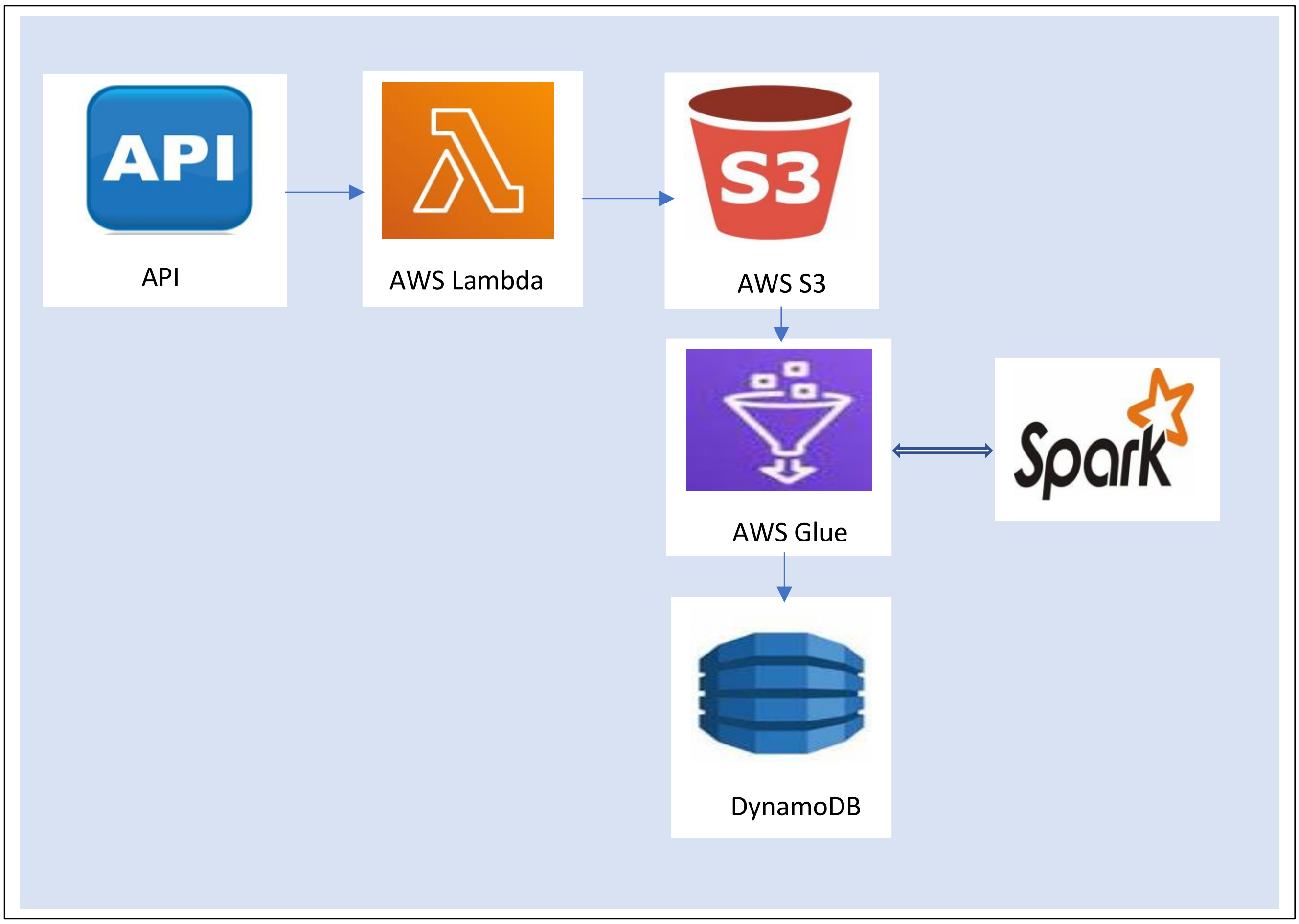
AWS PROJECT Version 2

Project Architecture:

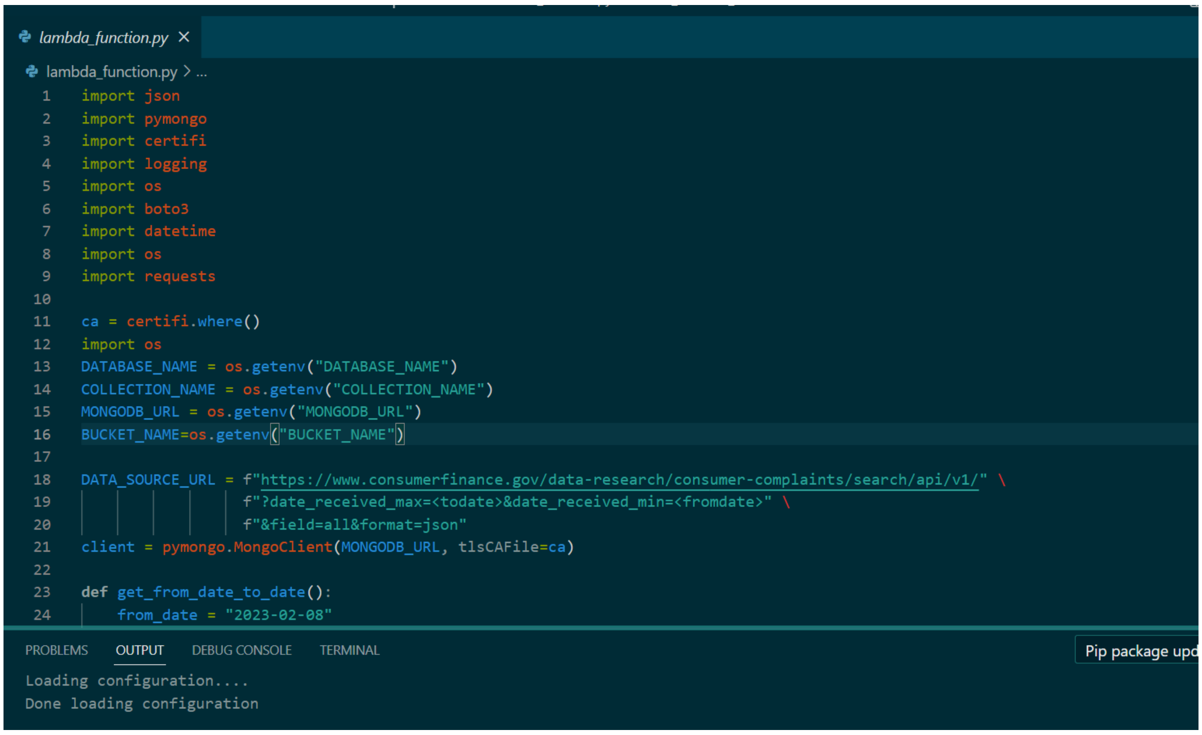


AWS Lambda:

Web API -----> JSON Data FROM_DATE – TO_DATE -----> S3 Bucket

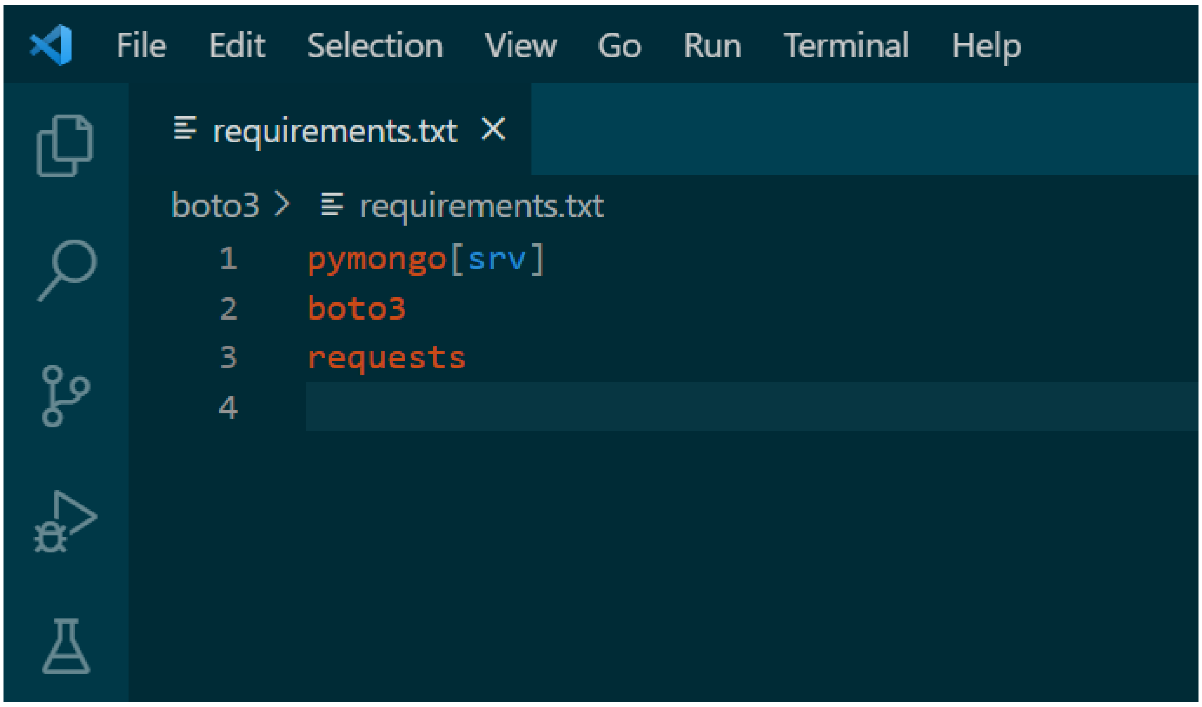
This Lambda function retrieves financial complaint data from the Consumer Finance Protection Bureau (CFPB) API between a specified date range. The date range is determined by checking if there is any previously downloaded data in a MongoDB database and using that to set the from_date, otherwise it defaults to a hardcoded from_date. The data is then filtered and saved in a JSON format in an S3 bucket. Finally, the from_date and to_date are saved in the MongoDB database for future reference.

Implementation for AWS Lambda:



```python
import json
import pymongo
import certifi
import logging
import os
import boto3
import datetime
import os
import requests

ca = certifi.where()
import os
DATABASE_NAME = os.getenv("DATABASE_NAME")
COLLECTION_NAME = os.getenv("COLLECTION_NAME")
MONGODB_URL = os.getenv("MONGODB_URL")
BUCKET_NAME=os.getenv("BUCKET_NAME")

DATA_SOURCE_URL = f"https://www.consumerfinance.gov/data-research/consumer-complaints/search/api/v1/" \
                  f"?date_received_max=<todate>&date_received_min=<fromdate>" \
                  f"&field=all&format=json"
client = pymongo.MongoClient(MONGODB_URL, tlsCAFile=ca)

def get_from_date_to_date():
    from_date = "2023-02-08"
```

```
Loading configuration....
Done loading configuration
```



```
pymongo[srv]
boto3
requests
```

conda create -p venv python==3.9 -y --> to create a new virtual environment using conda

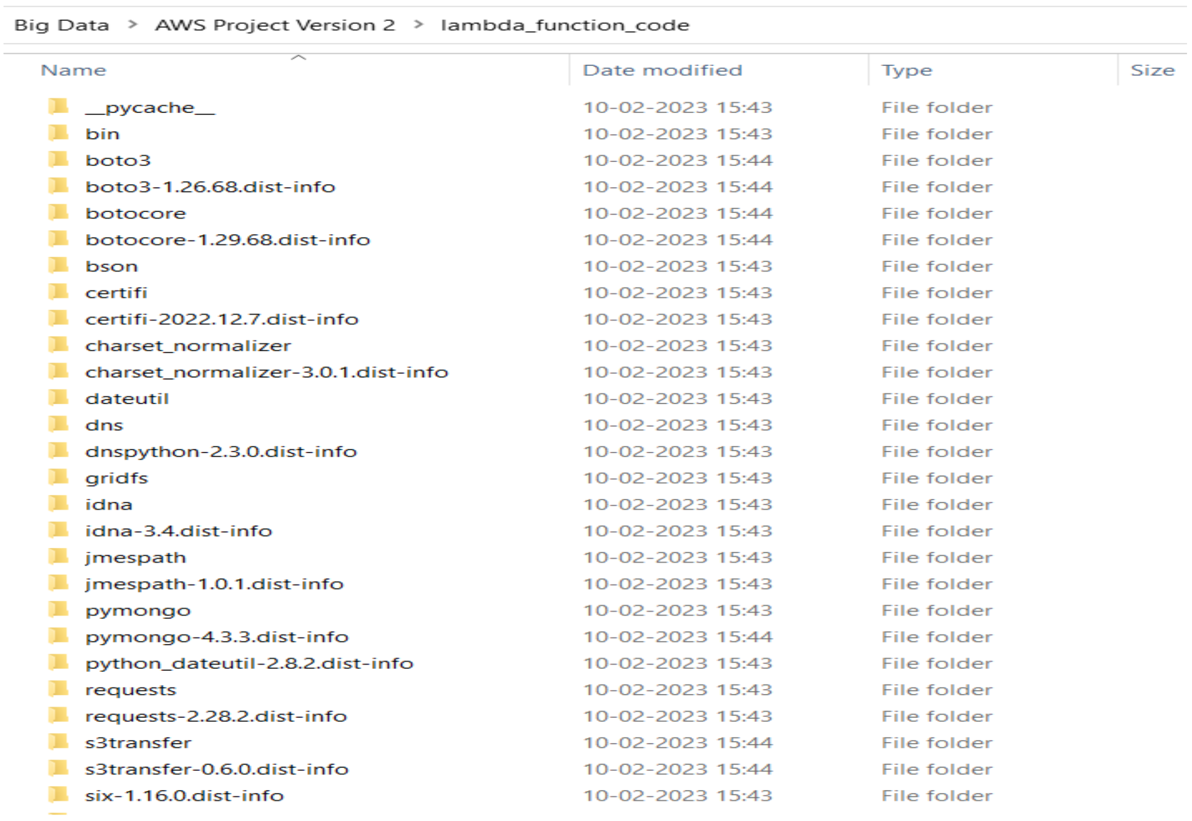
conda activate venv/ --> to activate the virtual environment

