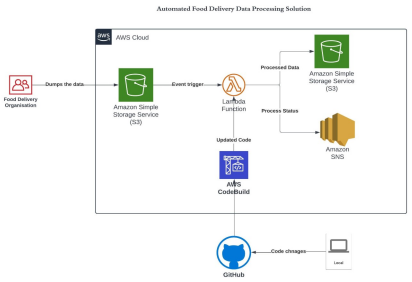
Streamlining Delivery Data Processing with AWS

Problem statement:

Create an automated AWS-based solution for processing daily delivery data from DoorDash. JSON files containing delivery records will be uploaded to an Amazon S3 bucket. An AWS Lambda function, triggered by the file upload, will filter the records based on delivery status and save the filtered data to another S3 bucket. Notifications regarding the processing outcome will be sent via Amazon SNS. Lambda should have CI/CD integration using CodeBuild so that any new requirements in processing can be quickly addressed.

Architecture diagram:



Step1:

Created JSON file, '2024-04-28-raw_data.json' with json data.


```
{"id": 1, "status": "delivered", "amount": 20.5, "date": "2024-03-09"}
{"id": 2, "status": "cancelled", "amount": 15.0, "date": "2024-03-09"}
{"id": 3, "status": "order placed", "amount": 22.5, "date": "2024-03-09"}
{"id": 4, "status": "delivered", "amount": 19.5, "date": "2024-03-09"}
{"id": 5, "status": "cancelled", "amount": 18.0, "date": "2024-03-09"}
{"id": 6, "status": "delivered", "amount": 23.5, "date": "2024-03-09"}
{"id": 7, "status": "order placed", "amount": 20.0, "date": "2024-03-09"}
{"id": 8, "status": "delivered", "amount": 25.5, "date": "2024-03-09"}
{"id": 9, "status": "delivered", "amount": 21.5, "date": "2024-03-09"}
{"id": 10, "status": "cancelled", "amount": 17.5, "date": "2024-03-09"}
```

