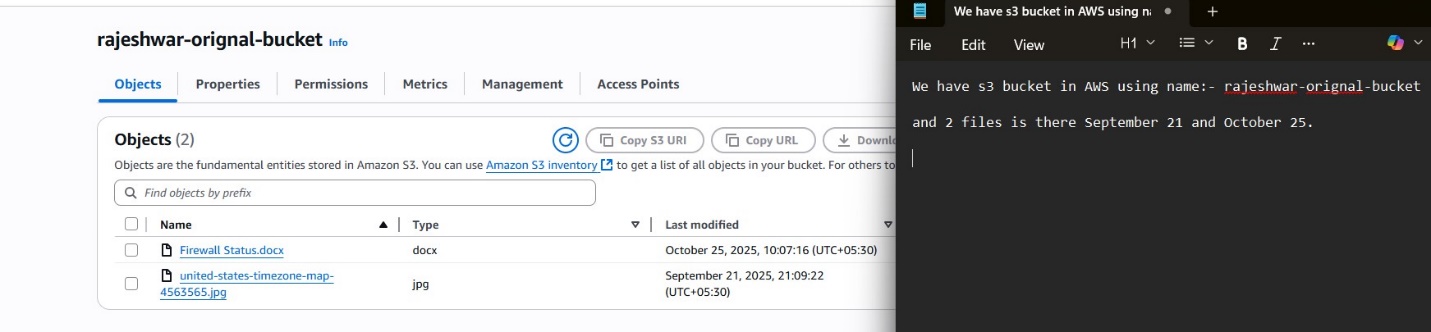
Assignment:- We have s3 bucket in AWS using name:- rajeshwar-orignal-bucket

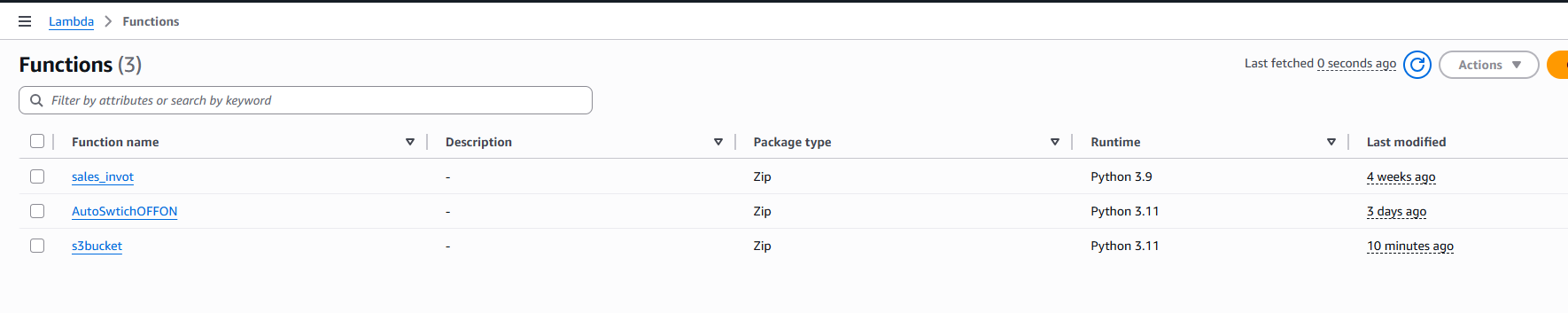
1. **Create bucket and upload 2 files as per below pic.**

and 2 files is there September 21 and October 25.

