Project Documentation: S3-Triggered AWS Lambda Function

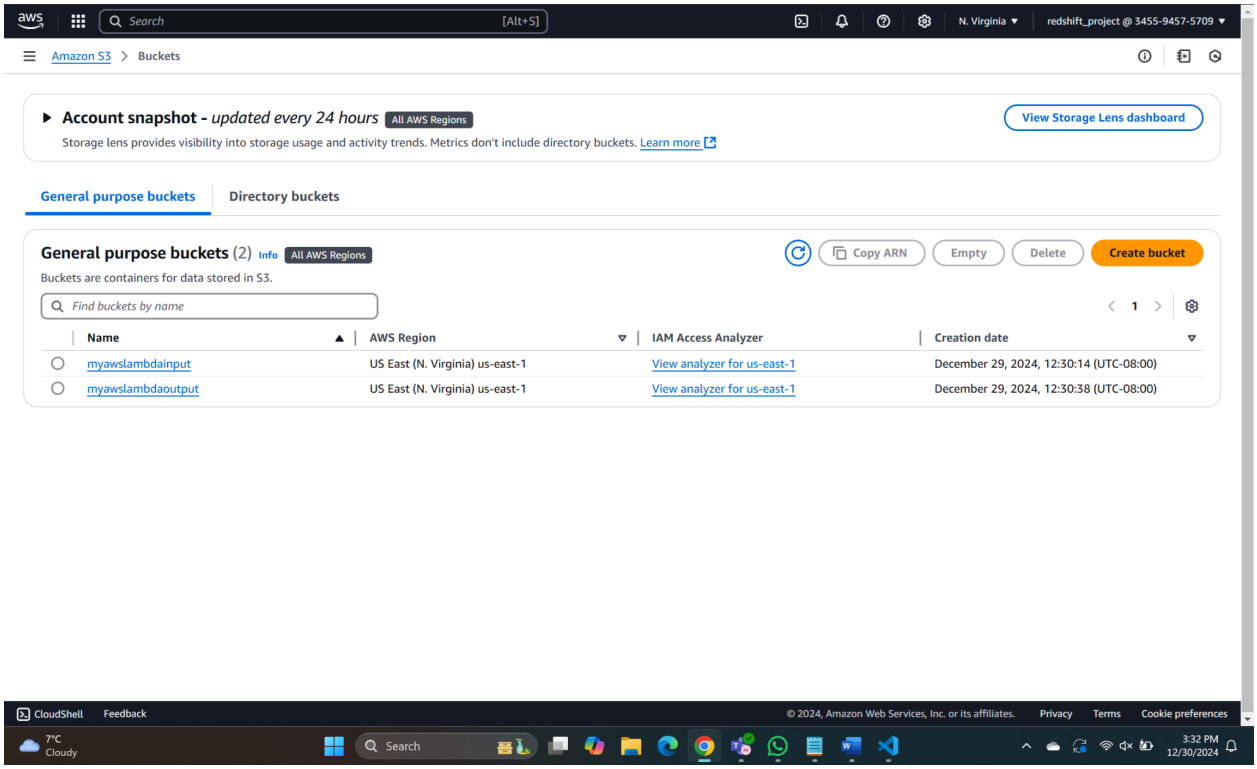
This project implements a Lambda function triggered by S3 file uploads. The setup includes:

- Two S3 buckets: one for input files and another for aggregated outputs.
- An IAM role granting permissions for the Lambda function.
- Configuring the Lambda function to process, aggregate, and store data in the destination bucket.

Creating S3 Buckets

Steps:

1. Open the AWS S3 Console.
2. Create a source bucket named `myawslambdainput`.
3. Create a destination bucket named `myawslambdaoutput`.
4. Ensure both buckets are configured without versioning or logging.

