard
□	010.txt	txt	June 23, 2022, 17:01:04 (UTC-03:00)	2.9 KB	Standard

3. Named Entity Files in the Target (or) “tags” S3 Bucket

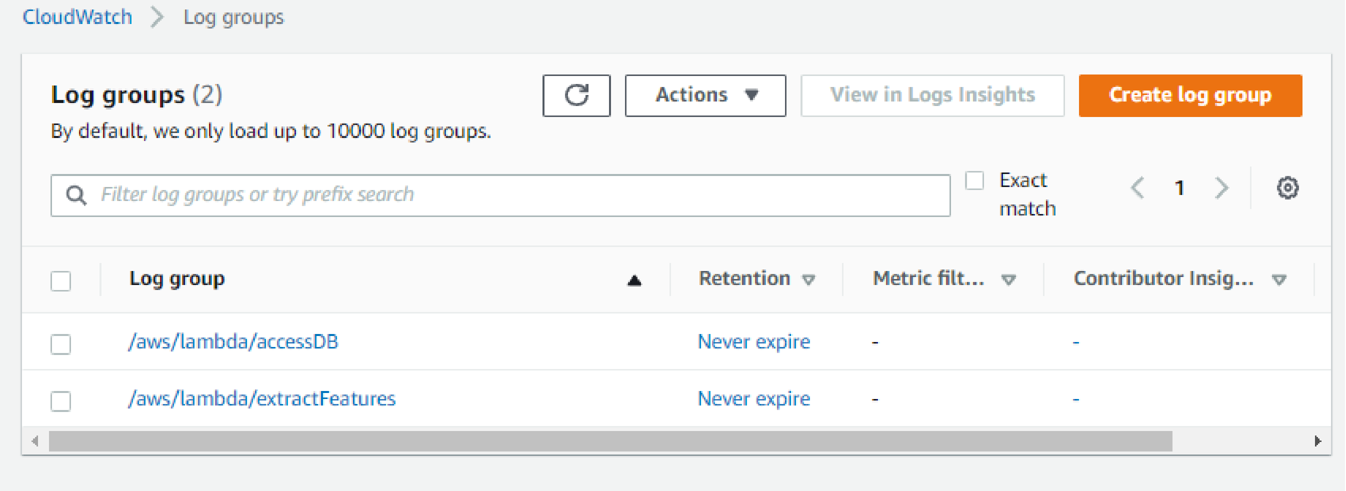


Objects (10)

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, 17:01:04 (UTC-03:00)	678.0 B	Standard
□	002ne.txt	txt	June 23, 2022, 17:01:04 (UTC-03:00)	293.0 B	Standard
□	003ne.txt	txt	June 23, 2022, 17:01:06 (UTC-03:00)	180.0 B	Standard
□	004ne.txt	txt	June 23, 2022, 17:01:05 (UTC-03:00)	342.0 B	Standard
□	005ne.txt	txt	June 23, 2022, 17:01:07 (UTC-03:00)	672.0 B	Standard
□	006ne.txt	txt	June 23, 2022, 17:01:07 (UTC-03:00)	593.0 B	Standard
□	007ne.txt	txt	June 23, 2022, 17:01:05 (UTC-03:00)	340.0 B	Standard
□	008ne.txt	txt	June 23, 2022, 17:01:05 (UTC-03:00)	445.0 B	Standard
□	009ne.txt	txt	June 23, 2022, 17:01:06 (UTC-03:00)	1.4 KB	Standard
□	010ne.txt	txt	June 23, 2022, 17:01:06 (UTC-03:00)	449.0 B	Standard