pip install -r requirements.txt --> to install the required packages

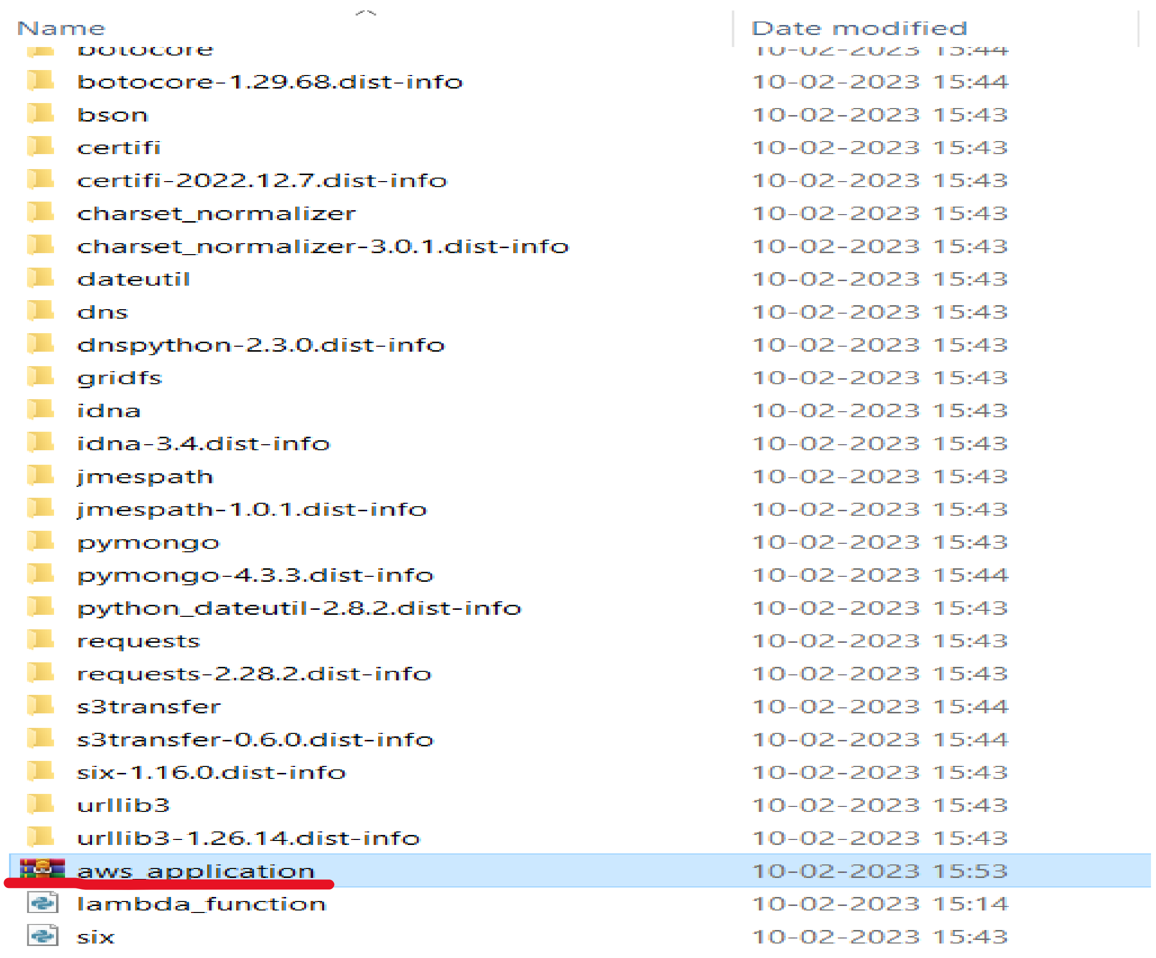
pip install --platform manylinux2014_x86_64 --target=lambda_function_code --implementation cp

--python==3.9 --only-binary=:all: --upgrade pymongo[srv] boto3 requests --> to install required packages for lambda function.

Select all and create a .zip file:


Big Data > AWS Project Version 2 > lambda_function_code

Name	Date modified	Type	Size
__pycache__	10-02-2023 15:43	File folder	
bin	10-02-2023 15:43	File folder	
boto3	10-02-2023 15:44	File folder	
boto3-1.26.68.dist-info	10-02-2023 15:44	File folder	
botocore	10-02-2023 15:44	File folder	
botocore-1.29.68.dist-info	10-02-2023 15:44	File folder	
bson	10-02-2023 15:43	File folder	
certifi	10-02-2023 15:43	File folder	
certifi-2022.12.7.dist-info	10-02-2023 15:43	File folder	
charset_normalizer	10-02-2023 15:43	File folder	
charset_normalizer-3.0.1.dist-info	10-02-2023 15:43	File folder	
dateutil	10-02-2023 15:43	File folder	
dns	10-02-2023 15:43	File folder	
dnspython-2.3.0.dist-info	10-02-2023 15:43	File folder	
gridfs	10-02-2023 15:43	File folder	
idna	10-02-2023 15:43	File folder	
idna-3.4.dist-info	10-02-2023 15:43	File folder	
jmespath	10-02-2023 15:43	File folder	
jmespath-1.0.1.dist-info	10-02-2023 15:43	File folder	
pymongo	10-02-2023 15:43	File folder	
pymongo-4.3.3.dist-info	10-02-2023 15:44	File folder	
python_dateutil-2.8.2.dist-info	10-02-2023 15:43	File folder	
requests	10-02-2023 15:43	File folder	
requests-2.28.2.dist-info	10-02-2023 15:43	File folder	
s3transfer	10-02-2023 15:44	File folder	
s3transfer-0.6.0.dist-info	10-02-2023 15:44	File folder	
six-1.16.0.dist-info	10-02-2023 15:43	File folder	

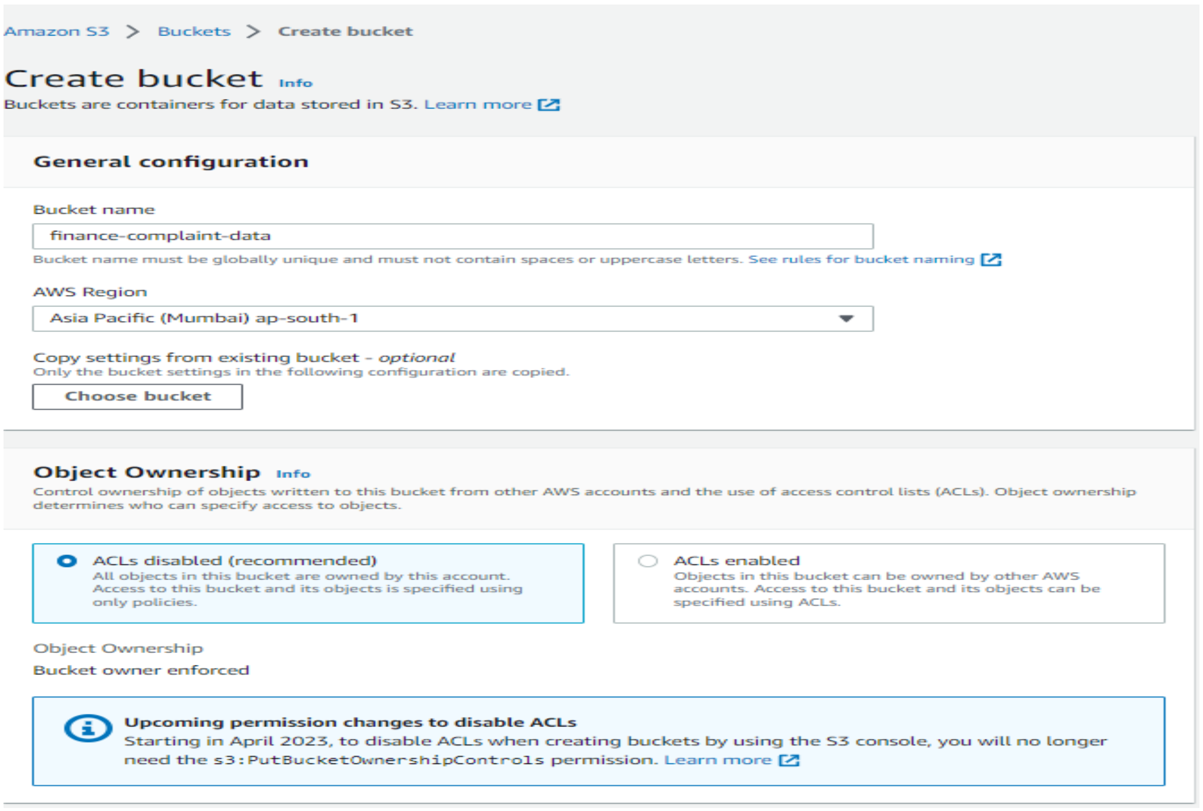


Name	Date modified
botocore	10-02-2023 15:44
botocore-1.29.68.dist-info	10-02-2023 15:44
bson	10-02-2023 15:43
certifi	10-02-2023 15:43
certifi-2022.12.7.dist-info	10-02-2023 15:43
charset_normalizer	10-02-2023 15:43
charset_normalizer-3.0.1.dist-info	10-02-2023 15:43
dateutil	10-02-2023 15:43
dns	10-02-2023 15:43
dnspython-2.3.0.dist-info	10-02-2023 15:43
gridfs	10-02-2023 15:43
idna	10-02-2023 15:43
idna-3.4.dist-info	10-02-2023 15:43
jmespath	10-02-2023 15:43
jmespath-1.0.1.dist-info	10-02-2023 15:43
pymongo	10-02-2023 15:43
pymongo-4.3.3.dist-info	10-02-2023 15:44
python_dateutil-2.8.2.dist-info	10-02-2023 15:43
requests	10-02-2023 15:43
requests-2.28.2.dist-info	10-02-2023 15:43
s3transfer	10-02-2023 15:44
s3transfer-0.6.0.dist-info	10-02-2023 15:44
six-1.16.0.dist-info	10-02-2023 15:43
urllib3	10-02-2023 15:43
urllib3-1.26.14.dist-info	10-02-2023 15:43
aws_application	10-02-2023 15:53
lambda_function	10-02-2023 15:14
six	10-02-2023 15:43

We will upload this .zip file to AWS Lambda Function later.

AWS S3:

Create S3 Bucket -> finance-complaint-data:



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name

finance-complaint-data

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region

Asia Pacific (Mumbai) ap-south-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

[✓ ACLs disabled (recommended)] All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

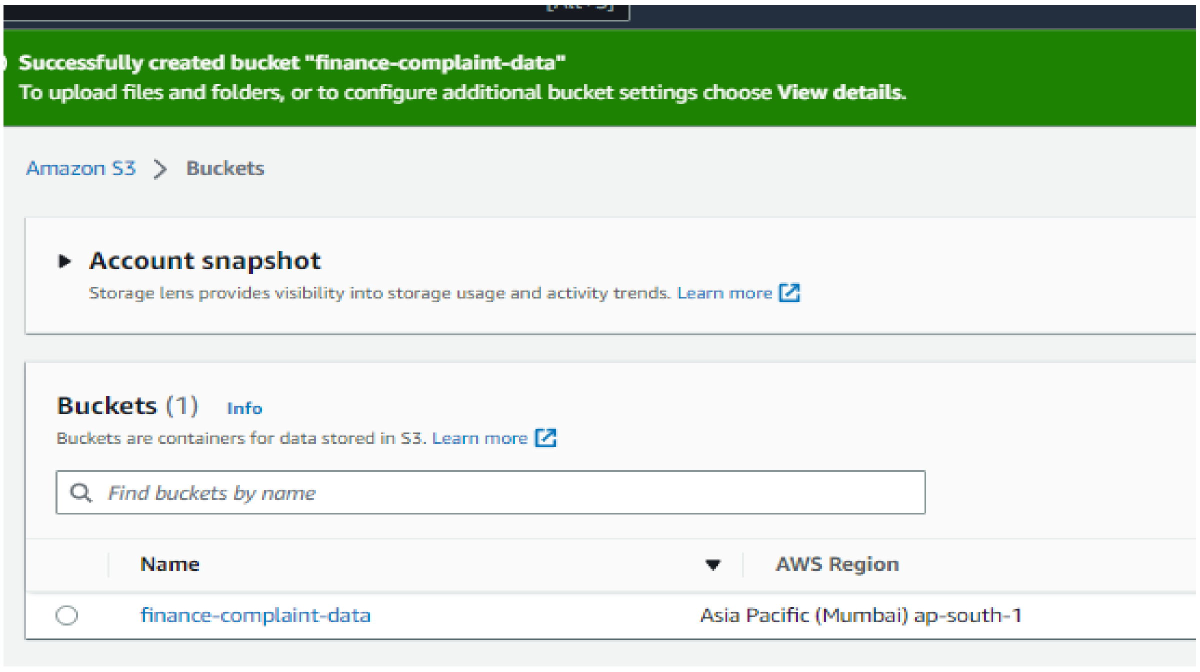
ACLs enabled Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

Upcoming permission changes to disable ACLs

Starting in April 2023, to disable ACLs when creating buckets by using the S3 console, you will no longer need the s3:PutBucketOwnershipControls permission. Learn more



Successfully created bucket "finance-complaint-data"

To upload files and folders, or to configure additional bucket settings choose **View details**.

Amazon S3 > Buckets

Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

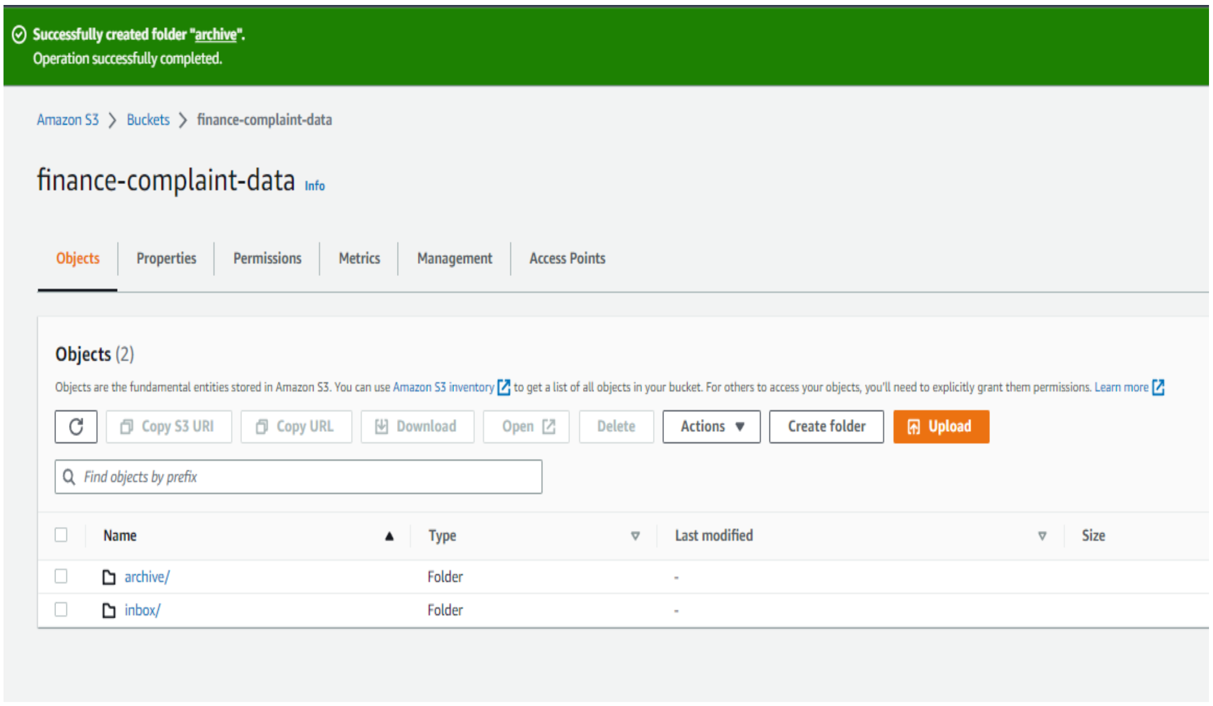
Find buckets by name

Name	AWS Region
finance-complaint-data	Asia Pacific (Mumbai) ap-south-1

Create two folder inbox and archive inside bucket finance-complaint-data:

Inbox -> to store the downloaded data.