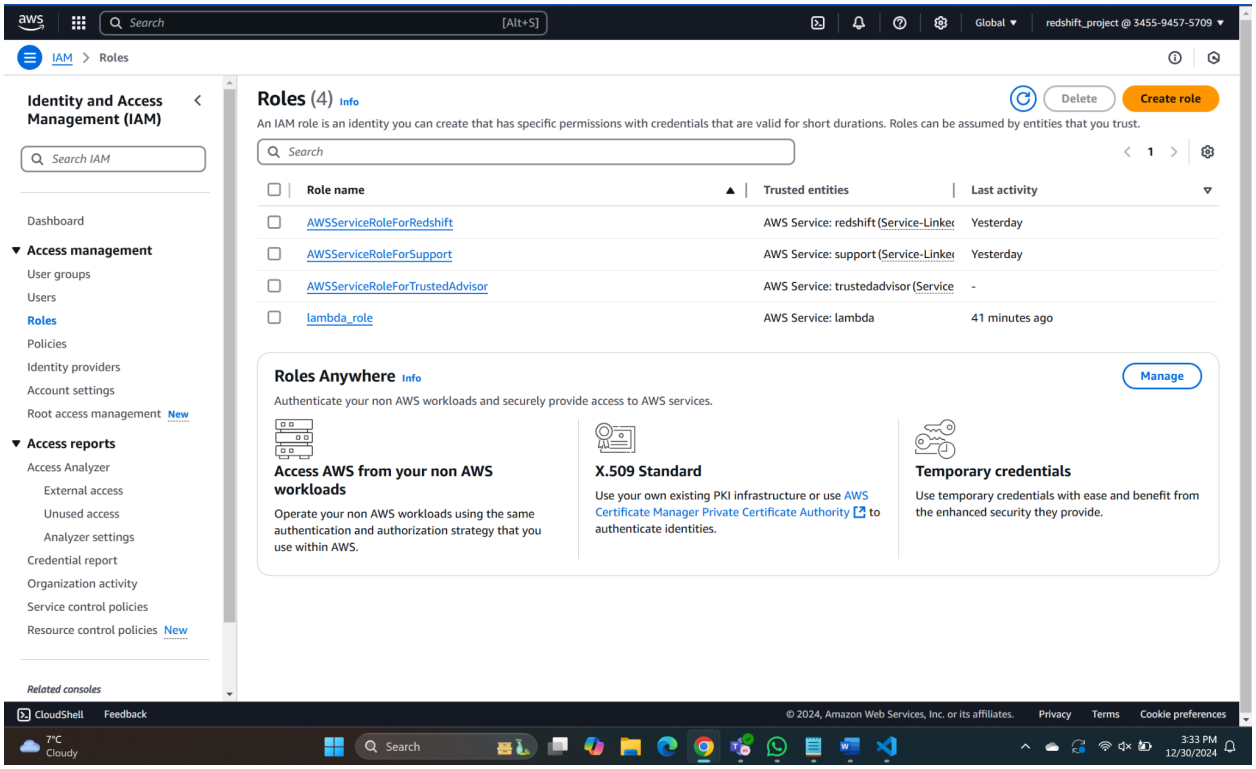


Setting up IAM Role

Steps:

1. Navigate to the AWS IAM Console.
2. Select **Roles** > **Create role**.

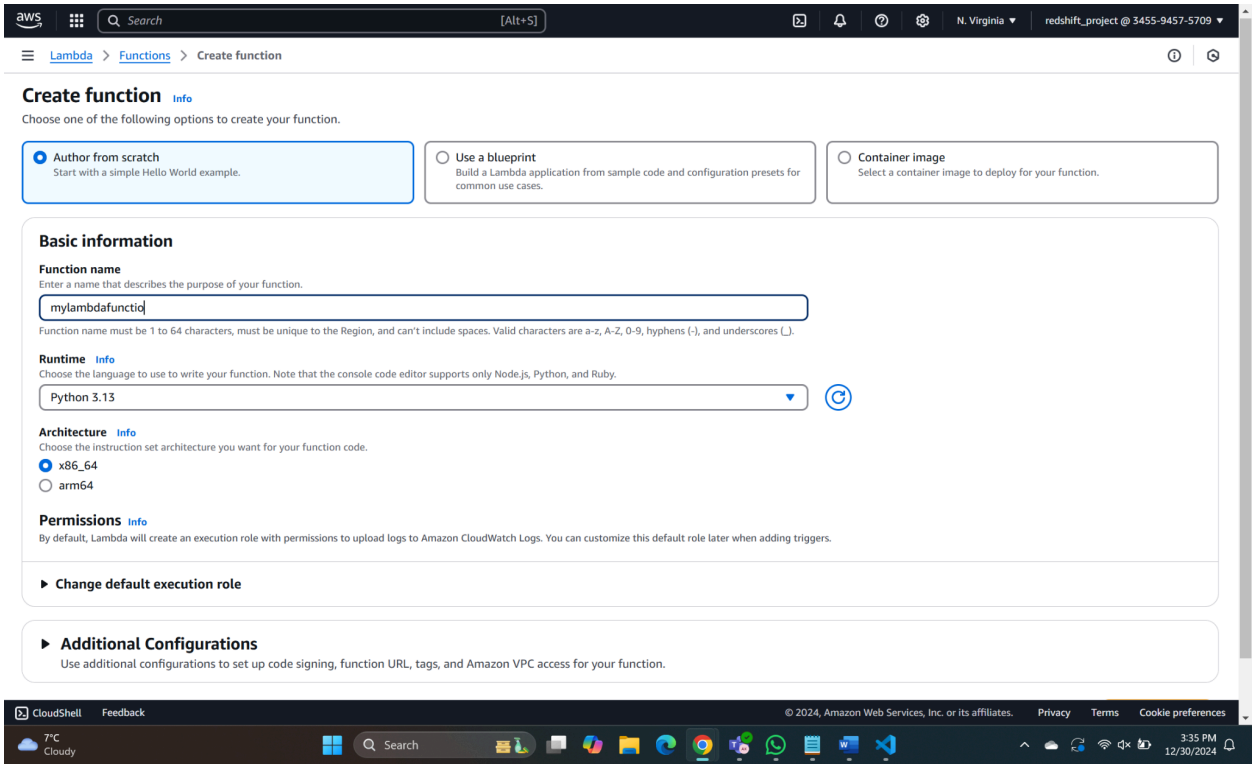
3. Choose **Lambda** as the trusted service.
4. Attach the following policies:
    - `AWSLambdaBasicExecutionRole`
    - `AmazonS3FullAccess`
5. Name the role **`lambda_role`**