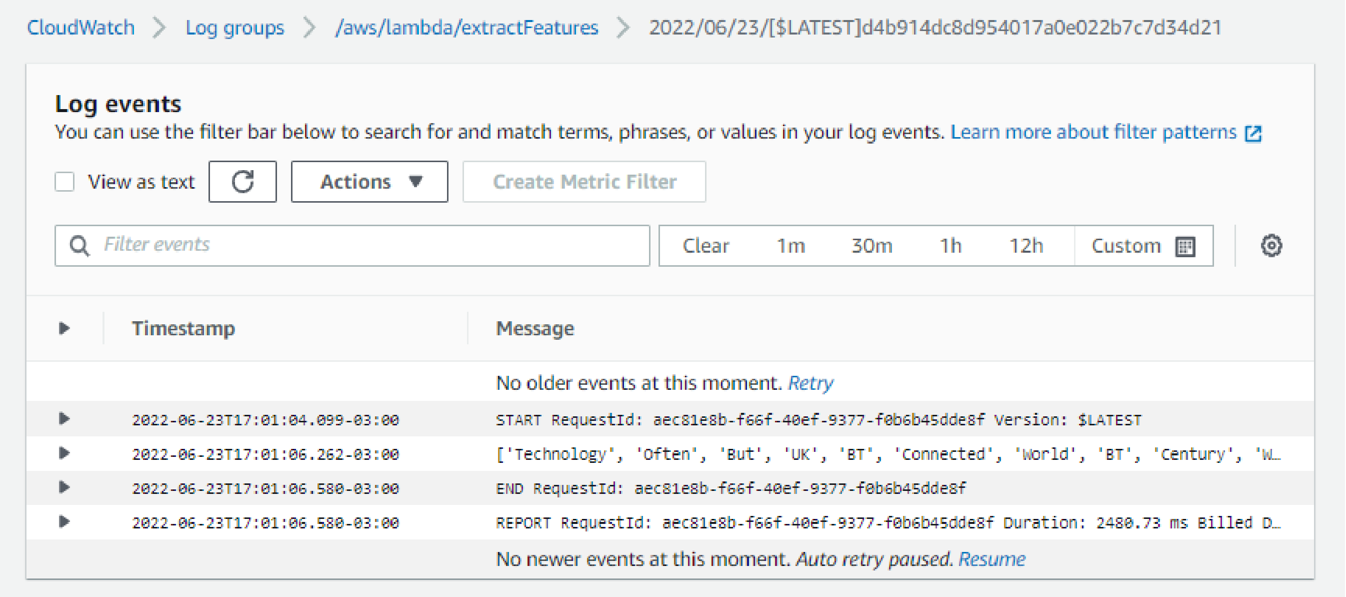
4. CloudWatch Log for the File Uploads



Log groups (2)

By default, we only load up to 10000 log groups.

□	Log group ▲	Retention ▽	Metric filt... ▽	Contributor Insig... ▽
□	/aws/lambda/accessDB	Never expire	-	-
□	/aws/lambda/extractFeatures	Never expire	-	-



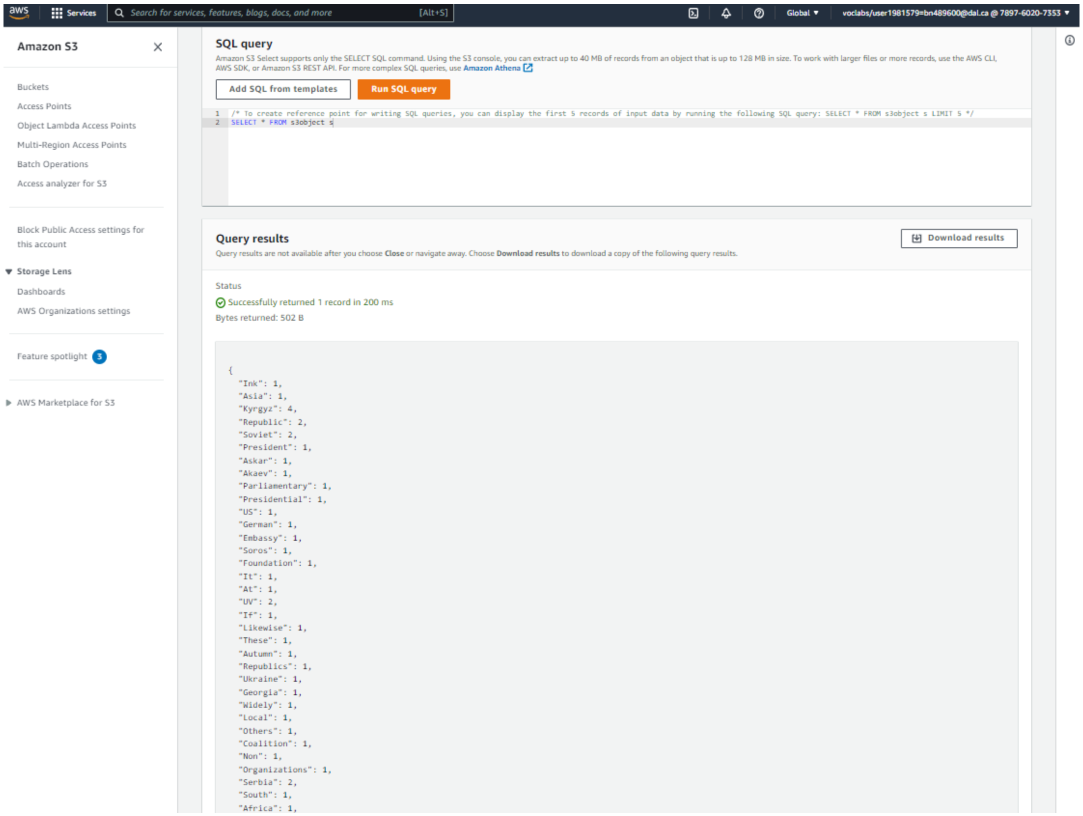
CloudWatch > Log groups > /aws/lambda/extractFeatures > 2022/06/23/[$LATEST]d4b914dc8d954017a0e022b7c7d34d21
□ View as text
Actions ▼
Create Metric Filter
Filter events
Clear
1m
30m
1h
12h
Custom

