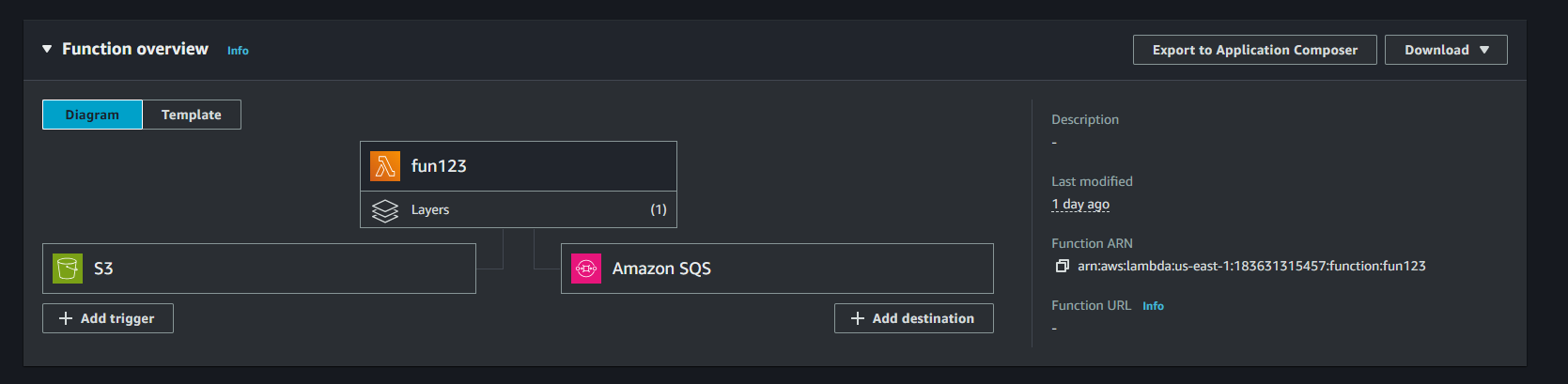
This lambda function is written in python to be triggered when there is a “PUT” event type in the s3 bucket, the function takes the order from the bucket and sends it to a sqs queue.



import json

import boto3

import time

def lambda\_handler(event, context):

s3 = boto3.client('s3')

sqs = boto3.client('sqs')

sqs\_queue\_url = "https://sqs.us-east-1.amazonaws.com/183631315457/labq"

print("Event received:", json.dumps(event))

if 'Records' not in event:

print("No 'Records' found in the event.")

return {

'statusCode': 400,

'body': json.dumps('No records found in the event.')

}

for record in event['Records']:

if 's3' not in record or 'bucket' not in record['s3'] or 'name' not in record['s3']['bucket']:

print("S3 bucket information is missing.")

return {

'statusCode': 400,

'body': json.dumps('S3 bucket information is missing.')

}

if 'object' not in record['s3'] or 'key' not in record['s3']['object']:

print("S3 object key information is missing.")

return {

'statusCode': 400,

'body': json.dumps('S3 object key information is missing.')

}

bucket = record['s3']['bucket']['name']

key = record['s3']['object']['key']

try:

start\_time = time.time()

print(f"Fetching object from S3 bucket: {bucket}, key: {key}")

response = s3.get\_object(Bucket=bucket, Key=key)

fetch\_duration = time.time() - start\_time

print(f"S3 fetch duration: {fetch\_duration:.2f} seconds")

file\_content = response['Body'].read().decode('utf-8')

order = json.loads(file\_content)

# Display the order content

print("Order content:")

print(json.dumps(order, indent=4))

# Send the order to SQS

sqs\_response = sqs.send\_message(

QueueUrl=sqs\_queue\_url,

MessageBody=json.dumps(order)

)

print(f"Message sent to SQS: {sqs\_response['MessageId']}")

except Exception as e:

print(f"Error processing order: {str(e)}")

return {

'statusCode': 500,

'body': json.dumps('Error processing order')

}

return {

'statusCode': 200,

'body': json.dumps('Order processed and sent to SQS successfully!')

}