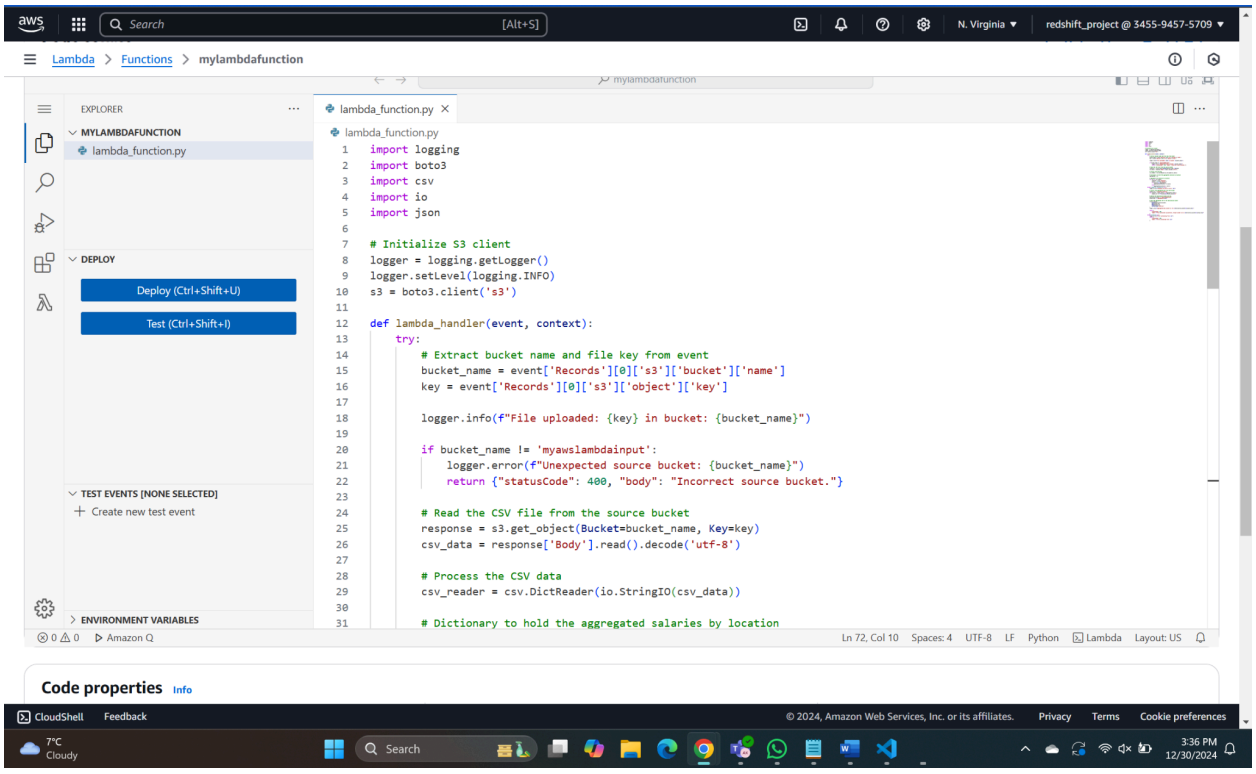


Creating the Lambda Function

Steps:

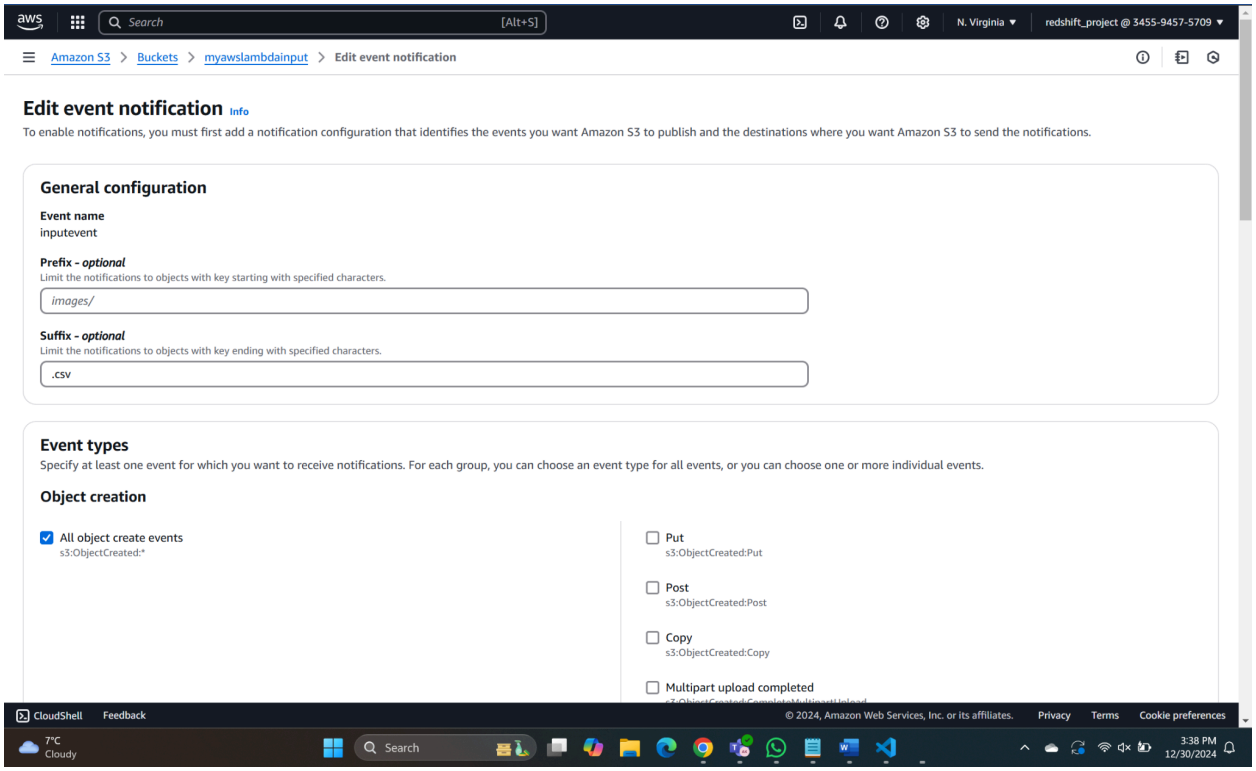
1. Open the AWS Lambda Console.
2. Select **Create function** > **Author from scratch**.
3. Name the function (e.g.,mylambdafunction) and choose Python 3.13 as the runtime.
4. Assign the IAM role **`lambda_role`**.
5. Write the Python code to:
    - Read CSV files uploaded to the source bucket.
    - Aggregate salaries by location.
    - Save results in the destination bucket.