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
2022-06-23T17:01:04.099-03:00    START RequestId: aec81e8b-f66f-40ef-9377-f0b6b45dde8f Version: $LATEST
2022-06-23T17:01:06.262-03:00    ['Technology', 'Often', 'But', 'UK', 'BT', 'Connected', 'World', 'BT', 'Century', 'W…
2022-06-23T17:01:06.580-03:00    END RequestId: aec81e8b-f66f-40ef-9377-f0b6b45dde8f
2022-06-23T17:01:06.580-03:00    REPORT RequestId: aec81e8b-f66f-40ef-9377-f0b6b45dde8f Duration: 2480.73 ms Billed D…
                                 No newer events at this moment. Auto retry paused. Resume
```

5. Named Entities and their Frequencies in each file under the “tags” S3 Bucket


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▾
voclabs/user1981579=bn489600@dal.ca @ 7897-6020-7353 ▾
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
Add SQL from templates
Run SQL query
Download results

SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use Amazon Athena

```
/* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL query: SELECT * FROM s3object s LIMIT 5 */
SELECT * FROM s3object s
```

Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

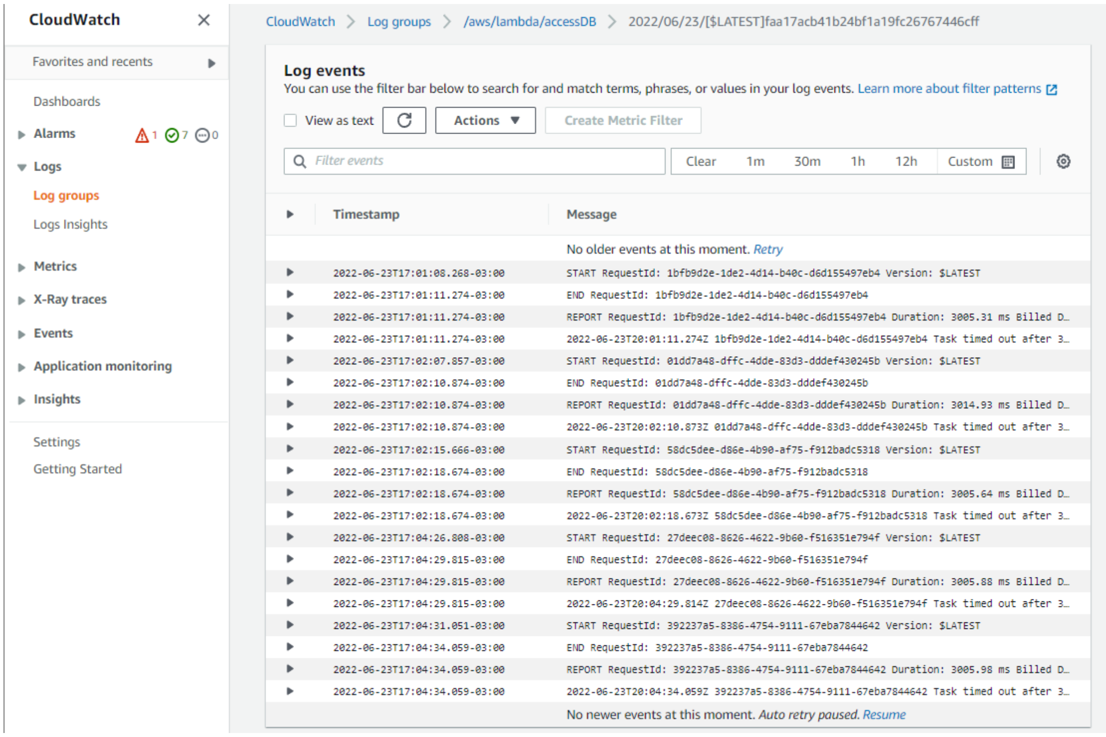
Status

Successfully returned 1 record in 200 ms

Bytes returned: 502 B