Archive -> to store processed data.


Successfully created folder "archive".
Operation successfully completed.

Amazon S3 > Buckets > finance-complaint-data

finance-complaint-data Info

Objects | Properties | Permissions | Metrics | Management | Access Points

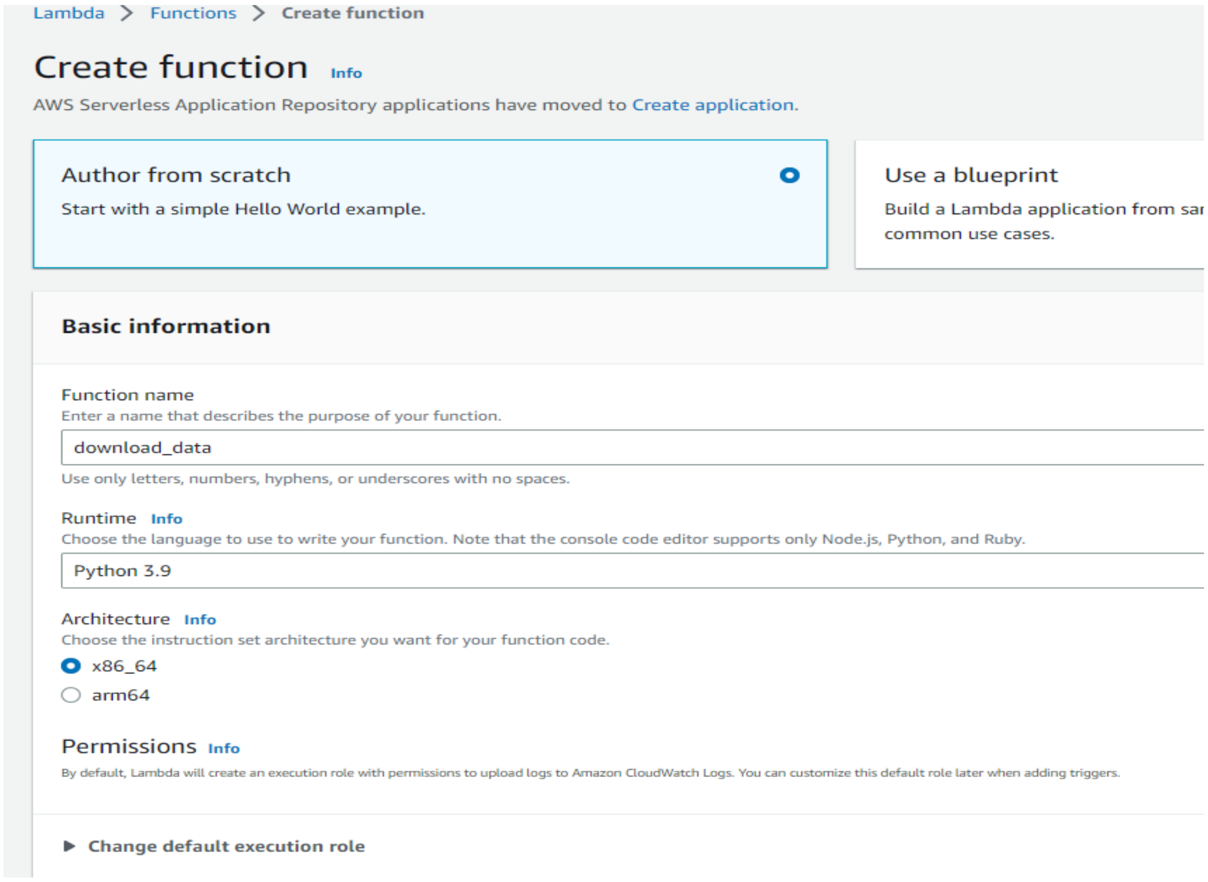
Objects (2)

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
archive/	Folder	-	
inbox/	Folder	-	

AWS Lambda:

Create lambda function download_data:



Lambda > Functions > Create function

Create function Info

AWS Serverless Application Repository applications have moved to Create application.

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

Use a blueprint — Build a Lambda application from sa
common use cases.

Basic information

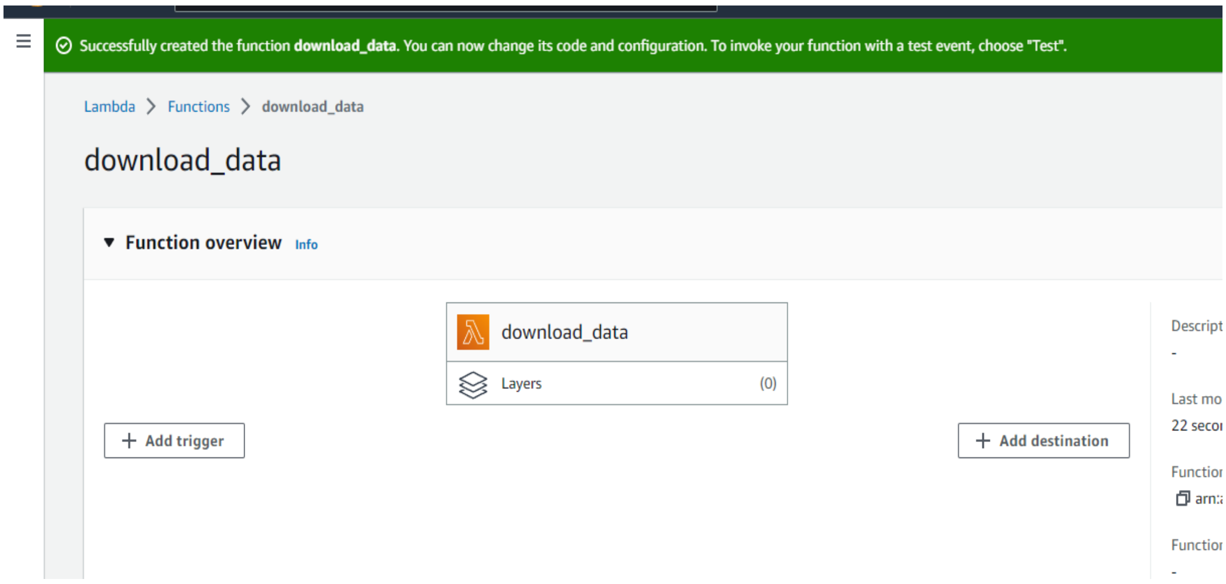
Function name
Enter a name that describes the purpose of your function.
download_data
Use only letters, numbers, hyphens, or underscores with no spaces.

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



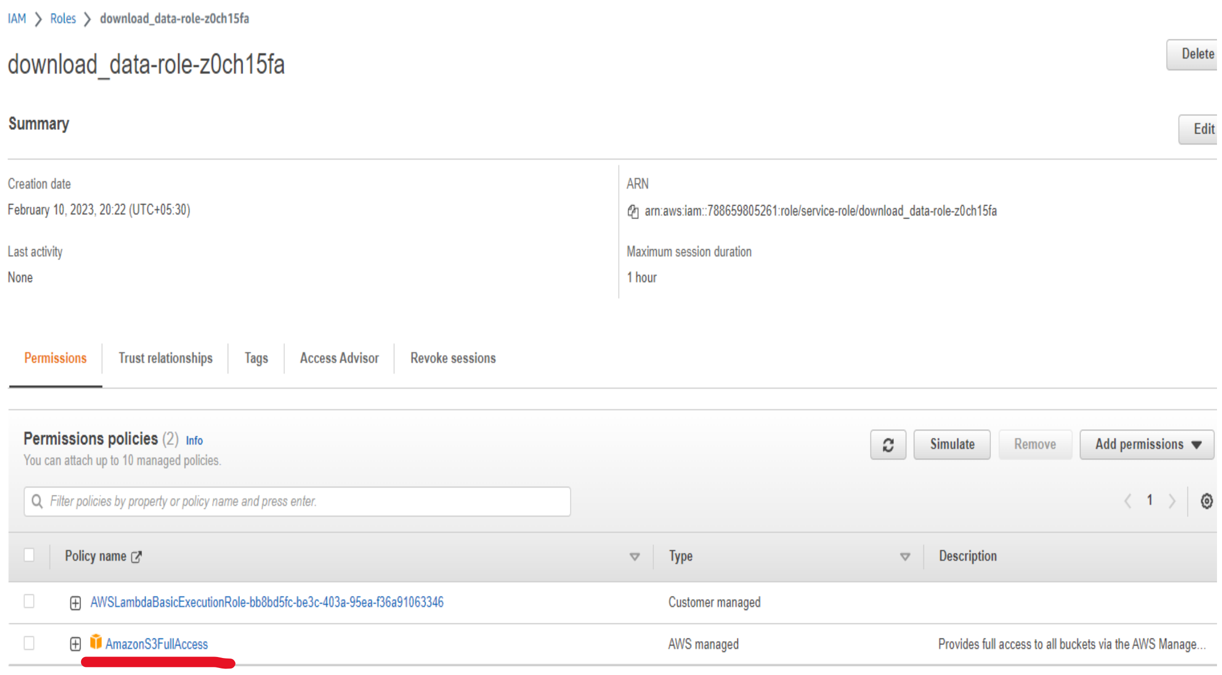
Successfully created the function **download_data**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > download_data

download_data

▼ Function overview Info

Update IAM Role: Go to configuration-> open role -> attach policy-> AWSS3FullAccess



IAM > Roles > download_data-role-z0ch15fa

download_data-role-z0ch15fa

Summary

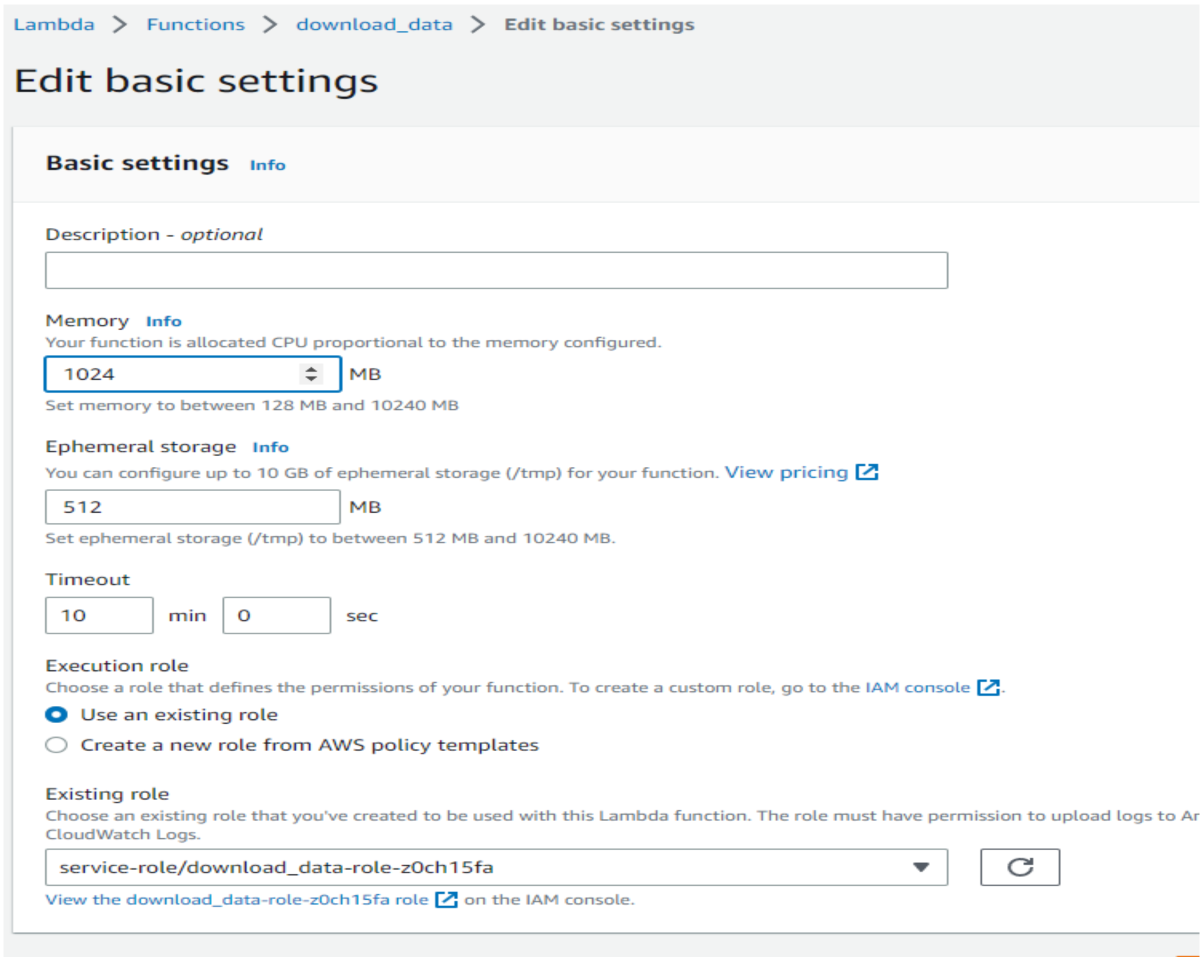
Creation date	ARN
February 10, 2023, 20:22 (UTC+05:30)	arn:aws:iam::788659805261:role/service-role/download_data-role-z0ch15fa
Last activity	Maximum session duration
None	1 hour

Permissions policies (2)

You can attach up to 10 managed policies.

Policy name	Type	Description
AWSLambdaBasicExecutionRole-bb8bd5fc-be3c-403a-95ea-f36a91063346	Customer managed	
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS Manage...

Update memory, timeout and other basic settings according to your requirement:



Lambda > Functions > download_data > Edit basic settings

Edit basic settings

Basic settings

Description - *optional*

Memory

Your function is allocated CPU proportional to the memory configured.