aws
Search
[Alt+S]
N. Virginia
redshift_project @ 3455-9457-5709
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.
Basic information
Function name
Enter a name that describes the purpose of your function.
mylambdafunctio
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).
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 3.13
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
Additional Configurations
Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
7°C
Cloudy
Search
3:35 PM
12/30/2024


aws
Search
[Alt+S]
N. Virginia
redshift_project @ 3455-9457-5709
Lambda > Functions > mylambdafunction
EXPLORER
MYLAMBDAFUNCTION
lambda_function.py
DEPLOY
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)
TEST EVENTS [NONE SELECTED]
Create new test event
ENVIRONMENT VARIABLES
lambda_function.py
import logging
import boto3
import csv
import io
import json
# Initialize S3 client
logger = logging.getLogger()
logger.setLevel(logging.INFO)
s3 = boto3.client('s3')
def lambda_handler(event, context):
try:
# Extract bucket name and file key from event
bucket_name = event['Records'][0]['s3']['bucket']['name']
key = event['Records'][0]['s3']['object']['key']
logger.info(f"File uploaded: {key} in bucket: {bucket_name}")
if bucket_name != 'myawslambdainput':
logger.error(f"Unexpected source bucket: {bucket_name}")
return {"statusCode": 400, "body": "Incorrect source bucket."}
# Read the CSV file from the source bucket
response = s3.get_object(Bucket=bucket_name, Key=key)
csv_data = response['Body'].read().decode('utf-8')
# Process the CSV data
csv_reader = csv.DictReader(io.StringIO(csv_data))
# Dictionary to hold the aggregated salaries by location
Ln 72, Col 10
Spaces: 4
UTF-8
LF
Python
Lambda
Layout: US
Code properties Info
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
7°C
Cloudy
Search
3:36 PM
12/30/2024