```
{
  "Ink": 1,
  "Asia": 1,
  "Kyrgyz": 4,
  "Republic": 2,
  "Soviet": 2,
  "President": 1,
  "Askar": 1,
  "Akaev": 1,
  "Parliamentary": 1,
  "Presidential": 1,
  "US": 1,
  "German": 1,
  "Embassy": 1,
  "Soros": 1,
  "Foundation": 1,
  "It": 1,
  "At": 1,
  "UV": 2,
  "If": 1,
  "Likewise": 1,
  "These": 1,
  "Autumn": 1,
  "Republics": 1,
  "Ukraine": 1,
  "Georgia": 1,
  "Widely": 1,
  "Local": 1,
  "Others": 1,
  "Coalition": 1,
  "Non": 1,
  "Organizations": 1,
  "Serbia": 2,
  "South": 1,
  "Africa": 1,
```

6. CloudWatch Log for the File Uploads



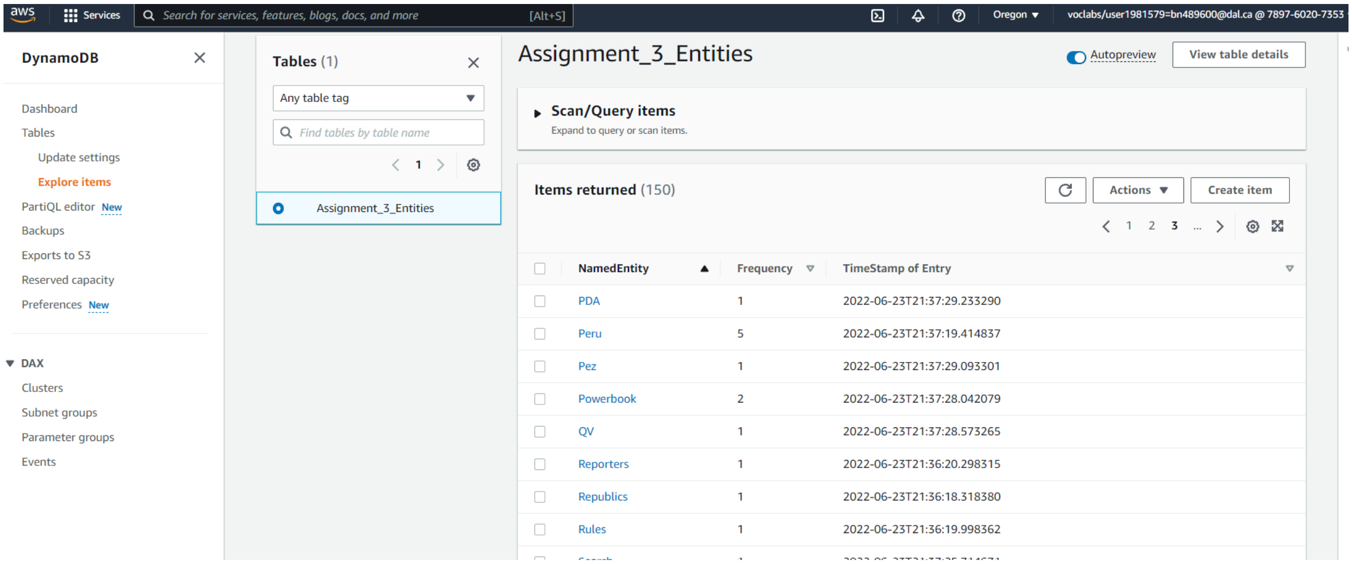
CloudWatch > Log groups > /aws/lambda/accessDB > 2022/06/23/[$LATEST]faa17acb41b24bf1a19fc26767446cff

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
2022-06-23T17:01:08.268-03:00    START RequestId: 1bfb9d2e-1de2-4d14-b40c-d6d155497eb4 Version: $LATEST
2022-06-23T17:01:11.274-03:00    END RequestId: 1bfb9d2e-1de2-4d14-b40c-d6d155497eb4
2022-06-23T17:01:11.274-03:00    REPORT RequestId: 1bfb9d2e-1de2-4d14-b40c-d6d155497eb4 Duration: 3005.31 ms Billed D…
2022-06-23T17:01:11.274-03:00    2022-06-23T20:01:11.274Z 1bfb9d2e-1de2-4d14-b40c-d6d155497eb4 Task timed out after 3…
2022-06-23T17:02:07.857-03:00    START RequestId: 01dd7a48-dffc-4dde-83d3-dddef430245b Version: $LATEST
2022-06-23T17:02:10.874-03:00    END RequestId: 01dd7a48-dffc-4dde-83d3-dddef430245b
2022-06-23T17:02:10.874-03:00    REPORT RequestId: 01dd7a48-dffc-4dde-83d3-dddef430245b Duration: 3014.93 ms Billed D…