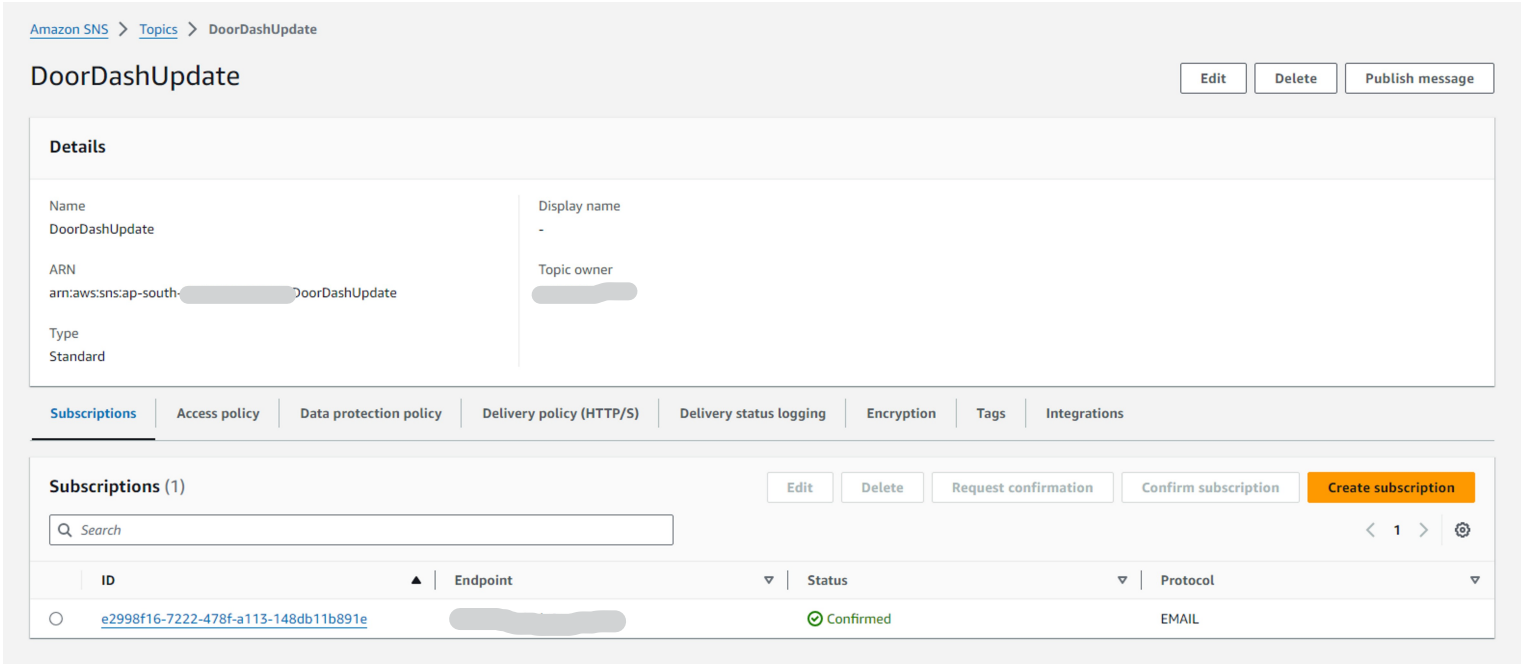

Step2:

Create two general-purpose Amazon S3 buckets. One designated for raw data landing and the other for processed data. Implement lifecycle rules to manage object storage, automatically transitioning objects between storage classes or deleting them after a specified number of days, aimed at cost optimization and audit trail maintenance.

Name	AWS Region	IAM Access Analyzer	Creation date
doordash-target-buckett	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 27, 2024, 15:33:20 (UTC+05:30)
doordash-landing-bucket	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 27, 2024, 15:32:20 (UTC+05:30)

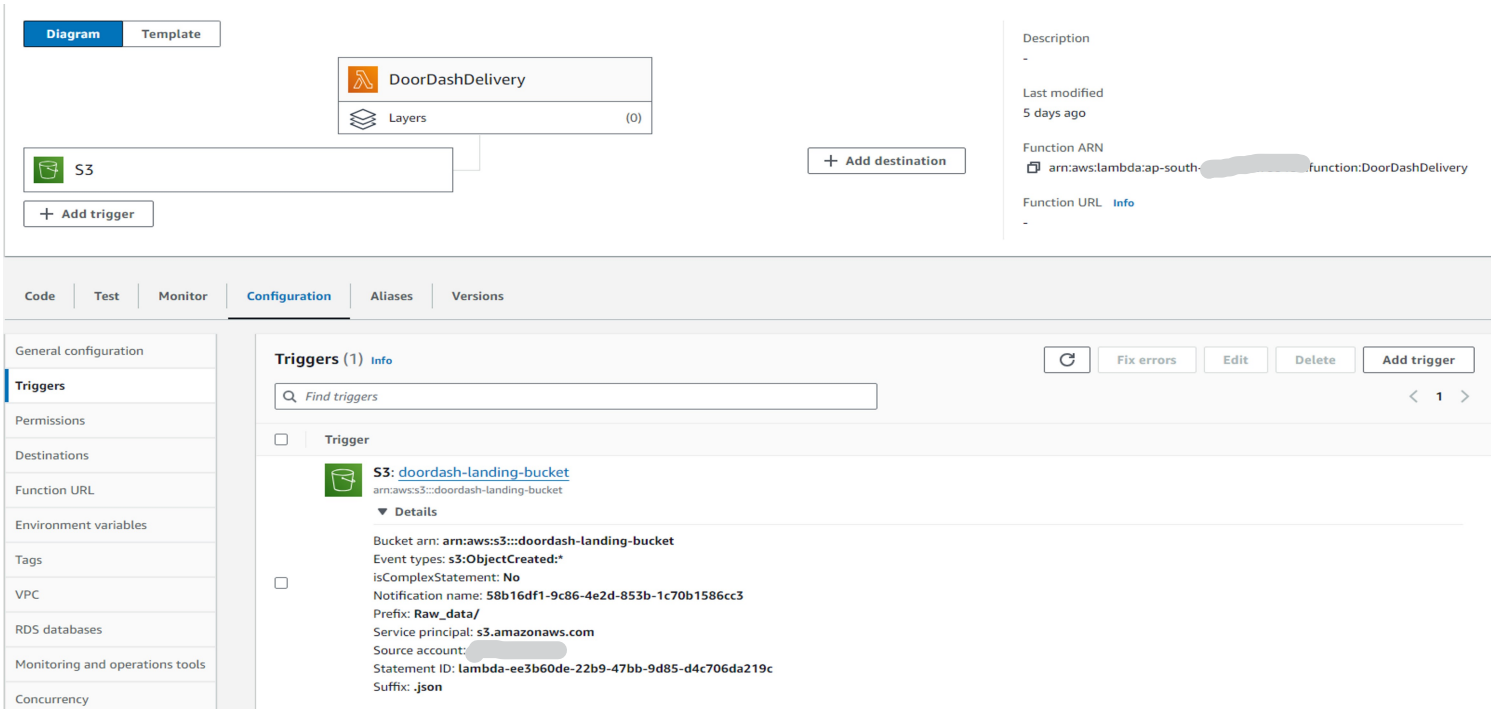
Step3:

Create an AWS Simple Notification Service (SNS) topic with a subscription using Email as the protocol for the endpoint.
Configure the Lambda function to send a notification to this SNS topic once the processing is completed, updating the status of the process to stakeholders via email.



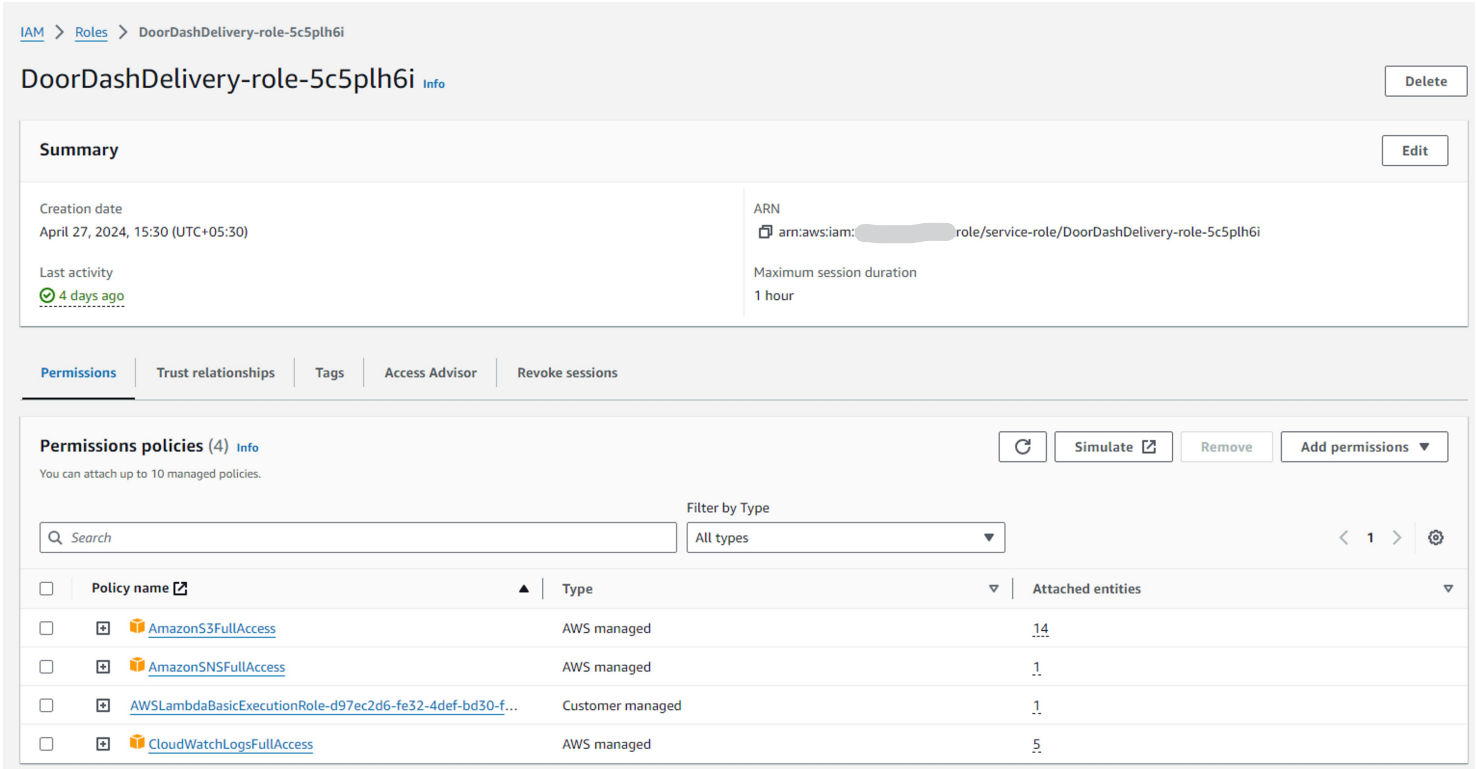
Step4:

Create a Lambda function and configure it with an S3 trigger to be invoked when an object with the suffix "Raw_Data" and prefix ".json" is uploaded to the dorrdash-landing S3 bucket.



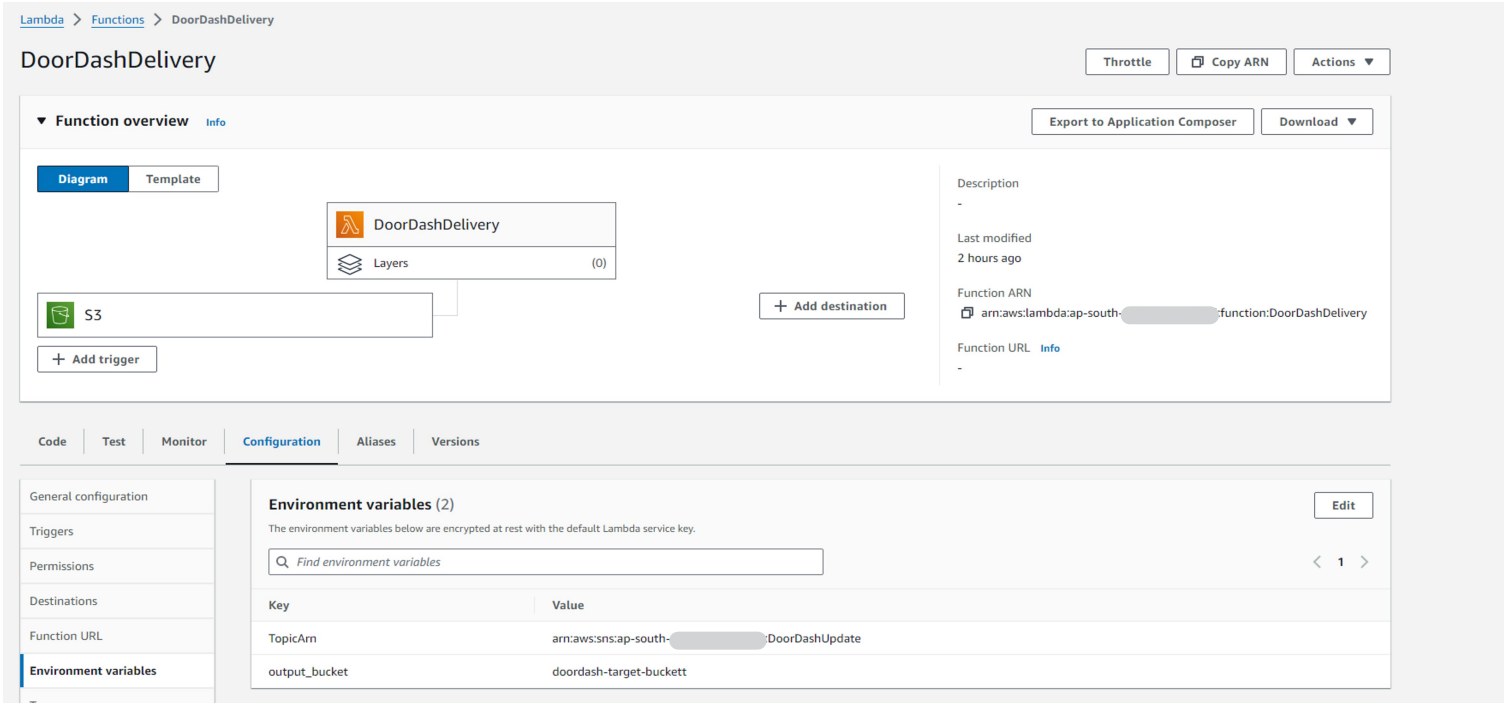
Step5:

Update the Lambda function's execution role permissions to include the necessary permissions to read data from the S3 bucket, publish messages through SNS, and write logs to CloudWatch.