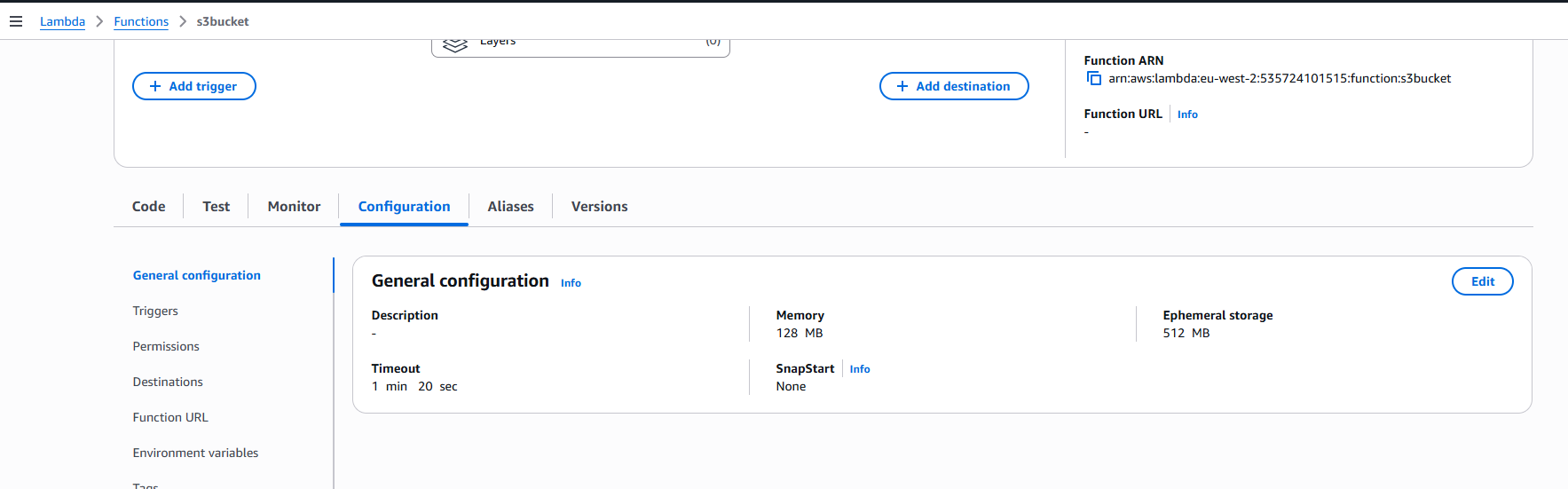
As per pic 1



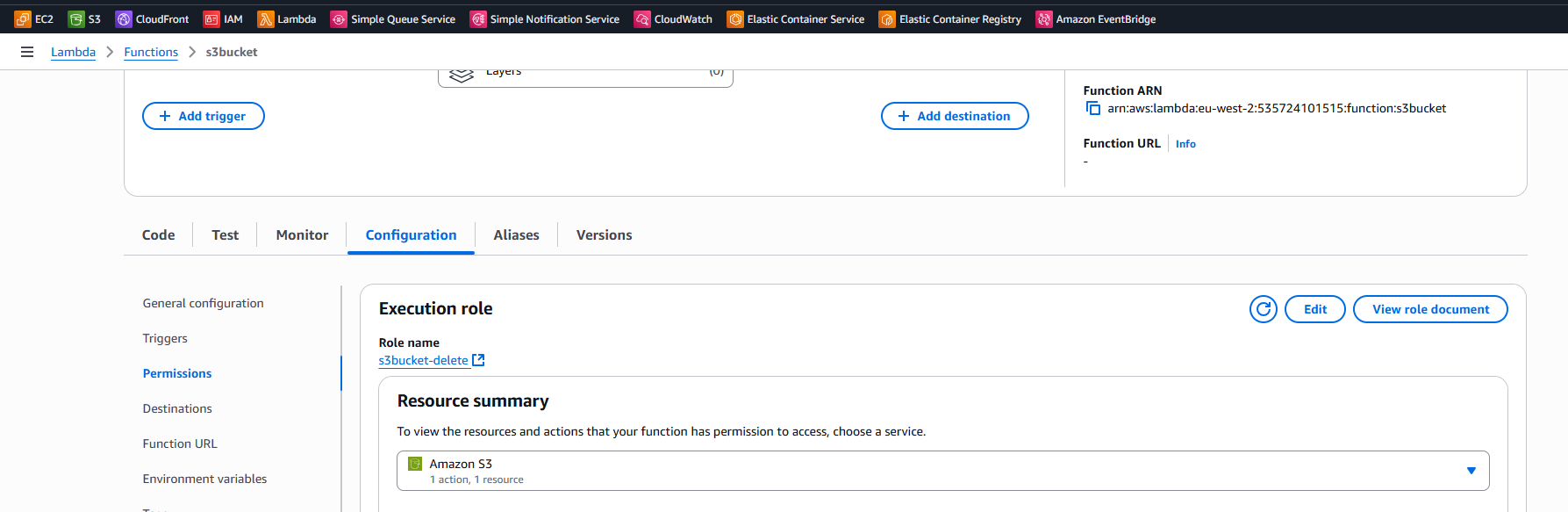
1. **Create Lambda function to delete files older than 30 days.**
2. Here we create lambda function name “s3bucket”.