2022-06-23T17:02:10.874-03:00    2022-06-23T20:02:10.873Z 01dd7a48-dffc-4dde-83d3-dddef430245b Task timed out after 3…
2022-06-23T17:02:15.666-03:00    START RequestId: 58dc5dee-d86e-4b90-af75-f912badc5318 Version: $LATEST
2022-06-23T17:02:18.674-03:00    END RequestId: 58dc5dee-d86e-4b90-af75-f912badc5318
2022-06-23T17:02:18.674-03:00    REPORT RequestId: 58dc5dee-d86e-4b90-af75-f912badc5318 Duration: 3005.64 ms Billed D…
2022-06-23T17:02:18.674-03:00    2022-06-23T20:02:18.673Z 58dc5dee-d86e-4b90-af75-f912badc5318 Task timed out after 3…
2022-06-23T17:04:26.808-03:00    START RequestId: 27deec08-8626-4622-9b60-f516351e794f Version: $LATEST
2022-06-23T17:04:29.815-03:00    END RequestId: 27deec08-8626-4622-9b60-f516351e794f
2022-06-23T17:04:29.815-03:00    REPORT RequestId: 27deec08-8626-4622-9b60-f516351e794f Duration: 3005.88 ms Billed D…
2022-06-23T17:04:29.815-03:00    2022-06-23T20:04:29.814Z 27deec08-8626-4622-9b60-f516351e794f Task timed out after 3…
2022-06-23T17:04:31.051-03:00    START RequestId: 392237a5-8386-4754-9111-67eba7844642 Version: $LATEST
2022-06-23T17:04:34.059-03:00    END RequestId: 392237a5-8386-4754-9111-67eba7844642
2022-06-23T17:04:34.059-03:00    REPORT RequestId: 392237a5-8386-4754-9111-67eba7844642 Duration: 3005.98 ms Billed D…
2022-06-23T17:04:34.059-03:00    2022-06-23T20:04:34.059Z 392237a5-8386-4754-9111-67eba7844642 Task timed out after 3…
                                 No newer events at this moment. Auto retry paused. Resume
```