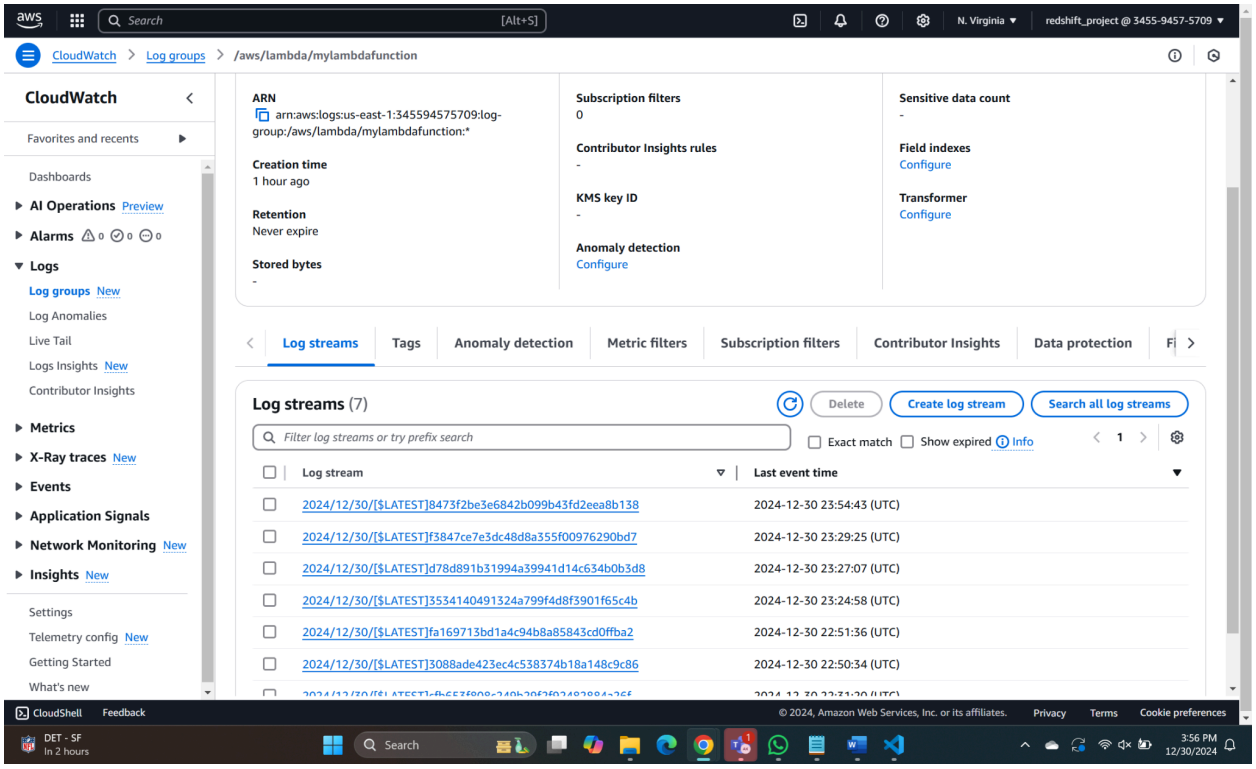
Configuring S3 Event Notifications

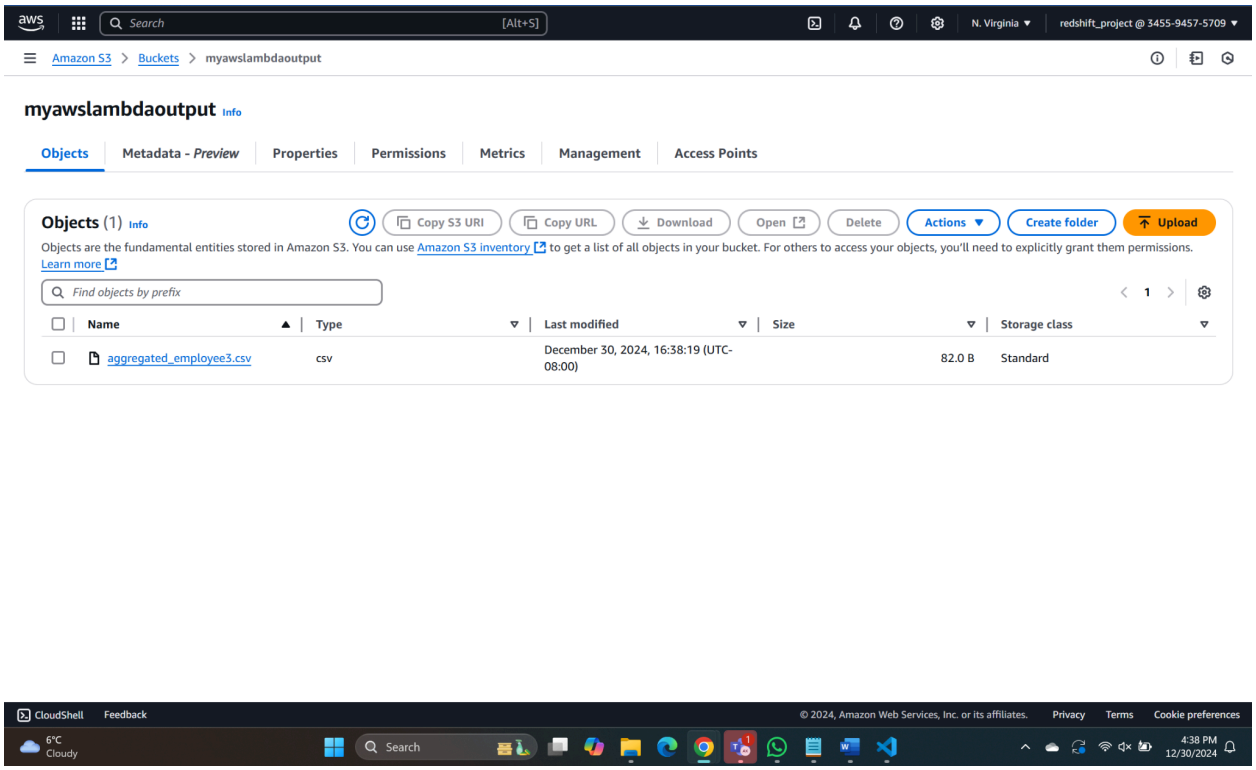
1. Open the `myawslambdainput` bucket in the S3 Console.
2. Go to **Properties** and configure an event notification:
    - Name: **NewFileUploadTrigger**
    - Event type: **All objects create events**
    - Destination: Select the created Lambda function (**`mylambdafunction`**).
3. Save the configuration.