1024 MB

Set memory to between 128 MB and 10240 MB

Ephemeral storage

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing

512 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

Timeout

10 min 0 sec

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

● Use an existing role

○ Create a new role from AWS policy templates

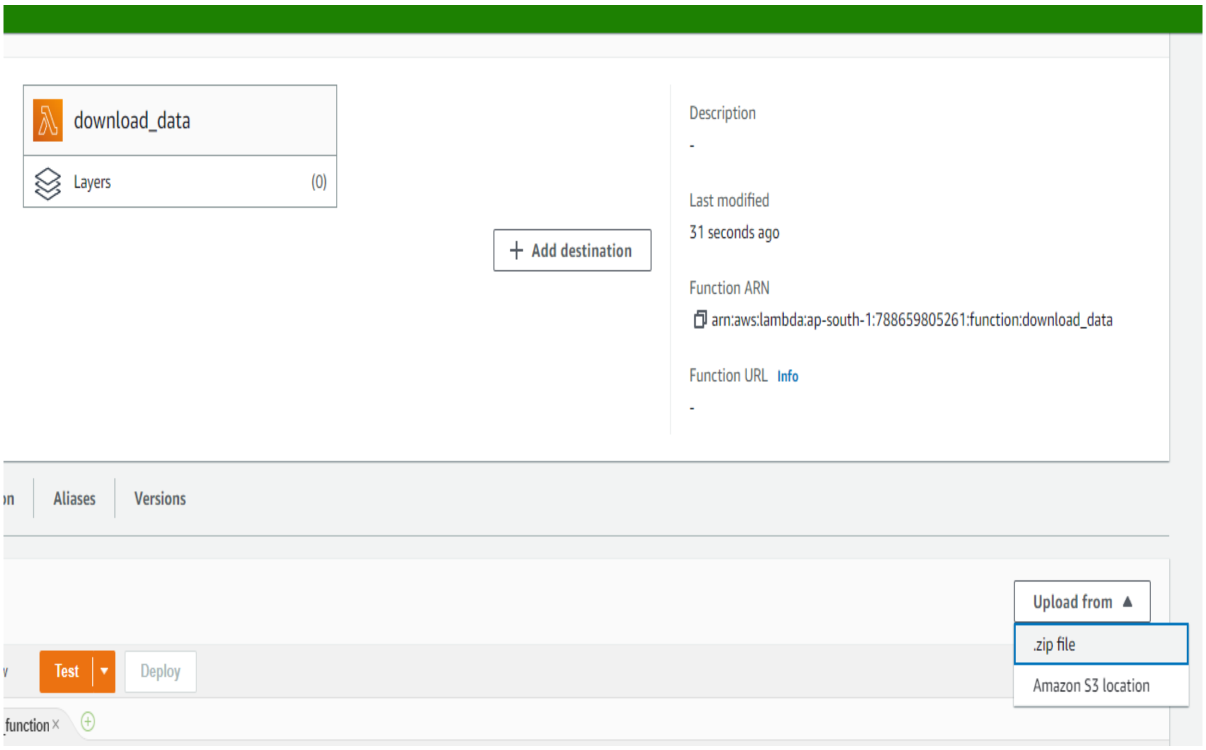
Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Ar
CloudWatch Logs.

service-role/download_data-role-z0ch15fa

View the download_data-role-z0ch15fa role on the IAM console.

Add code through .zip file:


download_data

Layers (0)

Add destination

Description

-

Last modified

31 seconds ago

Function ARN

arn:aws:lambda:ap-south-1:788659805261:function:download_data

Function URL Info

-

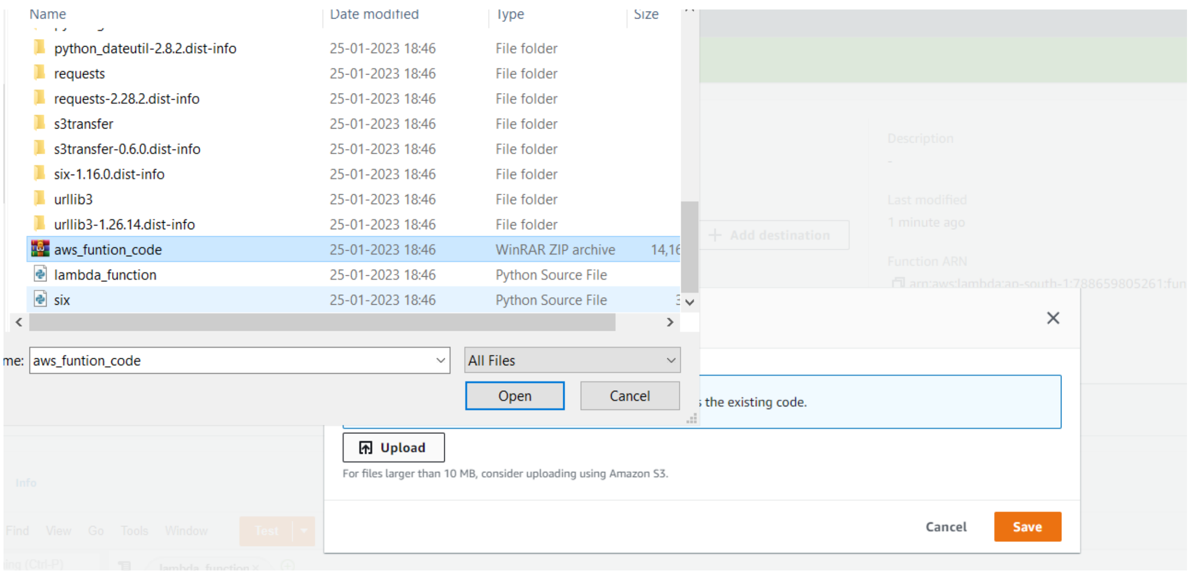
Aliases | Versions

Upload from ▲

.zip file

Amazon S3 location

Test | Deploy



Name	Date modified	Type	Size
python_dateutil-2.8.2.dist-info	25-01-2023 18:46	File folder	
requests	25-01-2023 18:46	File folder	
requests-2.28.2.dist-info	25-01-2023 18:46	File folder	
s3transfer	25-01-2023 18:46	File folder	
s3transfer-0.6.0.dist-info	25-01-2023 18:46	File folder	
six-1.16.0.dist-info	25-01-2023 18:46	File folder	
urllib3	25-01-2023 18:46	File folder	
urllib3-1.26.14.dist-info	25-01-2023 18:46	File folder	
aws_funtion_code	25-01-2023 18:46	WinRAR ZIP archive	14,16
lambda_function	25-01-2023 18:46	Python Source File	
six	25-01-2023 18:46	Python Source File	

me: aws_funtion_code | All Files

Open | Cancel

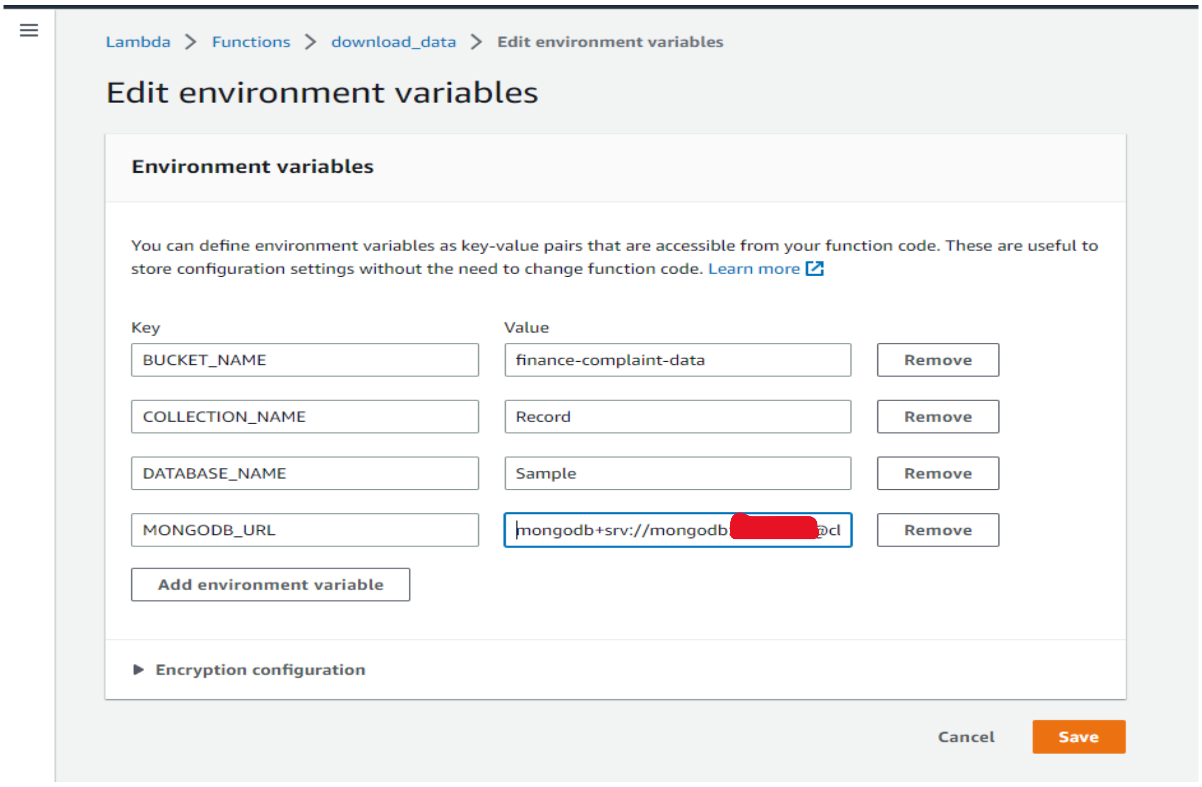
the existing code.

Upload

For files larger than 10 MB, consider uploading using Amazon S3.

Cancel | Save

Add environment variables:



Lambda > Functions > download_data > Edit environment variables

Edit environment variables

Environment variables

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

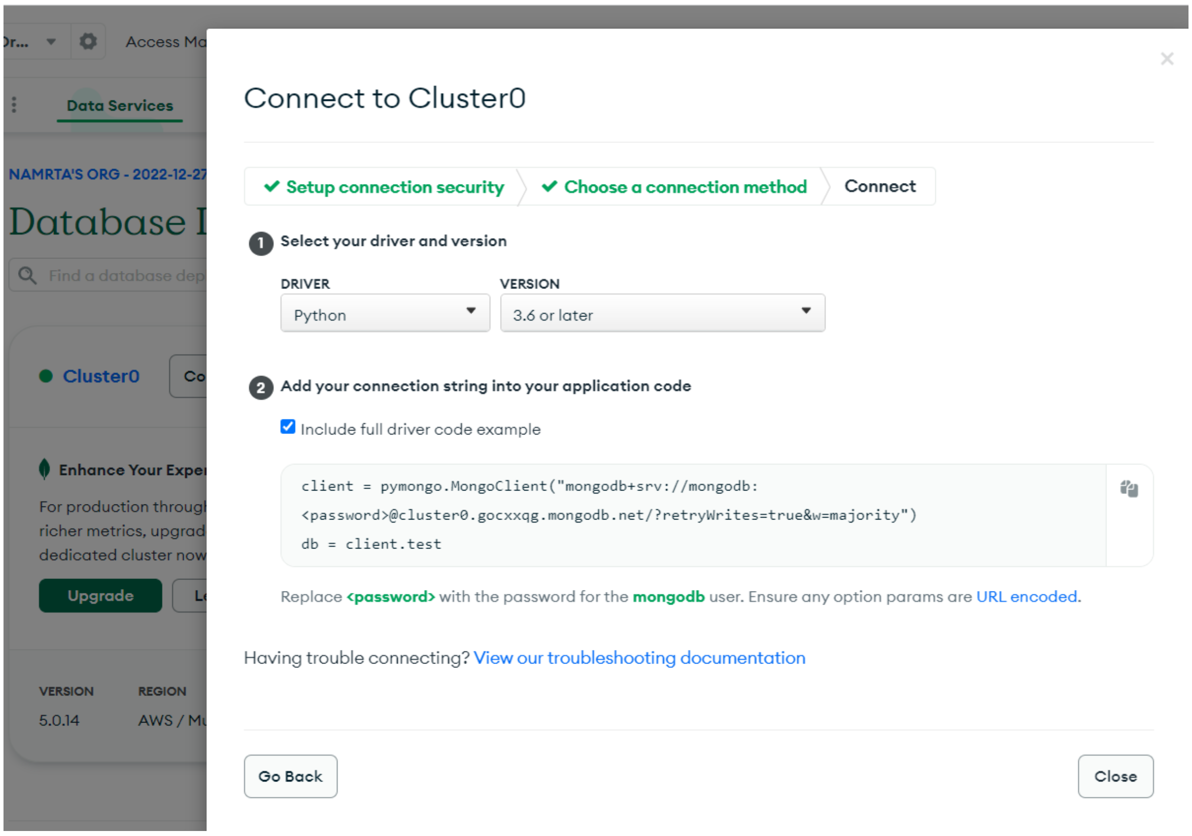
Key	Value	
BUCKET_NAME	finance-complaint-data	Remove
COLLECTION_NAME	Record	Remove
DATABASE_NAME	Sample	Remove
MONGODB_URL	mongodb+srv://mongodb[redacted]@cl	Remove

Add environment variable

▶ Encryption configuration

Cancel Save

MONGODB_URL to connect to the mongodb cluster:



Connect to Cluster0

✔ Setup connection security ✔ Choose a connection method Connect

1 Select your driver and version

DRIVER: Python

VERSION: 3.6 or later

2 Add your connection string into your application code

☑ Include full driver code example

```
client = pymongo.MongoClient("mongodb+srv://mongodb:
<password>@cluster0.gocxxqg.mongodb.net/?retryWrites=true&w=majority")
db = client.test
```

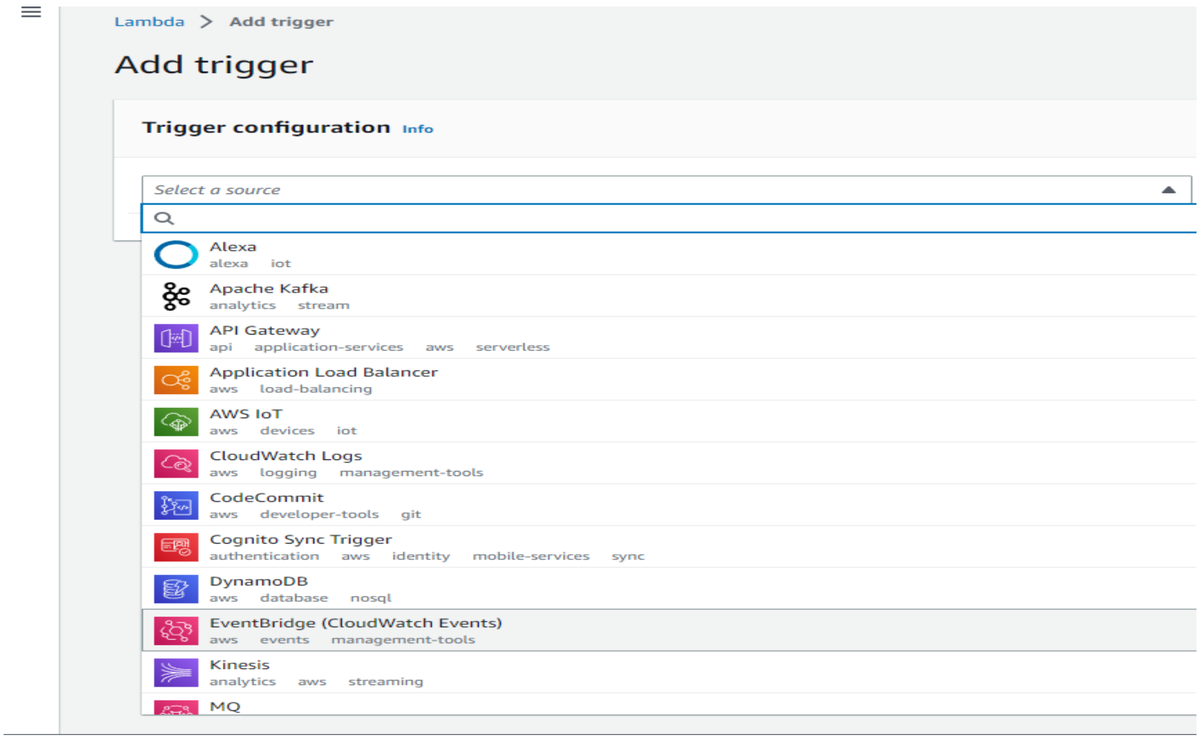
Replace <password> with the password for the mongodb user. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation

Go Back Close

AWS EventBridge:

Add trigger to schedule lambda function using EventBridge:



Lambda > Add trigger

Add trigger

Trigger configuration Info

Select a source

Alexa
alexa iot

Apache Kafka
analytics stream

API Gateway
api application-services aws serverless

Application Load Balancer
aws load-balancing

AWS IoT
aws devices iot

CloudWatch Logs
aws logging management-tools

CodeCommit
aws developer-tools git

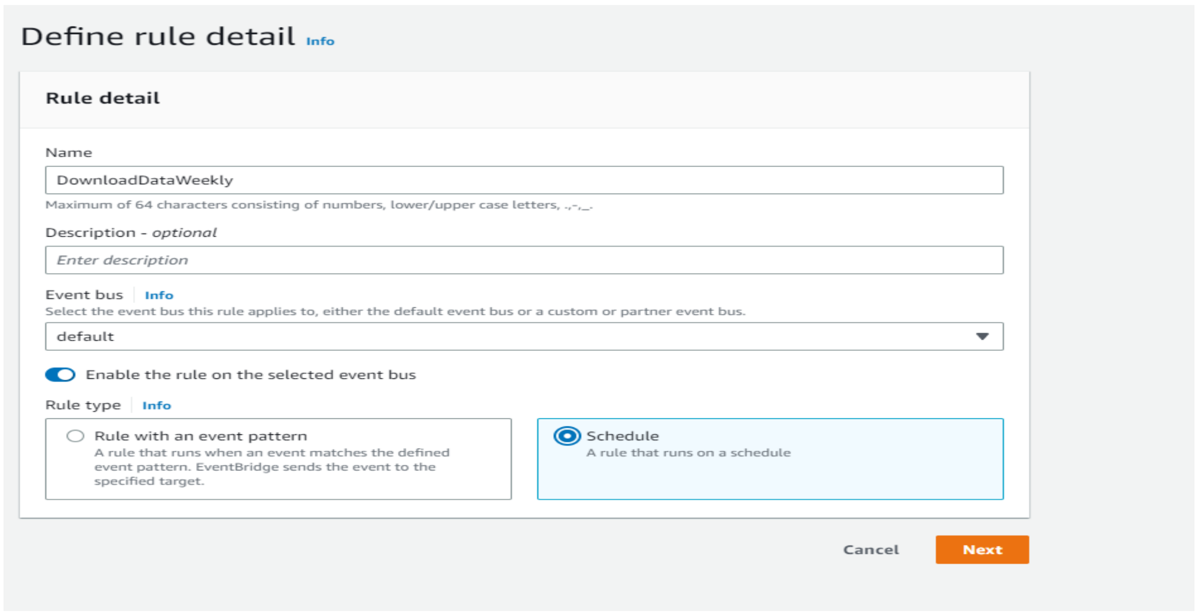
Cognito Sync Trigger
authentication aws identity mobile-services sync

DynamoDB
aws database nosql

EventBridge (CloudWatch Events)
aws events management-tools

Kinesis
analytics aws streaming

MQ



Define rule detail Info

Rule detail

Name
DownloadDataWeekly
Maximum of 64 characters consisting of numbers, lower/upper case letters, .,-,_.

Description - *optional*
Enter description

Event bus Info
Select the event bus this rule applies to, either the default event bus or a custom or partner event bus.
default

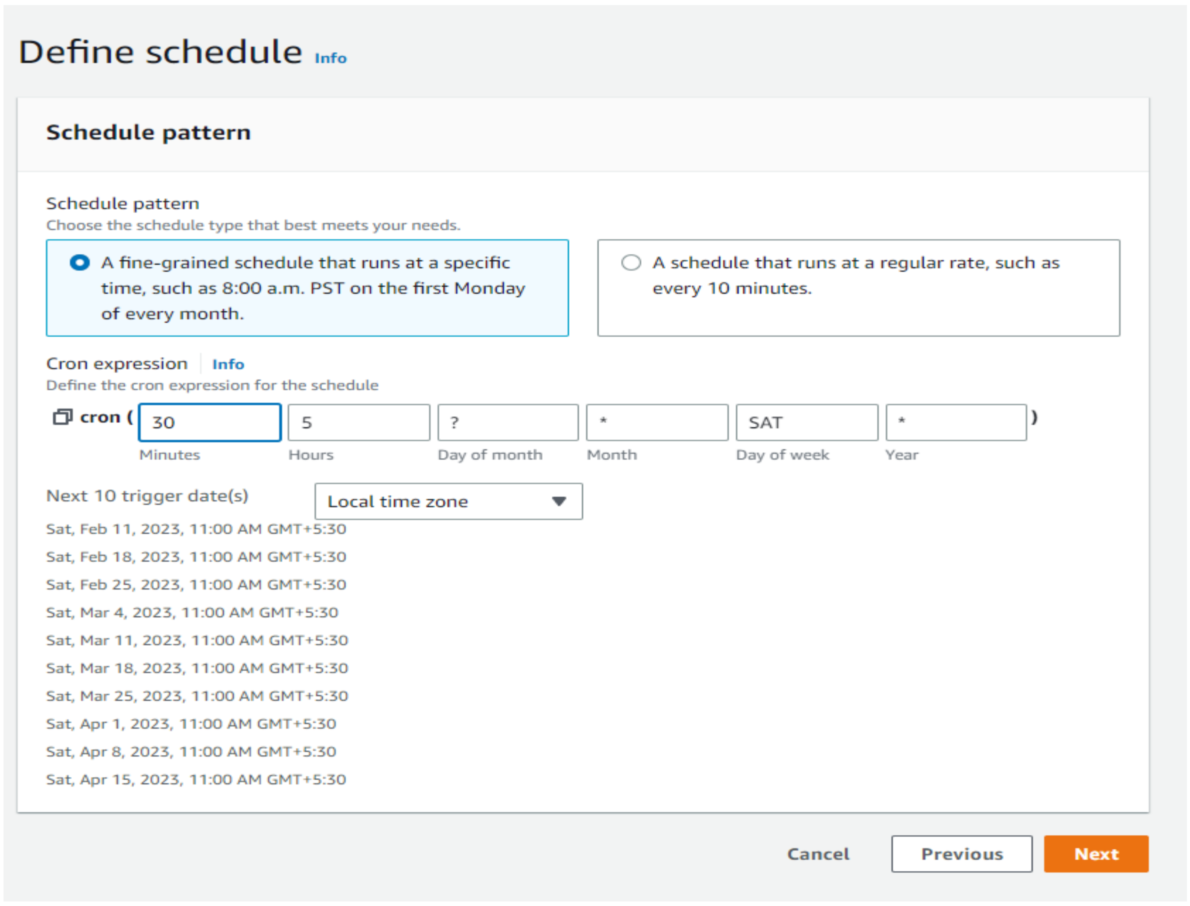
Enable the rule on the selected event bus

Rule type Info

Rule with an event pattern
A rule that runs when an event matches the defined event pattern. EventBridge sends the event to the specified target.

[✓ Schedule
A rule that runs on a schedule]

Cancel Next



Define schedule Info

Schedule pattern

Schedule pattern
Choose the schedule type that best meets your needs.

[✓ A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.] A schedule that runs at a regular rate, such as every 10 minutes.

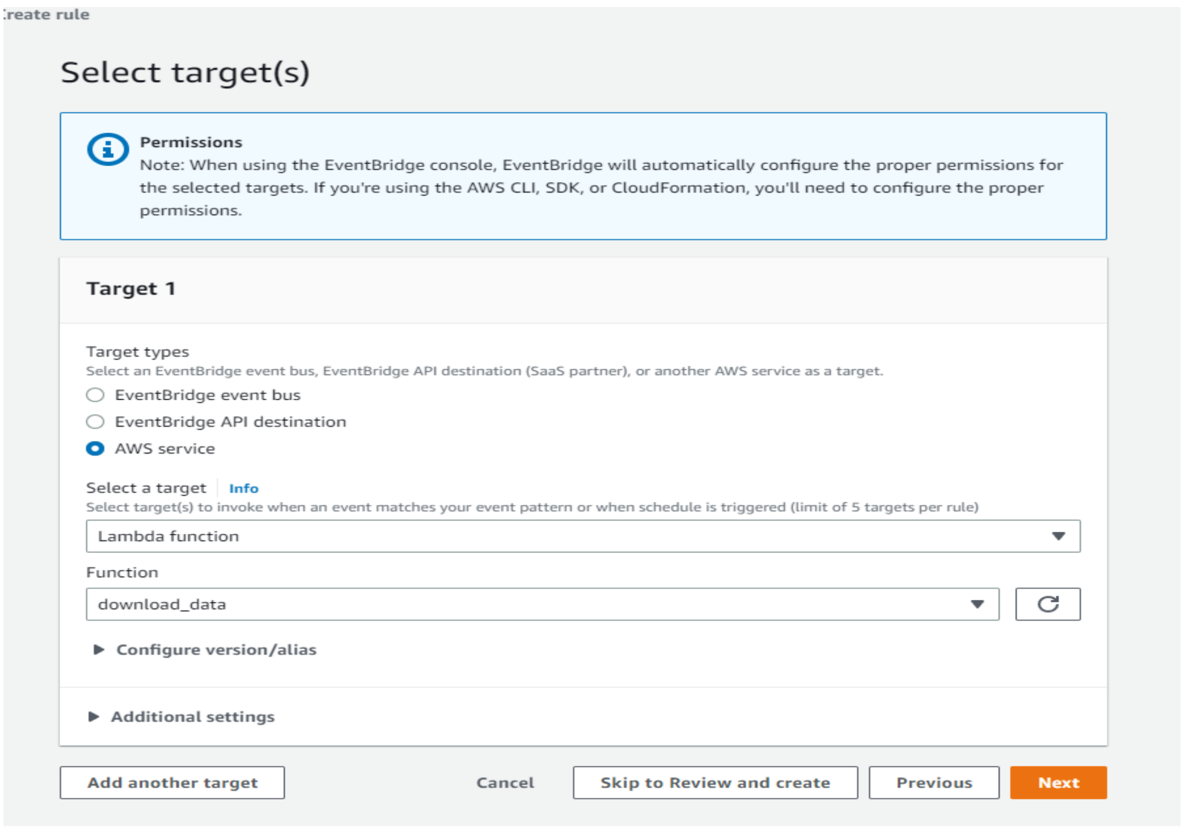
Cron expression Info
Define the cron expression for the schedule

Minutes	Hours	Day of month	Month	Day of week	Year
30	5	?	*	SAT	*

Next 10 trigger date(s) — Local time zone

```
Sat, Feb 11, 2023, 11:00 AM GMT+5:30
Sat, Feb 18, 2023, 11:00 AM GMT+5:30
Sat, Feb 25, 2023, 11:00 AM GMT+5:30
Sat, Mar 4, 2023, 11:00 AM GMT+5:30
Sat, Mar 11, 2023, 11:00 AM GMT+5:30
Sat, Mar 18, 2023, 11:00 AM GMT+5:30
Sat, Mar 25, 2023, 11:00 AM GMT+5:30
Sat, Apr 1, 2023, 11:00 AM GMT+5:30
Sat, Apr 8, 2023, 11:00 AM GMT+5:30
Sat, Apr 15, 2023, 11:00 AM GMT+5:30
```

Add target -> in this case lambda function download_data is the target.



:reate rule

Select target(s)

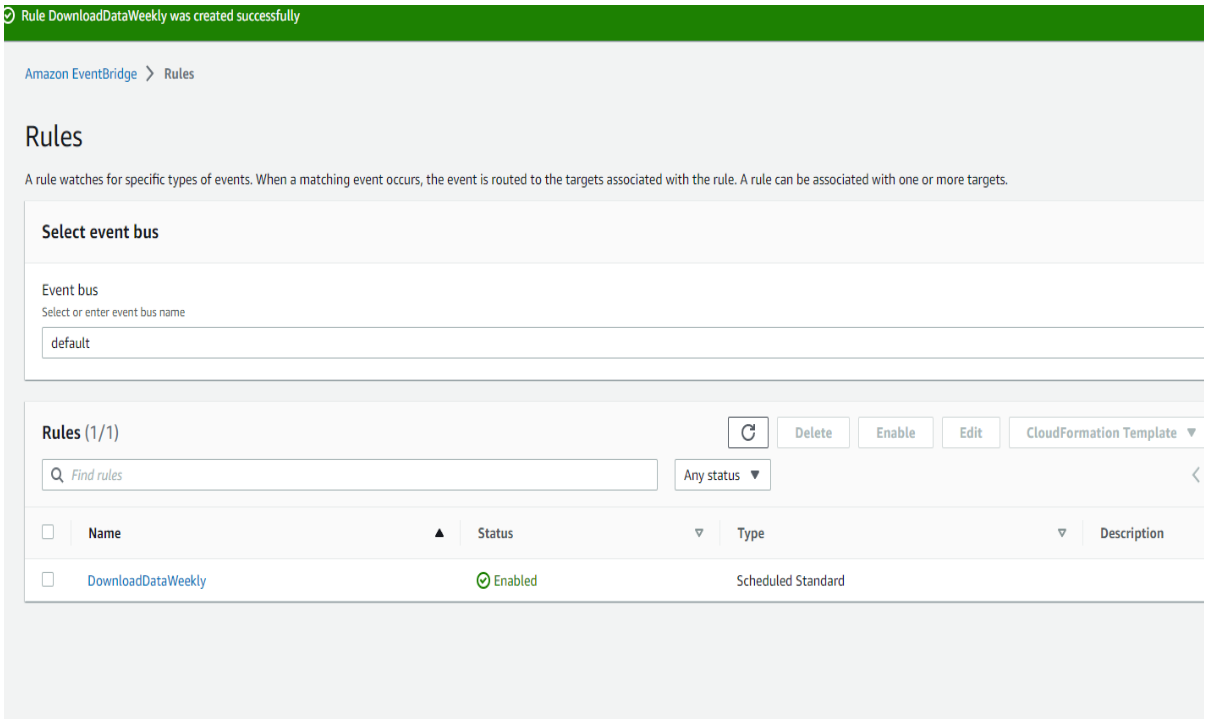
Permissions
Note: When using the EventBridge console, EventBridge will automatically configure the proper permissions for the selected targets. If you're using the AWS CLI, SDK, or CloudFormation, you'll need to configure the proper permissions.