Step6:

Store the final bucket name where the processed data has to be dumped and the SNS ARN in the Lambda's environmental variables to avoid hardcoding.



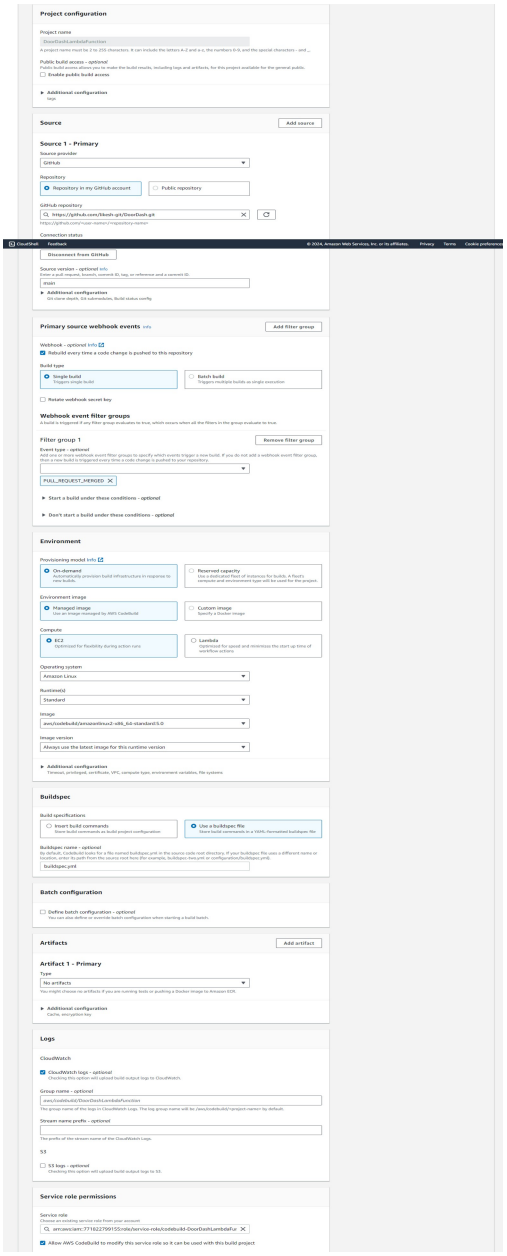
Step7:

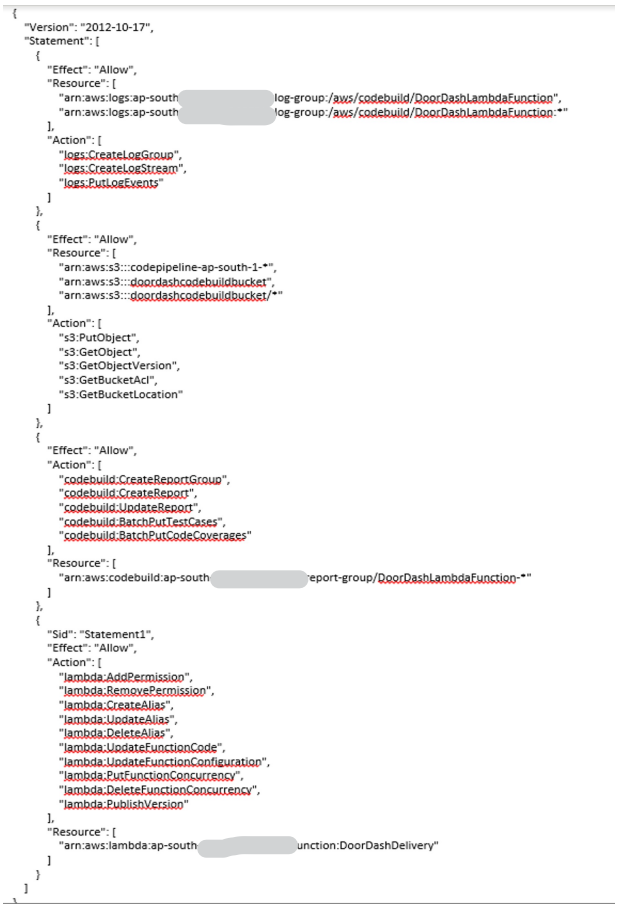
To implement continuous integration and continuous delivery (CI/CD) for the Lambda function:

- You would set up a pipeline that automatically builds, tests, and deploys changes to your Lambda function's code. This pipeline typically involves using AWS CodeBuild to compile and package your Lambda function code, along with any necessary dependencies. Once the code is built, it can be automatically deployed to your Lambda function. This CI/CD pipeline ensures that any updates or changes to your Lambda function can be quickly and reliably deployed to production, allowing you to iterate on your code efficiently and respond to new requirements or updates as they arise.