7. Entities, Frequencies and Timestamp in the DynamoDB table



Assignment_3_Entities

Items returned (150)

NamedEntity	Frequency	TimeStamp of Entry
PDA	1	2022-06-23T21:37:29.233290
Peru	5	2022-06-23T21:37:19.414837
Pez	1	2022-06-23T21:37:29.093301
Powerbook	2	2022-06-23T21:37:28.042079
QV	1	2022-06-23T21:37:28.573265
Reporters	1	2022-06-23T21:36:20.298315
Republics	1	2022-06-23T21:36:18.318380
Rules	1	2022-06-23T21:36:19.998362

Code Blocks

CreateS3Bucket.java [1-4]

```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;

public class CreateS3Buckets
{

    public static void createBucket(AmazonS3 s3cli,String bucketName)
    {
        var it = s3cli.createBucket(bucketName);
    }

    public static void main(String[] args)
    {
        String id = Credentials.aws_access_key_id;
        String key = Credentials.aws_secret_access_key;
        String token = Credentials.aws_session_token;
        BasicSessionCredentials sessionCredentials = new
BasicSessionCredentials(id,key,token);
        AmazonS3 s3cli = AmazonS3ClientBuilder.standard().withCredentials(new
AWSStaticCredentialsProvider(sessionCredentials)).withRegion(Regions.DEFAULT_REGION)
.build();
        createBucket( s3cli,"sourceb00899629");
        createBucket( s3cli,"tagsb00899629");
    }
}
```

UploadFiles.java

```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import java.io.File;
import java.util.concurrent.TimeUnit;


public class UploadFiles
{
    public static void main(String[] args) throws InterruptedException
    {
        CreateS3Buckets obj = new CreateS3Buckets();
        String id = Credentials.aws_access_key_id;
        String key = Credentials.aws_secret_access_key;
        String token = Credentials.aws_session_token;
        BasicSessionCredentials sessionCredentials = new BasicSessionCredentials(id,key,token);
        AmazonS3 s3cli = AmazonS3ClientBuilder.standard().withCredentials(new AWSStaticCredentialsProvider(sessionCredentials)).withRegion(Regions.US_WEST_2).build();
        String s3Bucket1 = "sourceb00899629";
        File folderPath = new File("./src/tech");
        File[] allFiles = folderPath.listFiles();
        int counter = 0; //Initialize a variable such that only 10 files from the folder - 'tech" are uploaded to the S3 Bucket
        for (File f : allFiles)
        {
            if(counter<10)
            {
                s3cli.putObject(s3Bucket1, f.getName(), f);
                System.out.println("A new file with name - " + f.getName() + " has been uploaded to the S3 Bucket - " + s3Bucket1);
                //Configuring timeouts of 200 ms for each file upload
                TimeUnit.MILLISECONDS.sleep(200);
                counter++;
            }
            else
            {
                break;
            }
            }
        }
    }
```

extractEntities.py [5-6,9]

```
import urllib.parse
import re
import json
import boto3

def lambda_handler(event, context):

    s3cli = boto3.client("s3")
    s3Upload = boto3.resource('s3')
    Upload_bucket = "tagsb00899629"
    dictionary = {}
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'],
encoding='utf-8')
    json_file_name = key.split('.')
    file_content = s3cli.get_object(Bucket=bucket_name,
Key=key)['Body'].read().decode("utf-8")
    file_content_main = file_content.replace('\n',' ');
    all_words = file_content_main.split(" ")
    entities = re.sub(r"\bThe\b", '',str(all_words))
    entities = re.sub(r"\bThis\b", '',entities)
    entities = re.sub(r"\bThen\b", '',entities)
    entities = re.sub(r"\bIt\b", '',entities)
    entities = re.sub(r"\bAt\b", '',entities)
    entities = re.sub(r"\bAnd\b", '',entities)
    entities = re.sub(r"\bIn\b", '',entities)
    entities = re.sub(r"\bHowever\b", '',entities)
    entities = re.sub(r"\bIf\b", '',entities)
    entities = re.findall(r'(?<!\.\s)(?!^)\b([A-Z]\w*(?:\s+[A-Z]\w*)*)',entities)
    print(entities)
    for words in entities:
        if words in dictionary:
            dictionary[str(words)] = dictionary[str(words)] + 1
        else:
            dictionary[str(words)] = 1

    final_json = json.dumps(dictionary,indent = 2)
    s3Upload.Object(Upload_bucket, json_file_name[0]+ "ne.txt").put(Body =
(final_json))
```

storeDynamo.py [7-8]

```
import json
import boto3
import urllib.parse
from datetime import datetime


def lambda_handler(event,context):

    dynamoDB_table = boto3.resource('dynamodb').Table('Assignment_3_Entities')
    s3Client = boto3.client("s3")
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'],
encoding='utf-8')
    json_file = s3Client.get_object(Bucket=bucket_name,
Key=key)['Body'].read().decode("utf-8")
    parsed_object = json.loads(json_file)

    for key,value in parsed_object.items():
        dynamoDB_table.put_item(Item={'NamedEntity': key,'Frequency': value, 'TimeStamp
of Entry' : datetime.now().isoformat()})
```