Target 1

Target types
Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.
EventBridge event bus
EventBridge API destination
[✓ AWS service]

Select a target Info
Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)
Lambda function

Function
download_data


Rule DownloadDataWeekly was created successfully

Amazon EventBridge > Rules

Rules

A rule watches for specific types of events. When a matching event occurs, the event is routed to the targets associated with the rule. A rule can be associated with one or more targets.

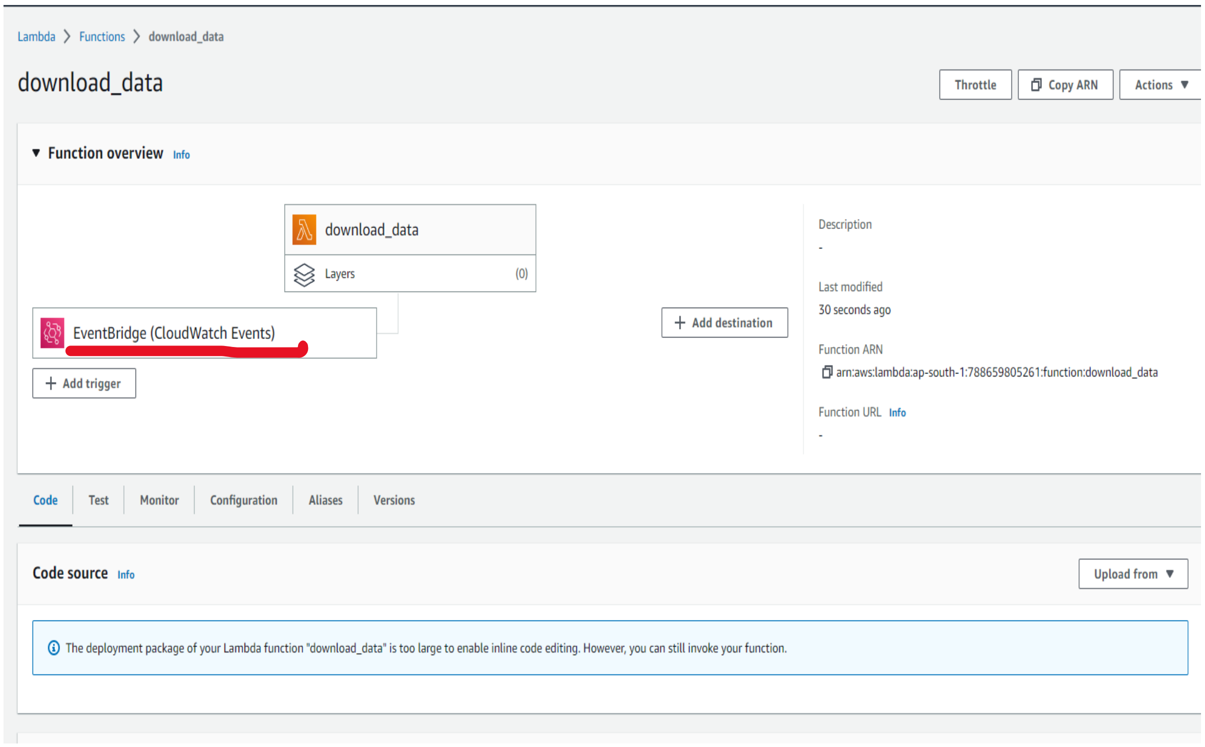
Select event bus

Event bus
Select or enter event bus name
default

Rules (1/1)

Name	Status	Type	Description
DownloadDataWeekly	Enabled	Scheduled Standard	

Trigger added.


Lambda > Functions > download_data

download_data

Function overview Info

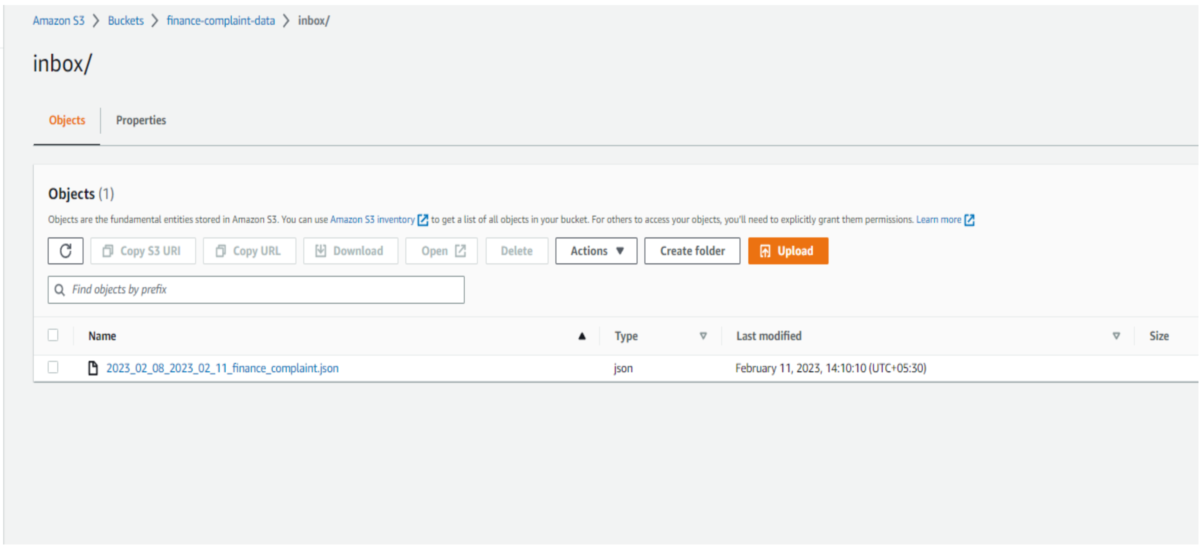
download_data
Layers (0)
EventBridge (CloudWatch Events)

Description
-
Last modified
30 seconds ago
Function ARN
arn:aws:lambda:ap-south-1:788659805261:function:download_data
Function URL Info
-

Code source Info

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

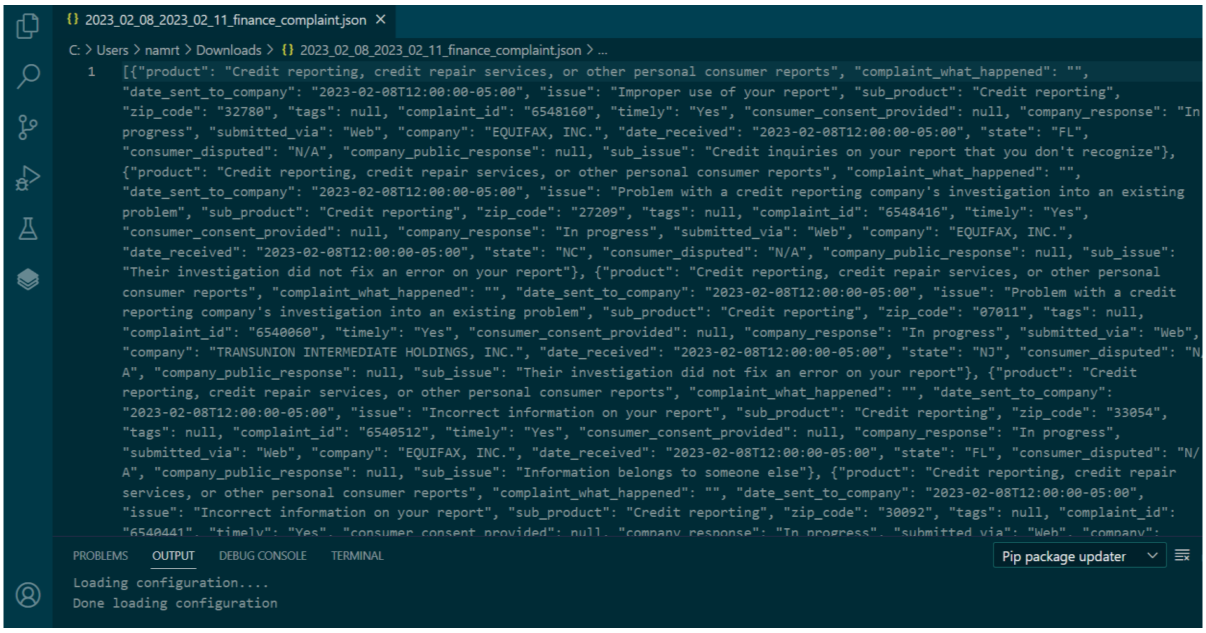
As soon as the trigger arrives, lambda function will download the data from API and store it in S3 bucket.



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
2023_02_08_2023_02_11_finance_complaint.json	json	February 11, 2023, 14:10:10 (UTC+05:30)	

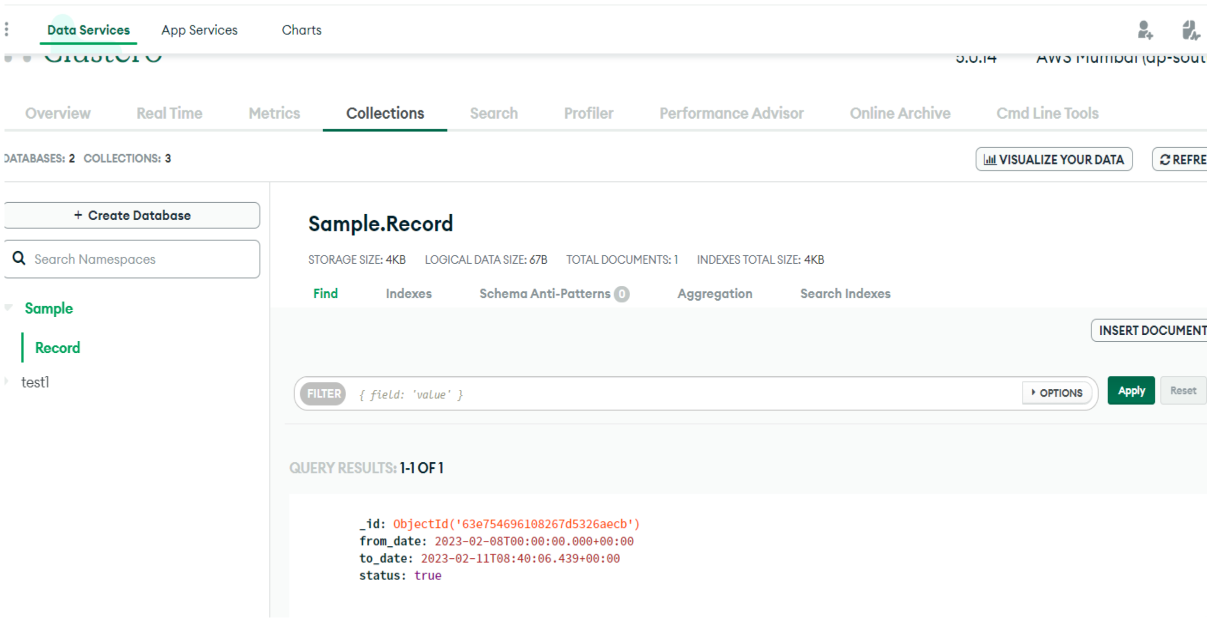
This is the data downloaded:


2023_02_08_2023_02_11_finance_complaint.json
C: > Users > namrt > Downloads > 2023_02_08_2023_02_11_finance_complaint.json > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Pip package updater

```
[{"product": "Credit reporting, credit repair services, or other personal consumer reports", "complaint_what_happened": "",
"date_sent_to_company": "2023-02-08T12:00:00-05:00", "issue": "Improper use of your report", "sub_product": "Credit reporting",
"zip_code": "32780", "tags": null, "complaint_id": "6548160", "timely": "Yes", "consumer_consent_provided": null, "company_response": "In
progress", "submitted_via": "Web", "company": "EQUIFAX, INC.", "date_received": "2023-02-08T12:00:00-05:00", "state": "FL",
"consumer_disputed": "N/A", "company_public_response": null, "sub_issue": "Credit inquiries on your report that you don't recognize"},
{"product": "Credit reporting, credit repair services, or other personal consumer reports", "complaint_what_happened": "",
"date_sent_to_company": "2023-02-08T12:00:00-05:00", "issue": "Problem with a credit reporting company's investigation into an existing
problem", "sub_product": "Credit reporting", "zip_code": "27209", "tags": null, "complaint_id": "6548416", "timely": "Yes",
"consumer_consent_provided": null, "company_response": "In progress", "submitted_via": "Web", "company": "EQUIFAX, INC.",
"date_received": "2023-02-08T12:00:00-05:00", "state": "NC", "consumer_disputed": "N/A", "company_public_response": null, "sub_issue":
"Their investigation did not fix an error on your report"}, {"product": "Credit reporting, credit repair services, or other personal
consumer reports", "complaint_what_happened": "", "date_sent_to_company": "2023-02-08T12:00:00-05:00", "issue": "Problem with a credit
reporting company's investigation into an existing problem", "sub_product": "Credit reporting", "zip_code": "07011", "tags": null,
"complaint_id": "6540060", "timely": "Yes", "consumer_consent_provided": null, "company_response": "In progress", "submitted_via": "Web",
"company": "TRANSUNION INTERMEDIATE HOLDINGS, INC.", "date_received": "2023-02-08T12:00:00-05:00", "state": "NJ", "consumer_disputed": "N/
A", "company_public_response": null, "sub_issue": "Their investigation did not fix an error on your report"}, {"product": "Credit
reporting, credit repair services, or other personal consumer reports", "complaint_what_happened": "", "date_sent_to_company":
"2023-02-08T12:00:00-05:00", "issue": "Incorrect information on your report", "sub_product": "Credit reporting", "zip_code": "33054",
"tags": null, "complaint_id": "6540512", "timely": "Yes", "consumer_consent_provided": null, "company_response": "In progress",
"submitted_via": "Web", "company": "EQUIFAX, INC.", "date_received": "2023-02-08T12:00:00-05:00", "state": "FL", "consumer_disputed": "N/
A", "company_public_response": null, "sub_issue": "Information belongs to someone else"}, {"product": "Credit reporting, credit repair
services, or other personal consumer reports", "complaint_what_happened": "", "date_sent_to_company": "2023-02-08T12:00:00-05:00",
"issue": "Incorrect information on your report", "sub_product": "Credit reporting", "zip_code": "30092", "tags": null, "complaint_id":
"6540441", "timely": "Yes", "consumer_consent_provided": null, "company_response": "In progress", "submitted_via": "Web", "company":
```

```
Loading configuration....
Done loading configuration
```

from_date and to_date are saved in the MongoDB database for future reference.