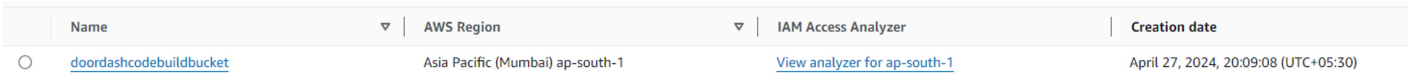
To achieve this:

a. Create a GitHub repository to host your Lambda function code and the build file required by CodeBuild.
b. Create a project on CodeBuild with the following parameters:
    a. Use the GitHub repository created above as the source.
    b. Configure CodeBuild to trigger builds automatically whenever a pull request is merged into the main branch of the repository.
    c. Ensure that the environment parameters in CodeBuild match those used for the Lambda function to prevent build failures.
    d. Select the option to use a buildspecfile.yml containing sequential steps for the entire build process, defining how CodeBuild should build, test, and deploy the Lambda function code.
c. Create another general-purpose Amazon S3 bucket to store the code after it is built by CodeBuild. This bucket will serve as the source for updating the Lambda function.
d. Update the CodeBuild's execution role by adding the necessary permissions, including the ARNs of the services.





```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": [
        "arn:aws:logs:ap-south[redacted]log-group:/aws/codebuild/DoorDashLambdaFunction",
        "arn:aws:logs:ap-south[redacted]log-group:/aws/codebuild/DoorDashLambdaFunction:*"
      ],
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ]
    },
    {
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::codepipeline-ap-south-1-*",
        "arn:aws:s3:::doordashcodebuildbucket",
        "arn:aws:s3:::doordashcodebuildbucket/*"
      ],
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketAcl",
        "s3:GetBucketLocation"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "codebuild:CreateReportGroup",
        "codebuild:CreateReport",
        "codebuild:UpdateReport",
        "codebuild:BatchPutTestCases",
        "codebuild:BatchPutCodeCoverages"
      ],
      "Resource": [
        "arn:aws:codebuild:ap-south[redacted]eport-group/DoorDashLambdaFunction-*"
      ]
    },
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "lambda:AddPermission",
        "lambda:RemovePermission",
        "lambda:CreateAlias",
        "lambda:UpdateAlias",
        "lambda:DeleteAlias",
        "lambda:UpdateFunctionCode",
        "lambda:UpdateFunctionConfiguration",
        "lambda:PutFunctionConcurrency",
        "lambda:DeleteFunctionConcurrency",
        "lambda:PublishVersion"
      ],
      "Resource": [
        "arn:aws:lambda:ap-south[redacted]unction:DoorDashDelivery"
      ]
    }
  ]
}
```



Name	AWS Region	IAM Access Analyzer	Creation date
doordashcodebuildbucket	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	April 27, 2024, 20:09:08 (UTC+05:30)

Let's go through the lambda_function.py and buildspec.yml files on the DoorDash repository:

Lambda_function:

a. It extracts the bucket name and object key from the event parameter when triggered.
b. Utilizing boto3, it retrieves the data of the JSON object, filters records with a "delivered" status, and converts them into a CSV format.
c. It uploads the CSV file to the target bucket using boto3.
d. In case of any errors during the upload process, it publishes a message to an SNS topic to notify the stakeholders.
e. Upon successful upload, it publishes a message to an SNS topic to inform stakeholders about the completion of the process.
f. Logs are utilized extensively throughout the function for monitoring and debugging purposes, providing insights into the execution flow and any encountered errors.

