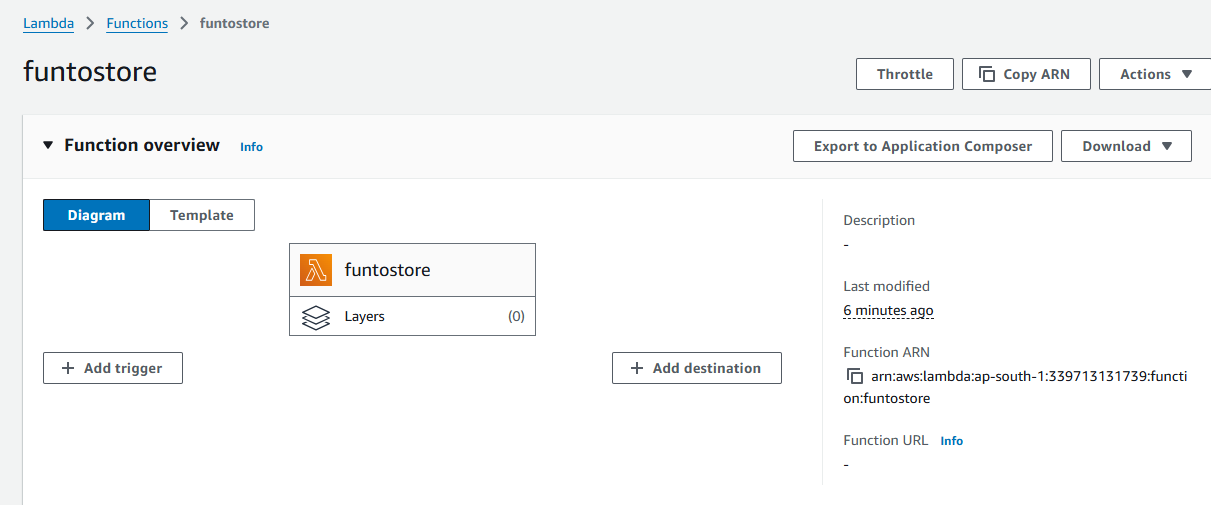


GIve permission for lambda to access s3 and dynamoDB



Now code and deploy

**Python code to run in lambda to store csv in dynamodb table**

import boto3

import csv

import os

import logging

# Initialize clients for S3 and DynamoDB

s3\_client = boto3.client('s3')

dynamodb\_client = boto3.resource('dynamodb')

table\_name = os.getenv('TABLE\_NAME') # Get DynamoDB table name from environment variable

bucket\_name = os.getenv('BUCKET\_NAME') # Get S3 bucket name from environment variable

table = dynamodb\_client.Table(table\_name)

# Set up logging

logger = logging.getLogger()

logger.setLevel(logging.INFO)

def lambda\_handler(event, context):

# Specify the CSV file key (update this with the actual path to your CSV file)

csv\_file\_key = 'data/myfile.csv' # Change to the actual CSV file path

try:

# Get the CSV file from S3

csv\_content = s3\_client.get\_object(Bucket=bucket\_name, Key=csv\_file\_key)['Body'].read().decode('utf-8')

# Parse CSV content

csv\_reader = csv.DictReader(csv\_content.splitlines())

# Insert each row into DynamoDB

with table.batch\_writer() as batch:

for row in csv\_reader:

batch.put\_item(Item=row)

logger.info(f"CSV file '{csv\_file\_key}' from bucket '{bucket\_name}' has been processed and stored in DynamoDB.")

except Exception as e:

logger.error(f"Error processing file {csv\_file\_key} from bucket {bucket\_name}: {e}")

return {

'statusCode': 500,

'body': f"Error processing the file '{csv\_file\_key}'."

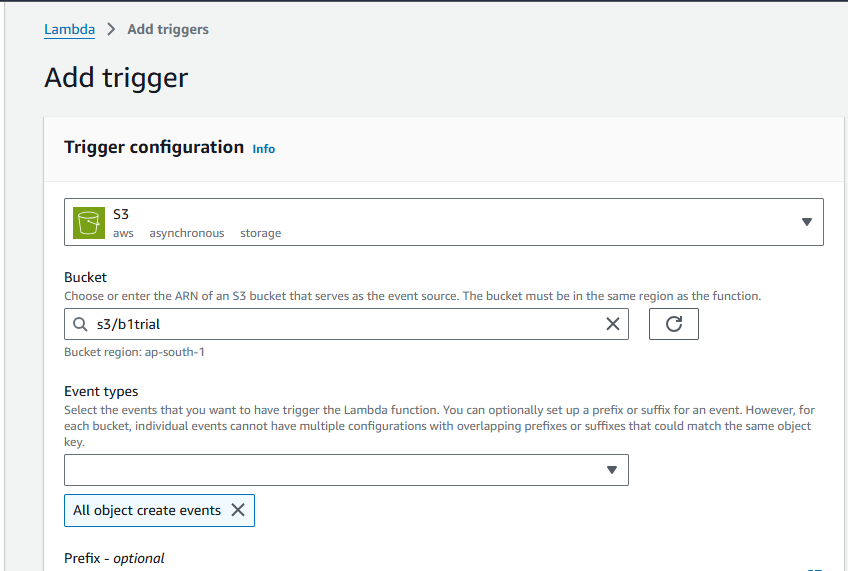
}

return {

'statusCode': 200,

'body': f"Processed CSV file '{csv\_file\_key}' and stored in DynamoDB table '{table\_name}'."

}



Add trigger to to run lambda