Sample.Record

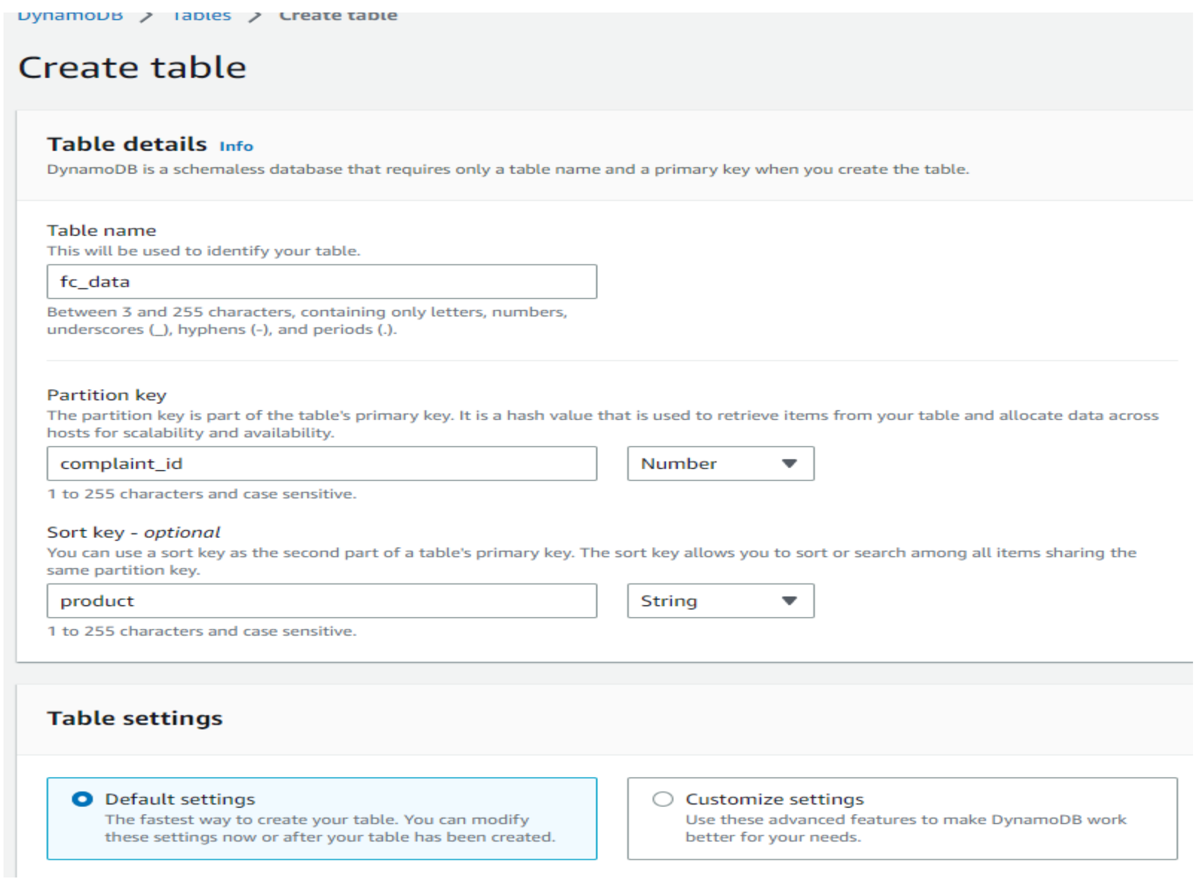
STORAGE SIZE: 4KB LOGICAL DATA SIZE: 67B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 4KB

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('63e754696108267d5326aecb')
from_date: 2023-02-08T00:00:00.000+00:00
to_date: 2023-02-11T08:40:06.439+00:00
status: true
```

DynamoDB:

Create DynamoDB table to store the final result.



DynamoDB > Tables > Create 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.

fc_data

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.

complaint_id — Number

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.

product — String

1 to 255 characters and case sensitive.

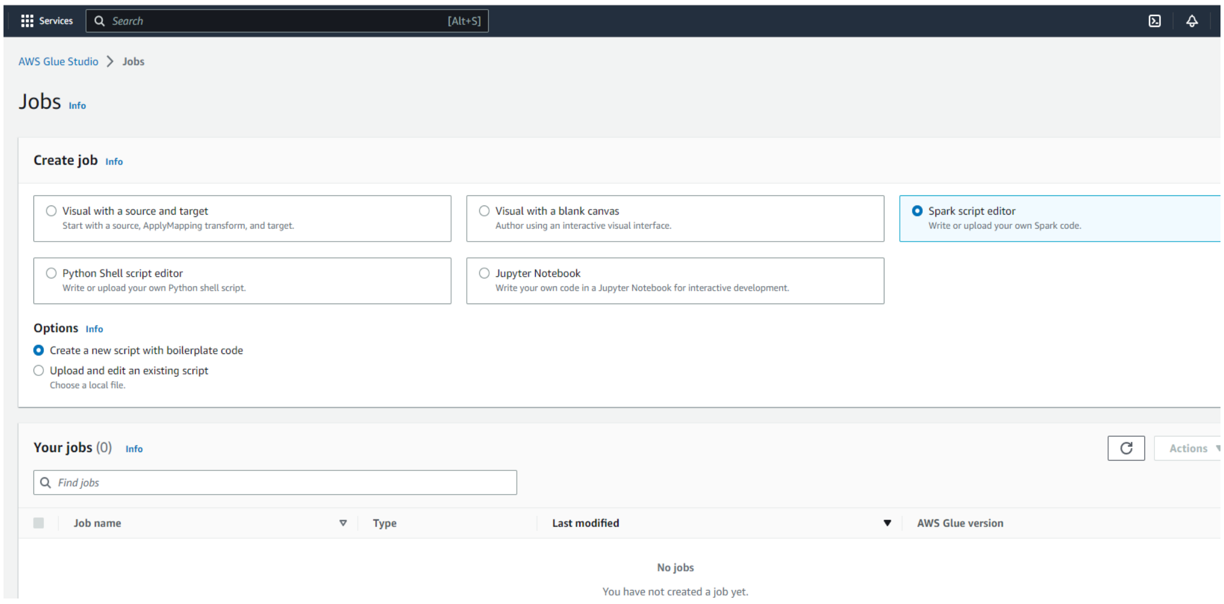
Table settings

[✓ 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.

AWS Glue:

Will use spark script to write our Glue job



Jobs Info

Create job Info

- ○ Visual with a source and target — Start with a source, ApplyMapping transform, and target.
- ○ Visual with a blank canvas — Author using an interactive visual interface.
- ● Spark script editor — Write or upload your own Spark code.
- ○ Python Shell script editor — Write or upload your own Python shell script.
- ○ Jupyter Notebook — Write your own code in a Jupyter Notebook for interactive development.

Options Info

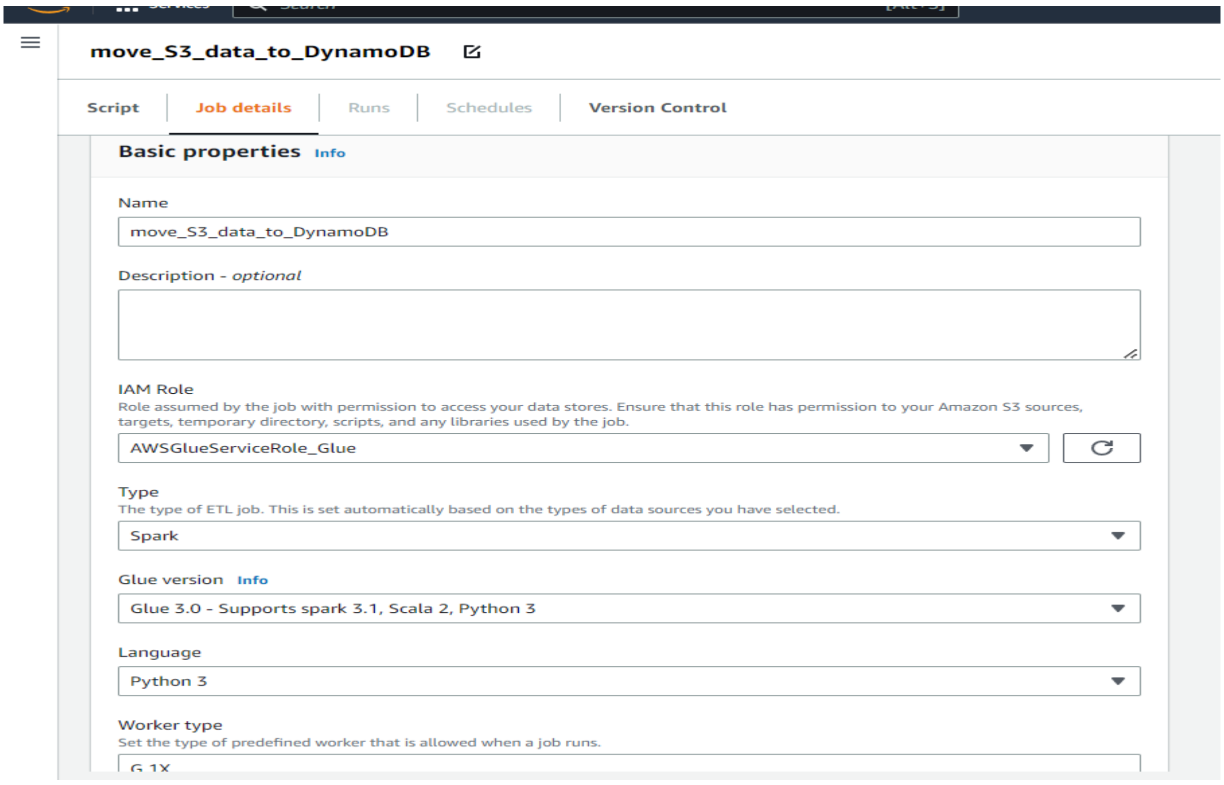
- ● Create a new script with boilerplate code
- ○ Upload and edit an existing script — Choose a local file.

Your jobs (0) Info

Job name	Type	Last modified	AWS Glue version

No jobs

You have not created a job yet.



move_S3_data_to_DynamoDB

Basic properties Info

Name: move_S3_data_to_DynamoDB

Description - *optional*

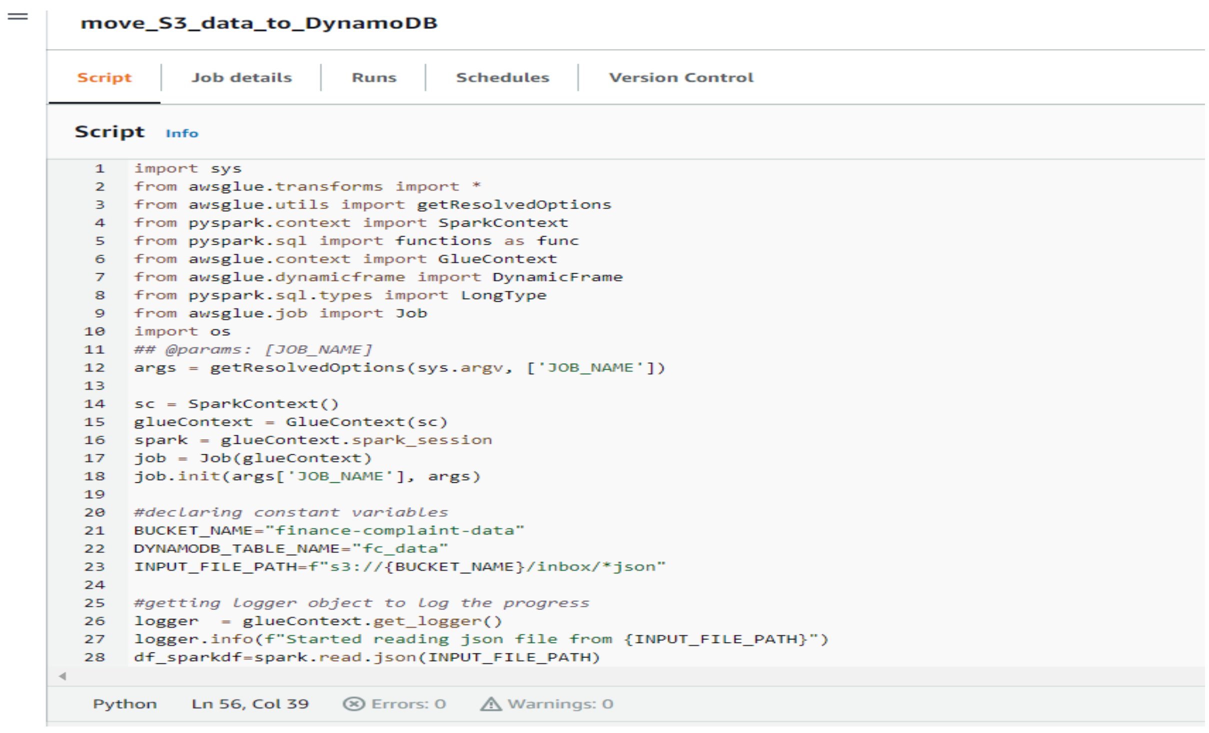
IAM Role
Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.
AWSGlueServiceRole_Glue

Type
The type of ETL job. This is set automatically based on the types of data sources you have selected.
Spark

Glue version Info
Glue 3.0 - Supports spark 3.1, Scala 2, Python 3

Language
Python 3

Worker type
Set the type of predefined worker that is allowed when a job runs.
G 1X

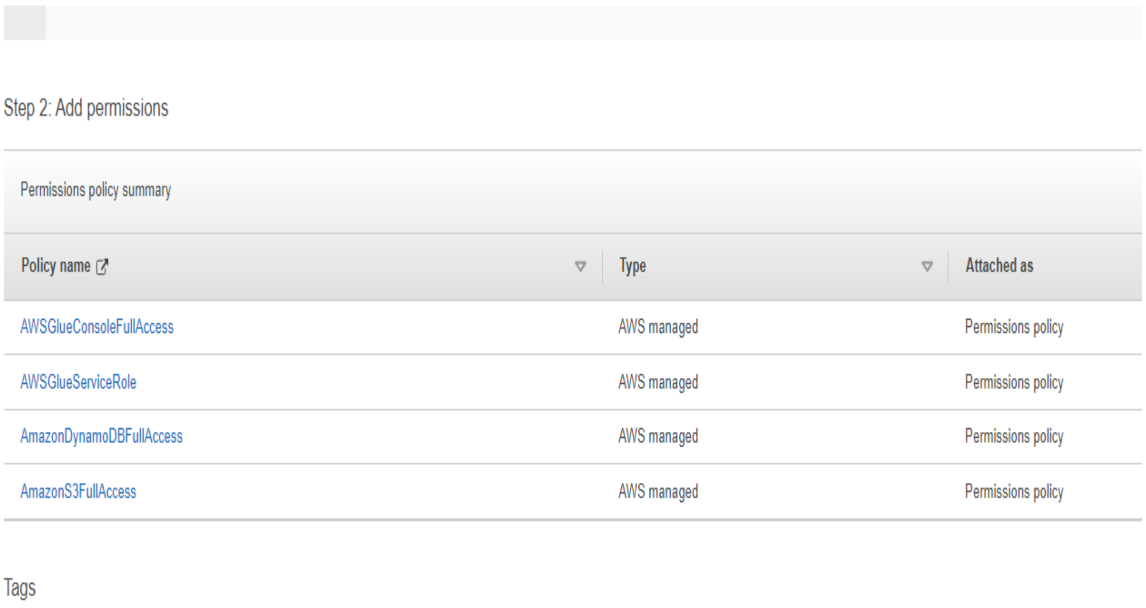


```python
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from pyspark.sql import functions as func
from awsglue.context import GlueContext
from awsglue.dynamicframe import DynamicFrame
from pyspark.sql.types import LongType
from awsglue.job import Job
import os
## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