Buildspec:

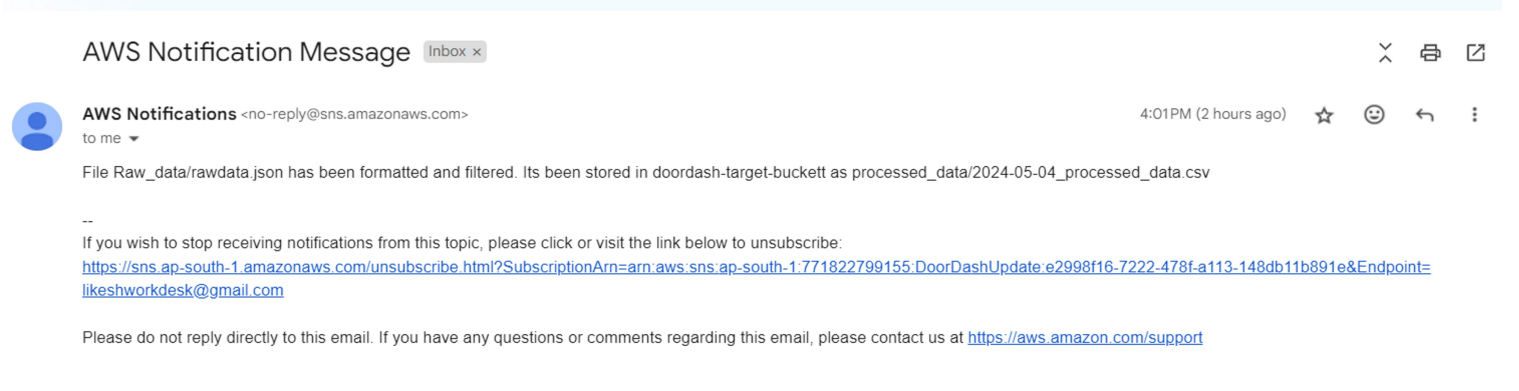
a. Version and Phases Declaration:
   - Version is set to 0.2.
   - Phases are defined: install, build, and post_build.
b. Install Phase:
   - Set the Python runtime version to 3.12.
   - Install dependencies listed in the requirements.txt file into the lib directory.
c. Build Phase:
   - Zip the contents of the lib directory to create a deployment package.
   - Add the lambda_function.py file to the deployment package.
d. Post_build Phase:
   - Upload the deployment package to the S3 bucket doordashcodebuildbucket.
   - Update the Lambda function DoorDashDelivery using the deployment package stored in the S3 bucket.
   - Print a message indicating the completion of the deployment process.

requirements.txt file contains all the dependencies used in the lambda_function.

After setting up everything, After merging changes to the main branch on GitHub, the configured CI/CD pipeline automatically initiates a code build process. The DoorDashDelivery lambda function gets updated.

Now, upload the 2024-04-28-raw_data.json file to the landing bucket. DoorDashDelivery lambda would be automatically triggered to process the data, and this is how the final notification

and logs look.



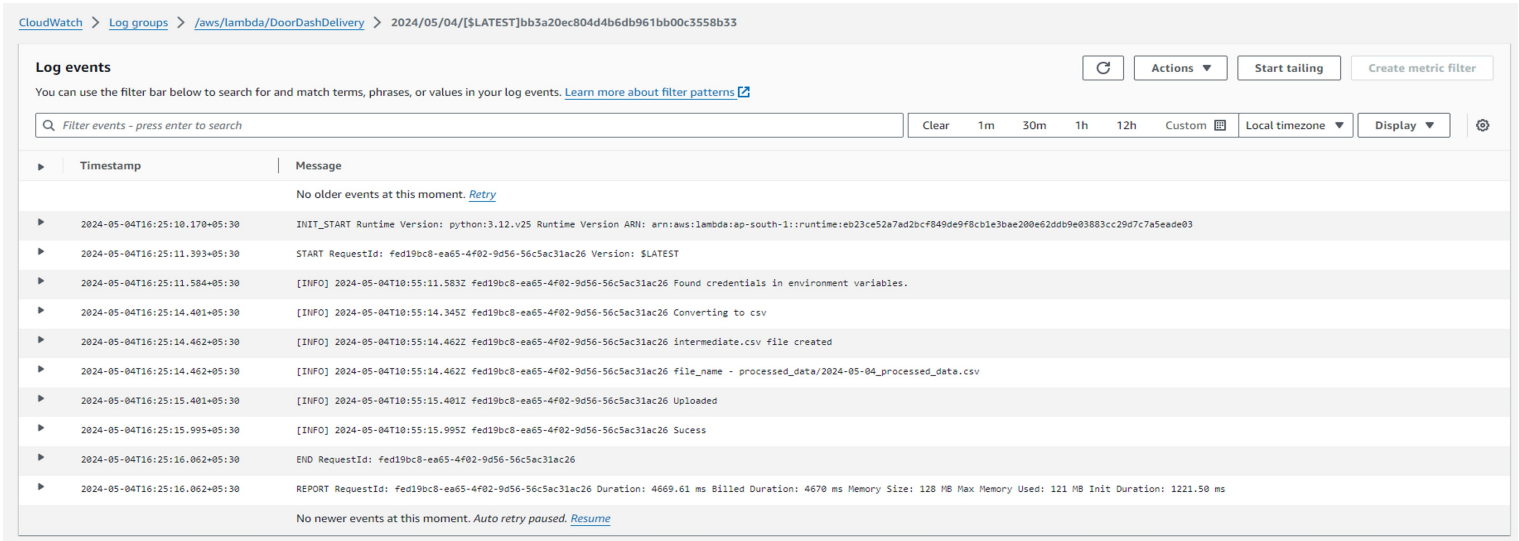
AWS Notification Message Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com> 4:01PM (2 hours ago)
to me