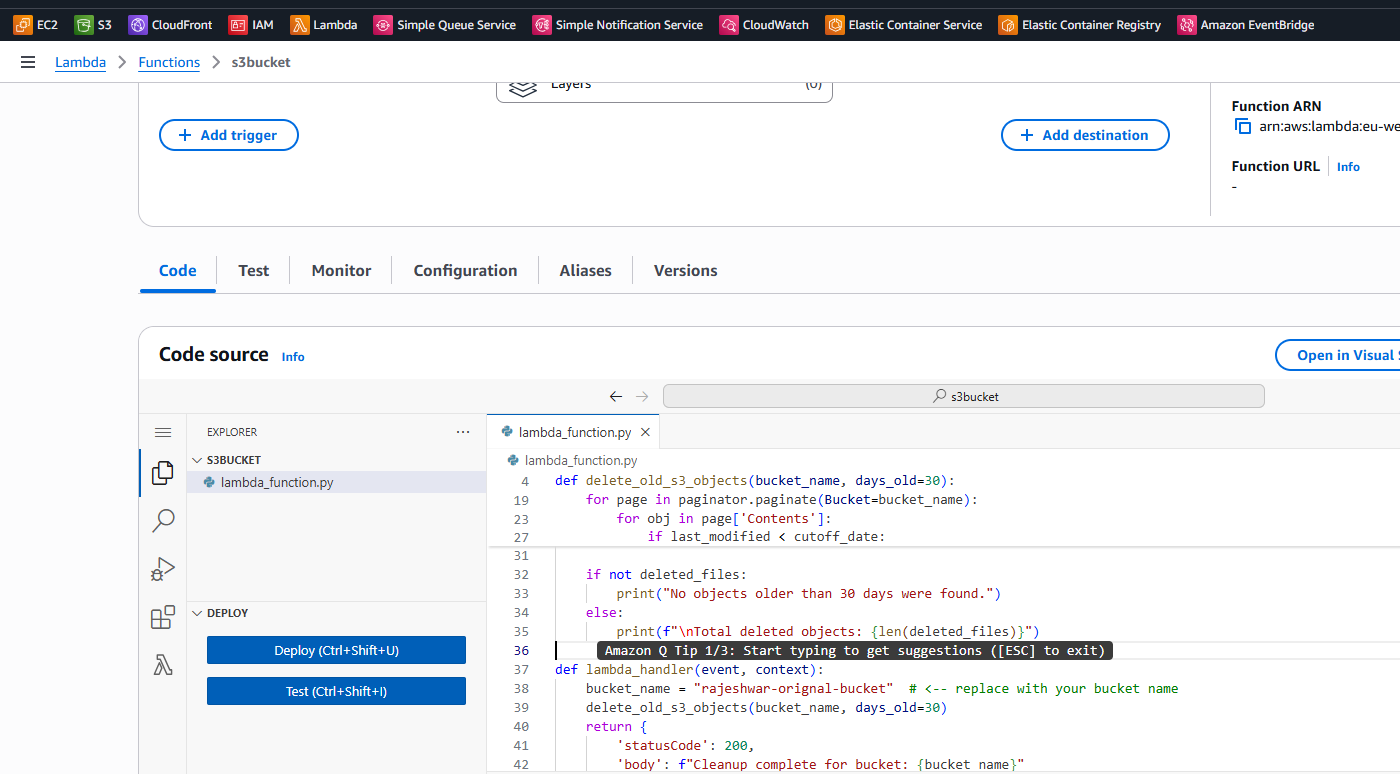
1. Change in configuration time “1.30 Minutes”



1. Change in permission add role “s3bucket-delete”.



1. Here we add the code for delete 30 days older files.



Below is the code use for delete 30 days old files.

import boto3

from datetime import datetime, timezone, timedelta

def delete\_old\_s3\_objects(bucket\_name, days\_old=30):

    """

    Deletes objects older than 'days\_old' days from the specified S3 bucket.

    Prints names of deleted objects for logging.