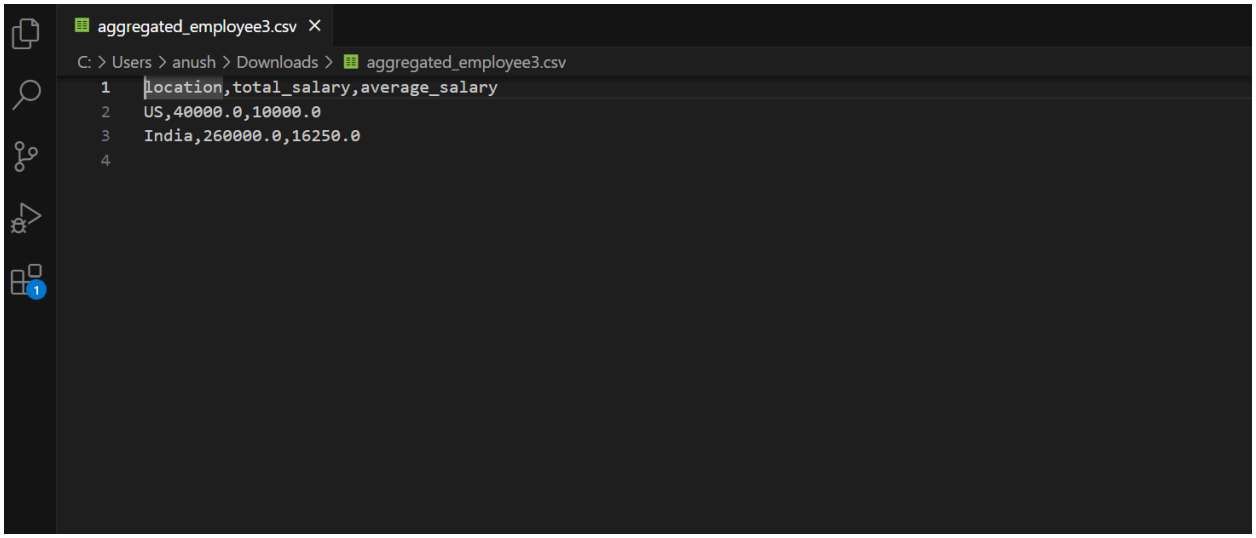


Testing the Solution

1. Upload a sample CSV file (`employee3.csv`) to `myawslambdainput`.
2. Check execution logs in CloudWatch to verify successful processing.
3. Confirm the aggregated file is stored in the destination bucket `myawslambdaoutput` under the `aggregated` folder.


aws
Search
[Alt+S]
N. Virginia
redshift_project @ 3455-9457-5709
CloudWatch
Log groups
/aws/lambda/mylambdafunction
CloudWatch
Favorites and recents
Dashboards
AI Operations Preview
Alarms
Logs
Log groups New
Log Anomalies
Live Tail
Logs Insights New
Contributor Insights
Metrics
X-Ray traces New
Events
Application Signals
Network Monitoring New
Insights New
Settings
Telemetry config New
Getting Started
What's new
ARN
arn:aws:logs:us-east-1:345594575709:log-group:/aws/lambda/mylambdafunction:*
Creation time
1 hour ago
Retention
Never expire
Stored bytes
-
Subscription filters
0
Contributor Insights rules
-
KMS key ID
-
Anomaly detection
Configure
Sensitive data count
-
Field indexes
Configure
Transformer
Configure
Log streams
Tags
Anomaly detection
Metric filters
Subscription filters
Contributor Insights
Data protection
Log streams (7)
Delete
Create log stream
Search all log streams
Filter log streams or try prefix search
Exact match
Show expired
Info
Log stream
Last event time
2024/12/30/[$LATEST]8473f2be3e6842b099b43fd2eea8b138
2024-12-30 23:54:43 (UTC)
2024/12/30/[$LATEST]f3847ce7e3dc48d8a355f00976290bd7
2024-12-30 23:29:25 (UTC)
2024/12/30/[$LATEST]d78d891b31994a39941d14c634b0b3d8
2024-12-30 23:27:07 (UTC)
2024/12/30/[$LATEST]3534140491324a799f4d8f3901f65c4b
2024-12-30 23:24:58 (UTC)
2024/12/30/[$LATEST]fa169713bd1a4c94b8a85843cd0ffba2
2024-12-30 22:51:36 (UTC)
2024/12/30/[$LATEST]3088ade423ec4c538374b18a148c9c86
2024-12-30 22:50:34 (UTC)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
DET - SF
in 2 hours
Search
3:56 PM
12/30/2024


aws
Search
[Alt+S]
N. Virginia
redshift_project @ 3455-9457-5709
Amazon S3
Buckets
myawslambdaoutput
myawslambdaoutput Info
Objects
Metadata - Preview
Properties
Permissions
Metrics
Management
Access Points
Objects (1) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.
Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
aggregated_employee3.csv
csv
December 30, 2024, 16:38:19 (UTC-08:00)
82.0 B
Standard
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
6°C
Cloudy
Search
4:38 PM
12/30/2024



Validation

1. Download the aggregated file from the destination bucket for review.
2. Verify the aggregated data aligns with expectations.

Conclusion

This setup demonstrates the automation of processing and aggregation of CSV files using AWS Lambda and S3. It is scalable for processing multiple files and serves as a reliable foundation for similar workflows.