#declaring constant variables
BUCKET_NAME="finance-complaint-data"
DYNAMODB_TABLE_NAME="fc_data"
INPUT_FILE_PATH=f"s3://{BUCKET_NAME}/inbox/*json"

#getting logger object to log the progress
logger  = glueContext.get_logger()
logger.info(f"Started reading json file from {INPUT_FILE_PATH}")
df_sparkdf=spark.read.json(INPUT_FILE_PATH)
```

Glue Job:

The script first reads the JSON file from the specified S3 location and converts it into a Spark DataFrame. Then it creates a DynamicFrame from the existing data in the DynamoDB table and converts it to spark dataframe. If the DynamoDB table has any data, the script performs a left join of the Spark DataFrame from S3 and the spark dataframe from DynamoDB on the "complaint_id" column, filters out the rows where the "existing_complaint_id" is null, and creates a new Spark DataFrame. If the DynamoDB table is empty, the Spark DataFrame from S3 is directly used as the new Spark DataFrame. The new Spark DataFrame is then converted back to a DynamicFrame and written to the DynamoDB table. The script also archives the original JSON file in S3 by copying it to another S3 location. The progress of the script is logged using a logger object.

Add all these policies to IAM Role so that AWS services can interact with one another:



Step 2: Add permissions

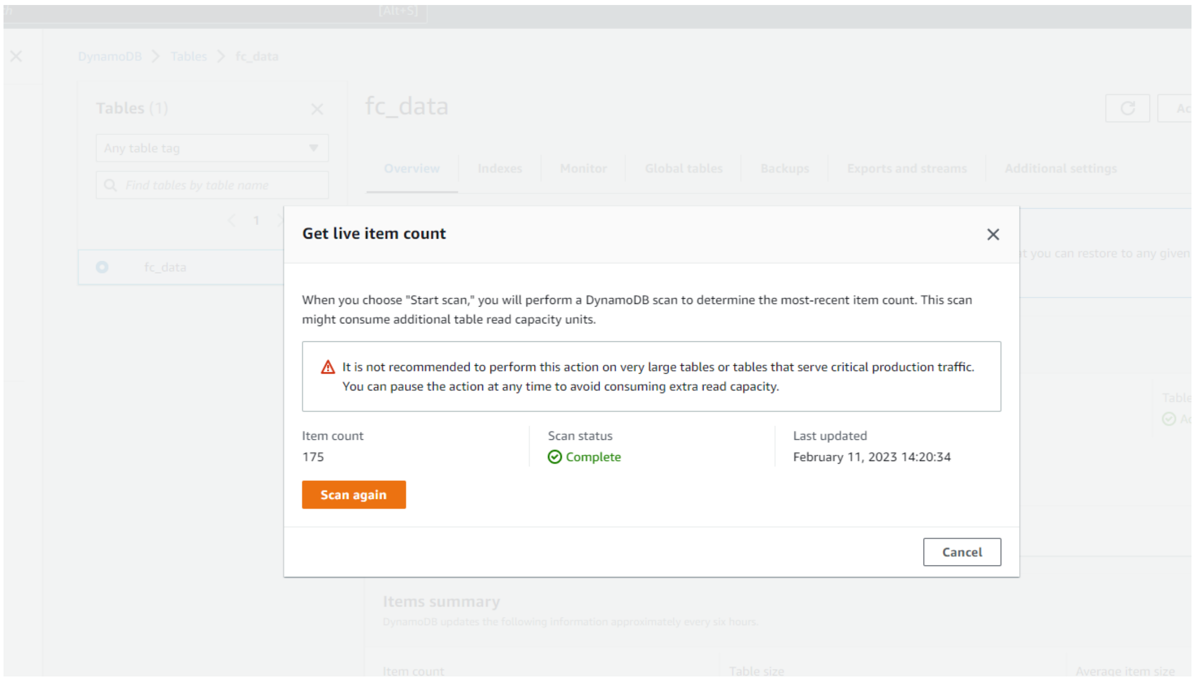
Permissions policy summary

Policy name	Type	Attached as
AWSGlueConsoleFullAccess	AWS managed	Permissions policy
AWSGlueServiceRole	AWS managed	Permissions policy
AmazonDynamoDBFullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

Tags

Run the Glue Job:

Can se the live item count in DynamoDB:



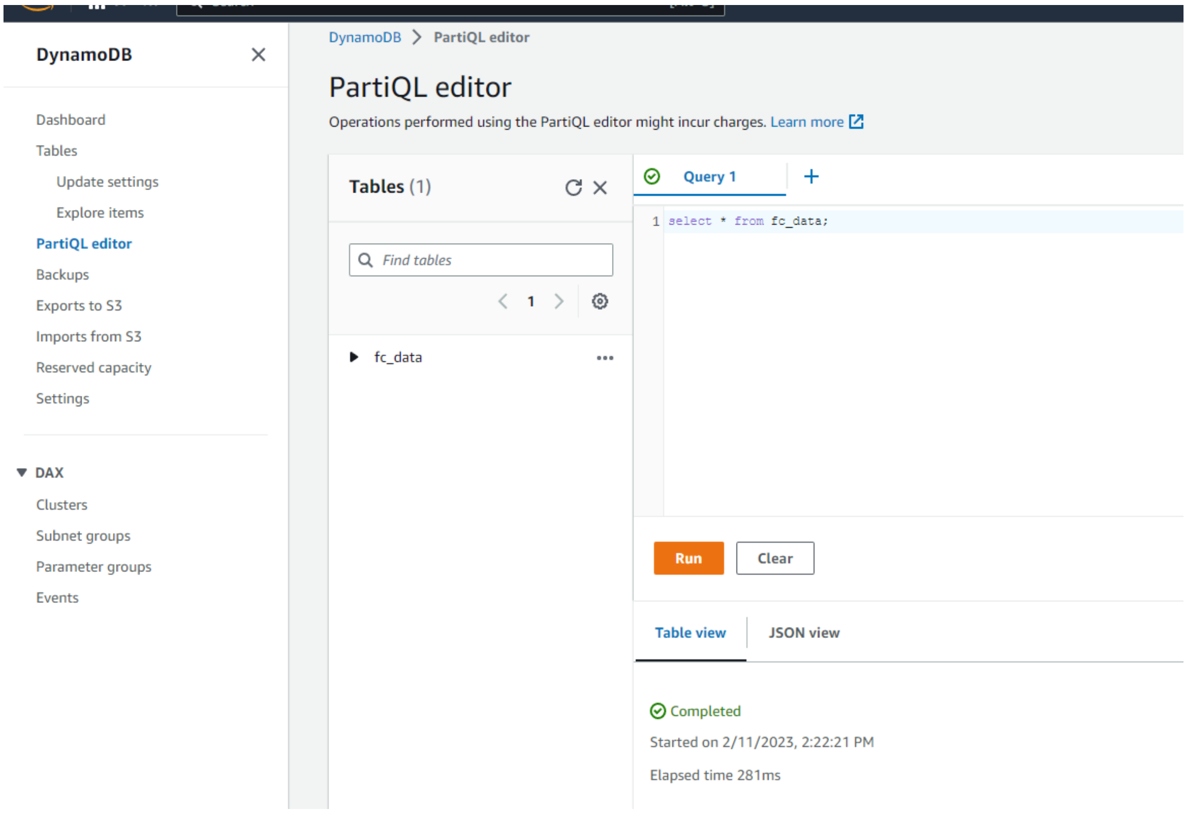
Get live item count

When you choose "Start scan," you will perform a DynamoDB scan to determine the most-recent item count. This scan might consume additional table read capacity units.

It is not recommended to perform this action on very large tables or tables that serve critical production traffic. You can pause the action at any time to avoid consuming extra read capacity.

Item count	Scan status	Last updated
175	Complete	February 11, 2023 14:20:34

Query the DynamoDB Table:



PartiQL editor

Operations performed using the PartiQL editor might incur charges. Learn more

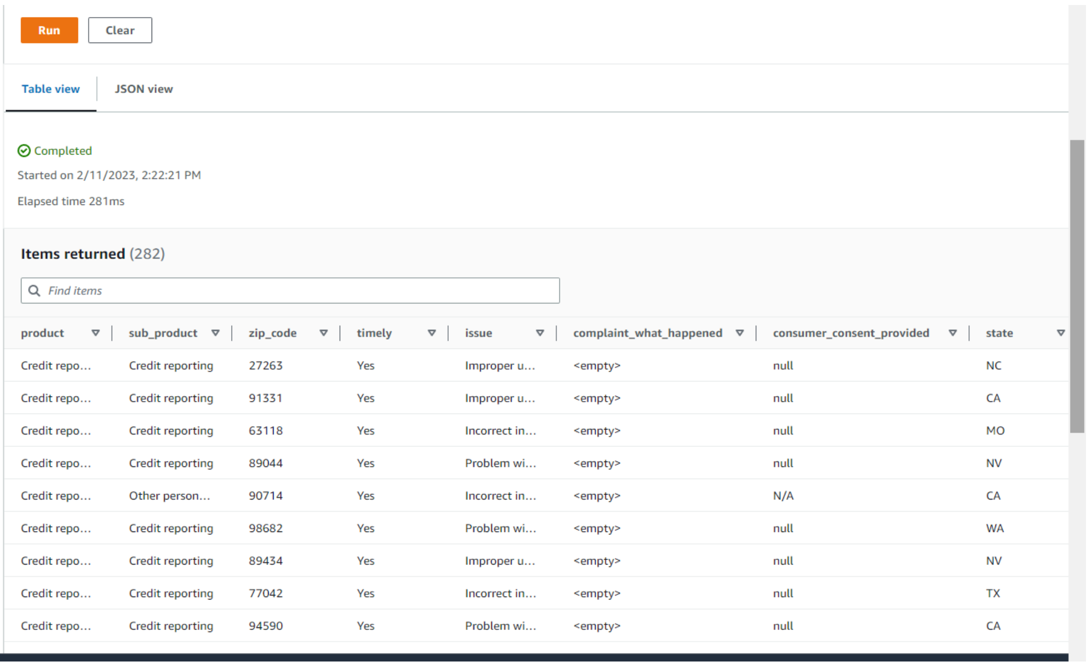
```
select * from fc_data;
```

Completed

Started on 2/11/2023, 2:22:21 PM

Elapsed time 281ms

Output:



Completed

Started on 2/11/2023, 2:22:21 PM

Elapsed time 281ms

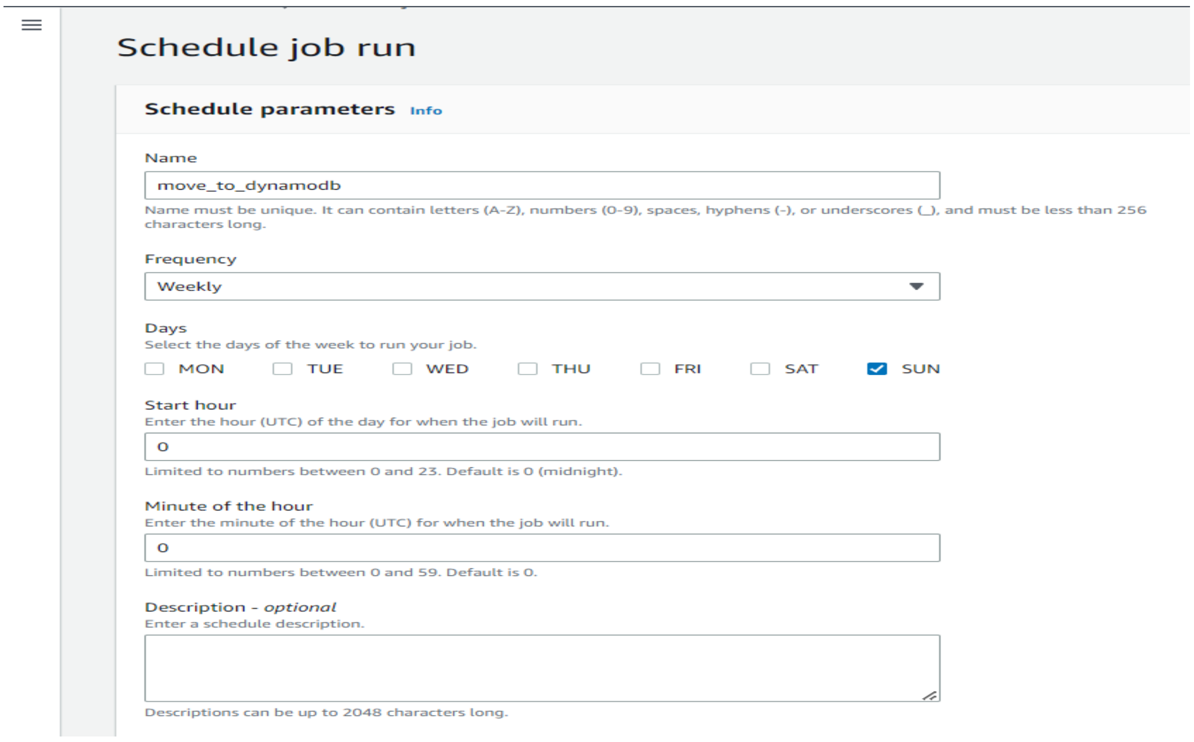
Items returned (282)

product	sub_product	zip_code	timely	issue	complaint_what_happened	consumer_consent_provided	state
Credit repo...	Credit reporting	27263	Yes	Improper u...	<empty>	null	NC
Credit repo...	Credit reporting	91331	Yes	Improper u...	<empty>	null	CA
Credit repo...	Credit reporting	63118	Yes	Incorrect in...	<empty>	null	MO
Credit repo...	Credit reporting	89044	Yes	Problem wi...	<empty>	null	NV
Credit repo...	Other person...	90714	Yes	Incorrect in...	<empty>	N/A	CA
Credit repo...	Credit reporting	98682	Yes	Problem wi...	<empty>	null	WA
Credit repo...	Credit reporting	89434	Yes	Improper u...	<empty>	null	NV
Credit repo...	Credit reporting	77042	Yes	Incorrect in...	<empty>	null	TX
Credit repo...	Credit reporting	94590	Yes	Problem wi...	<empty>	null	CA

We can also schedule our Glue job to run on specific frequency.

Here we are selecting weekly run.

As we are downloading our data on Saturday so I am scheduling my job to run on Sunday.



Schedule job run

Schedule parameters Info

Name

move_to_dynamodb

Name must be unique. It can contain letters (A-Z), numbers (0-9), spaces, hyphens (-), or underscores (_), and must be less than 256 characters long.

Frequency

Weekly

Days

Select the days of the week to run your job.

☐ MON ☐ TUE ☐ WED ☐ THU ☐ FRI ☐ SAT ✓ SUN

Start hour

Enter the hour (UTC) of the day for when the job will run.

0

Limited to numbers between 0 and 23. Default is 0 (midnight).

Minute of the hour

Enter the minute of the hour (UTC) for when the job will run.

0

Limited to numbers between 0 and 59. Default is 0.

Description - *optional*

Enter a schedule description.

Descriptions can be up to 2048 characters long.