    """

    s3 = boto3.client('s3')

    cutoff\_date = datetime.now(timezone.utc) - timedelta(days=days\_old)

    print(f"Scanning bucket: {bucket\_name}")

    print(f"Deleting objects older than: {cutoff\_date}")

    # Use paginator for large buckets

    paginator = s3.get\_paginator('list\_objects\_v2')

    deleted\_files = []

    for page in paginator.paginate(Bucket=bucket\_name):

        if 'Contents' not in page:

            continue

        for obj in page['Contents']:

            key = obj['Key']

            last\_modified = obj['LastModified']

            if last\_modified < cutoff\_date:

                s3.delete\_object(Bucket=bucket\_name, Key=key)

                deleted\_files.append(key)

                print(f"Deleted: {key}")

    if not deleted\_files:

        print("No objects older than 30 days were found.")

    else:

        print(f"\nTotal deleted objects: {len(deleted\_files)}")

def lambda\_handler(event, context):

    bucket\_name = "rajeshwar-orignal-bucket"  # <-- replace with your bucket name

    delete\_old\_s3\_objects(bucket\_name, days\_old=30)

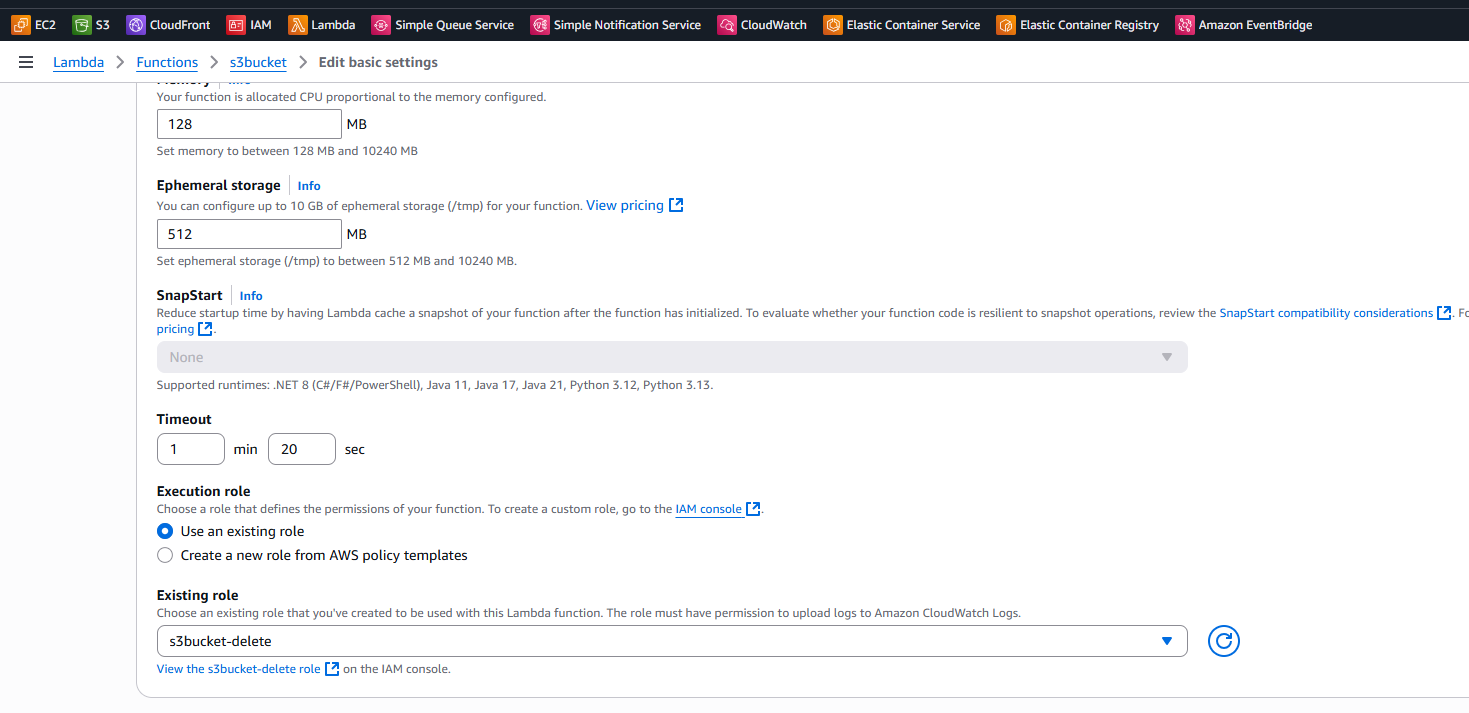
    return {

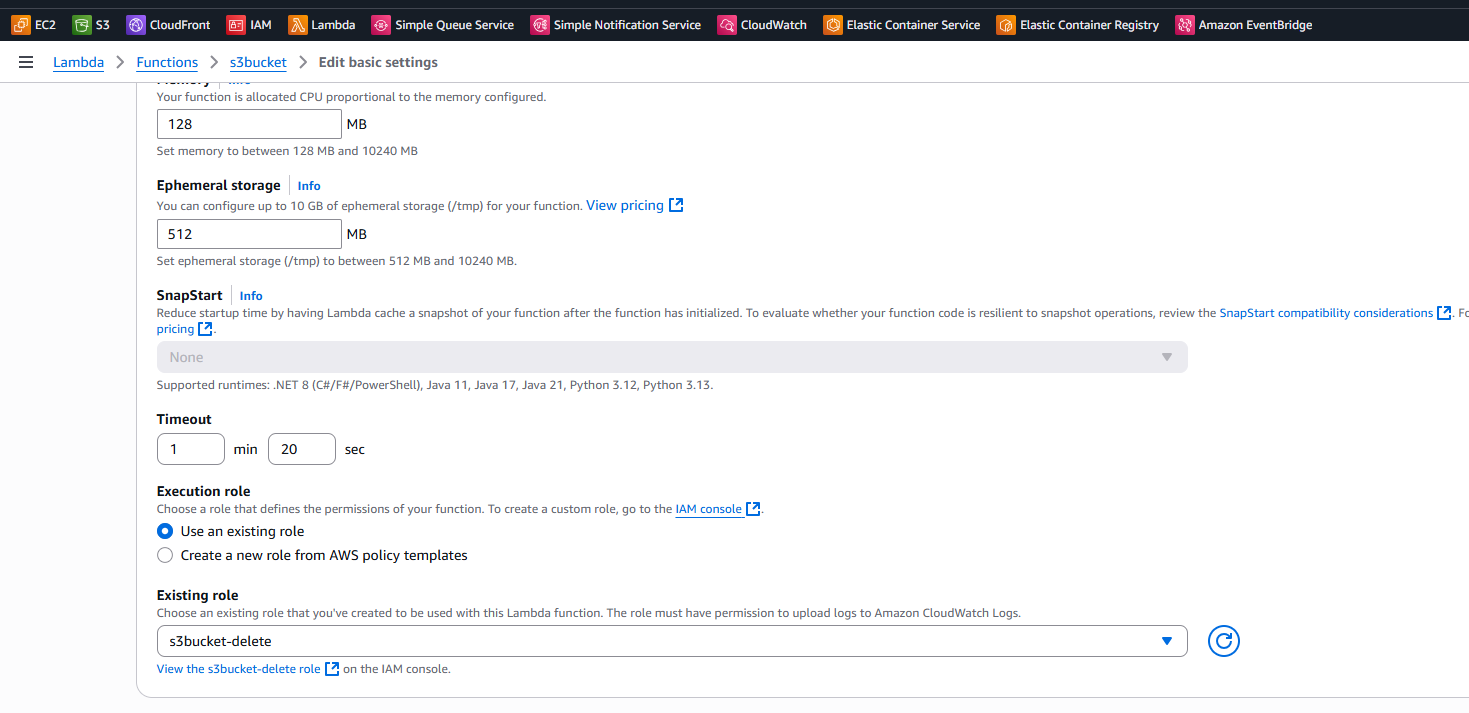
        'statusCode': 200,

        'body': f"Cleanup complete for bucket: {bucket\_name}"