Credentials.java

```
public class Credentials
{
public static String aws_access_key_id="ASIA3PYLANH44TJNTPFI";
public static String
aws_secret_access_key="QcReBN+F0n/dDqBqXbE7HLmmQkhHd7MInmsNP/Mn";
public static String
aws_session_token="FwoGZXIvYXdzEHUaDKSRjb6kVF9lk+/q0iLAAXd9WYWVu0z0W/12qKuUXbkGQMEda
QzgGP80N4U9ww3GCiQXYLSDDba9monVzbIviCi1UutFnPeAhl40FaSVw27Bdb0tayZ2dQA+K53TKZKYM5DYn
+eDh51Tz9tYwjoB7wcDB1vlyXu/TKWw5JfsGR154L93pA0tHod0cKQlVRoCPBCMI8+Hd+0S9QSYAf3vpPY1q
U1Hgmq/8iJC3aCXBCLf9PsUUEoA9w3GbyFQtlkQrXFylLNBFCHwLqSz+OtXAyiMhdOVBjItLbecv2rApPahR
v/JH6eH0gyOY24+iEowQBuDofl7ni/joEv78yMj4+LpFwLz";
}
```

Citations

[1] "AWS SDK for Java." *Amazon Web Services, Inc.*, 15 Sept. 2012, aws.amazon.com/sdk-for-java/. Accessed 21 June 2022.

[2] "Awsdocs/Aws-Doc-Sdk-Examples." *GitHub*, 5 Sept. 2019, github.com/awsdocs/aws-doc-sdk-examples/blob/main/java/example_code/s3/src/main/java/aws/example/s3/CreateBucket.java. Accessed 21 June 2022.

[3] baeldung. “AWS S3 with Java | Baeldung.” *Www.baeldung.com*, 24 July 2017, www.baeldung.com/aws-s3-java. Accessed 22 June 2022.

[4] “Managing Dependencies with AWS SDK for Java – Bill of Materials Module (BOM).” *Amazon Web Services*, 10 Aug. 2015, aws.amazon.com/blogs/developer/managing-dependencies-with-aws-sdk-for-java-bill-of-materials-module-bom. Accessed 22 May 2022.

[5] anubhava. “Python - Regular Expression to Extract Named Entities from Text Just Based on Capitalization.” *Stack Overflow*, 9 May 2017, stackoverflow.com/questions/43972800/regular-expression-to-extract-named-entities-from-text-just-based-on-capitalizat. Accessed 23 June 2022.

[6] AWS. “How to Read Files from S3 Using Python AWS Lambda.” *Gcptutorials*, 9 Mar. 2013, www.gcptutorials.com/post/how-to-read-files-from-s3-using-python-aws-lambda. Accessed 20 June 2022.

[7] “Datetime - ISO Time (ISO 8601) in Python.” *Stack Overflow*, 16 Aug. 2016, stackoverflow.com/questions/2150739/iso-time-iso-8601-in-python. Accessed 21 June 2022.

[8] “Datetime - What Data Type Should Be Used for Timestamp in DynamoDB?” *Stack Overflow*, 3 Dec. 2016, stackoverflow.com/questions/40561484/what-data-type-should-be-used-for-timestamp-in-dynamodb. Accessed 21 June 2022.

[9] “Tutorial: Using an Amazon S3 Trigger to Invoke a Lambda Function - AWS Lambda.” *Docs.aws.amazon.com*, 17 June 2013, docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html. Accessed 20 June 2022.

[10] D. Greene and P. Cunningham, *Derekgreene.com*, 2006. [Online]. Available: http://derekgreene.com/papers/greene06icml.pdf. [Accessed: 21- Jun- 2022].