File Raw_data/rawdata.json has been formatted and filtered. Its been stored in doordash-target-buckett as processed_data/2024-05-04_processed_data.csv

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.ap-south-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:ap-south-1:771822799155:DoorDashUpdate:e2998f16-7222-478f-a113-148db11b891e&Endpoint=likeshworkdesk@gmail.com

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



CloudWatch > Log groups > /aws/lambda/DoorDashDelivery > 2024/05/04/[$LATEST]bb3a20ec804d4b6db961bb00c3558b33

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

Actions | Start tailing | Create metric filter

Filter events - press enter to search | Clear | 1m | 30m | 1h | 12h | Custom | Local timezone | Display

Timestamp	Message
	No older events at this moment. Retry
2024-05-04T16:25:10.170+05:30	INIT_START Runtime Version: python:3.12.v25 Runtime Version ARN: arn:aws:lambda:ap-south-1::runtime:eb23ce52a7ad2bcf849de9f8cb1e3bae200e62ddb9e03883cc29d7c7a5eade03
2024-05-04T16:25:11.393+05:30	START RequestId: fed19bc8-ea65-4f02-9d56-56c5ac31ac26 Version: $LATEST
2024-05-04T16:25:11.584+05:30	[INFO] 2024-05-04T10:55:11.583Z fed19bc8-ea65-4f02-9d56-56c5ac31ac26 Found credentials in environment variables.
2024-05-04T16:25:14.401+05:30	[INFO] 2024-05-04T10:55:14.345Z fed19bc8-ea65-4f02-9d56-56c5ac31ac26 Converting to csv
2024-05-04T16:25:14.462+05:30	[INFO] 2024-05-04T10:55:14.462Z fed19bc8-ea65-4f02-9d56-56c5ac31ac26 intermediate.csv file created
2024-05-04T16:25:14.462+05:30	[INFO] 2024-05-04T10:55:14.462Z fed19bc8-ea65-4f02-9d56-56c5ac31ac26 file_name - processed_data/2024-05-04_processed_data.csv
2024-05-04T16:25:15.401+05:30	[INFO] 2024-05-04T10:55:15.401Z fed19bc8-ea65-4f02-9d56-56c5ac31ac26 Uploaded
2024-05-04T16:25:15.995+05:30	[INFO] 2024-05-04T10:55:15.995Z fed19bc8-ea65-4f02-9d56-56c5ac31ac26 Sucess
2024-05-04T16:25:16.062+05:30	END RequestId: fed19bc8-ea65-4f02-9d56-56c5ac31ac26
2024-05-04T16:25:16.062+05:30	REPORT RequestId: fed19bc8-ea65-4f02-9d56-56c5ac31ac26 Duration: 4669.61 ms Billed Duration: 4670 ms Memory Size: 128 MB Max Memory Used: 121 MB Init Duration: 1221.50 ms
	No newer events at this moment. *Auto retry paused.* Resume