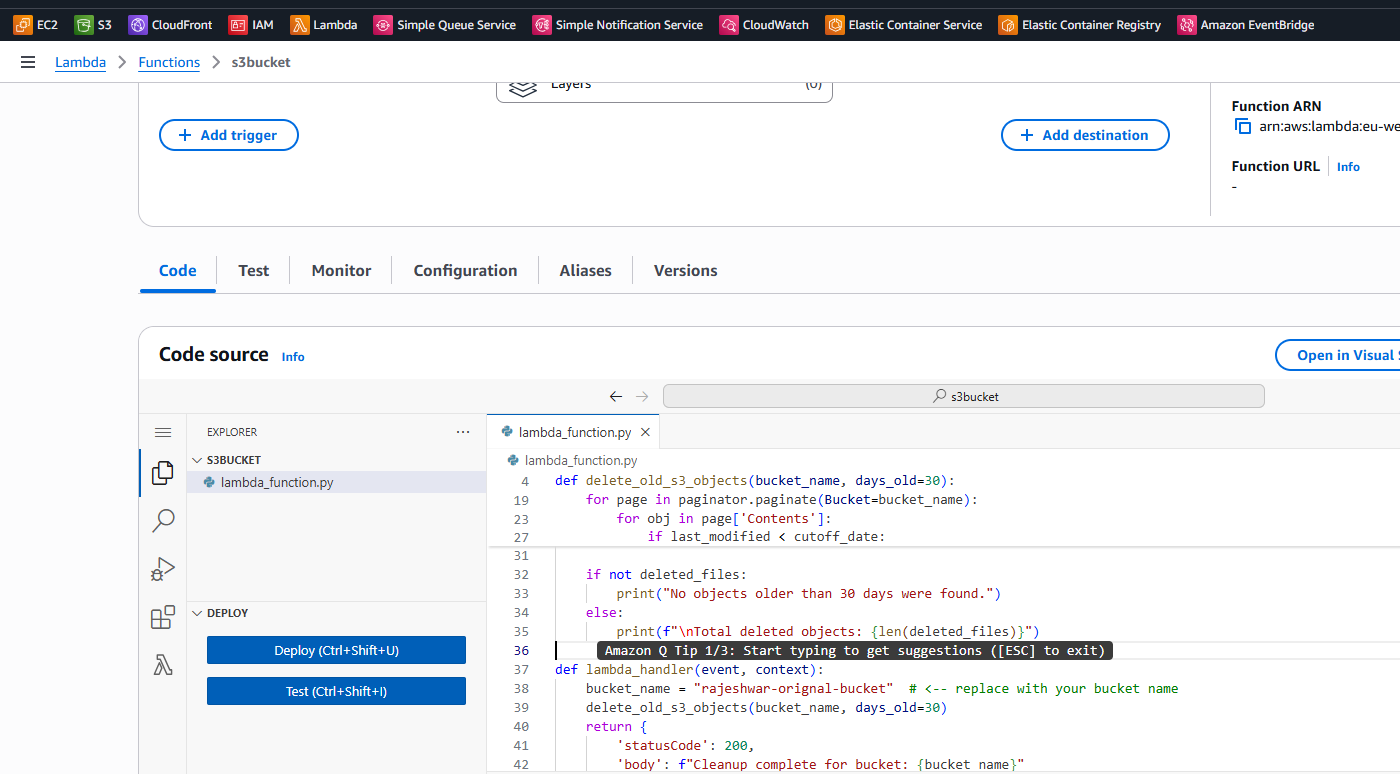
    }

1. **Go on IAM roles to create role with “AmazonS3FullAccess”. I have created role “s3bucket-delete” to add in lambda function permission.**





1. Test the code for work ok or not.



Click on Deploy.

Then click on “Test”.

If your function is working perfectly output status show succeeded.

1. Initialize a boto3 S3 client.

2. Delete objects older than 30 days.

3. Print the names of deleted objects for logging purposes

Output should be:-

Status: Succeeded

Test Event Name: s3bucket

Response:

{

"statusCode": 200,

"body": "Cleanup complete for bucket: rajeshwar-orignal-bucket"

}

Function Logs:

START RequestId: f3a41012-1314-405e-9bbf-aa9bd30e0ec5 Version: $LATEST

Scanning bucket: rajeshwar-orignal-bucket

Deleting objects older than: 2025-09-25 05:02:20.207987+00:00

Deleted: united-states-timezone-map-4563565.jpg

Total deleted objects: 1

END RequestId: f3a41012-1314-405e-9bbf-aa9bd30e0ec5

REPORT RequestId: f3a41012-1314-405e-9bbf-aa9bd30e0ec5 Duration: 3495.92 ms Billed Duration: 3719 ms Memory Size: 128 MB Max Memory Used: 91 MB Init Duration: 222.78 ms

Request ID: f3a41012-1314-405e-9bbf-aa9bd30e0ec5

