

基于无服务器的数据表增量同步Prototyping

- 部署成功过，该项目是prototyping, 请在环境中大规模反复测试
- 请务必根据自身环境修改各种配置变量和review code

角色权限

- 在iam创建一个角色，名字任意，假设名字叫做 lambda-data-analystist-role
- 分配下图权限

权限策略 (4) 您最多可以附加 10 个托管策略。				刷新	模拟	移除	添加权限
按属性或策略名称筛选策略，然后按下“Enter”				< 1 > 设置			
<input type="checkbox"/>	策略名称	类型	描述				
<input type="checkbox"/>	AmazonS3FullAccess	亚马逊云科技 托管	Provides full access to all bu				
<input type="checkbox"/>	AWSLambdaBasicExecutionRole	亚马逊云科技 托管	Provides write permissions t				
<input type="checkbox"/>	AmazonSQSFullAccess	亚马逊云科技 托管	Provides full access to Ama				
<input type="checkbox"/>	AmazonAthenaFullAccess	亚马逊云科技 托管	Provide full access to Amaz				

准备工作

- 需要运维同学创建好一个sqs的标准队列，配置取默认即可。请将sqs的url 给数据部门的同事（马经理）
- 数据部门的同事请下载如下代

```
1 git clone https://github.com/tx-customer/yunli-incremental-data.git
```

- 请马经理 修改 里面两个py文件。请根据绿色注释进行修改

```

# 需要修改配置的地方=====
# 需要运维同学创建一个 SQS。 sqs的url地址。 和 s3_listener.py中的是同一个
sqs_url = "https://sqs.cn-northwest-1.amazonaws.com.cn/027040934161/yunlidemo"
# 临时表的s3地址
target_s3 = "s3://example-output/yunli/json_data/"
# athena 中, 你的数据表放在那个db 下, 那个目录下。在athena web 界面能看到
athena_data_ctx = {'Database': 'default', 'Catalog': 'AwsDataCatalog'}

# athena 的 log地址
athena_output_cfg = {
    'OutputLocation': 's3://aws-glue-assets-027040934161-cn-northwest-1/'
}

# 目标表表名
target_table = "demo_prod_yunli_athena_tb"
# 临时表表名
source_temp_table = "demo_yunli_athena_tb3"
# athena 工作组
athena_work_group = 'primary'
# =====

```

```

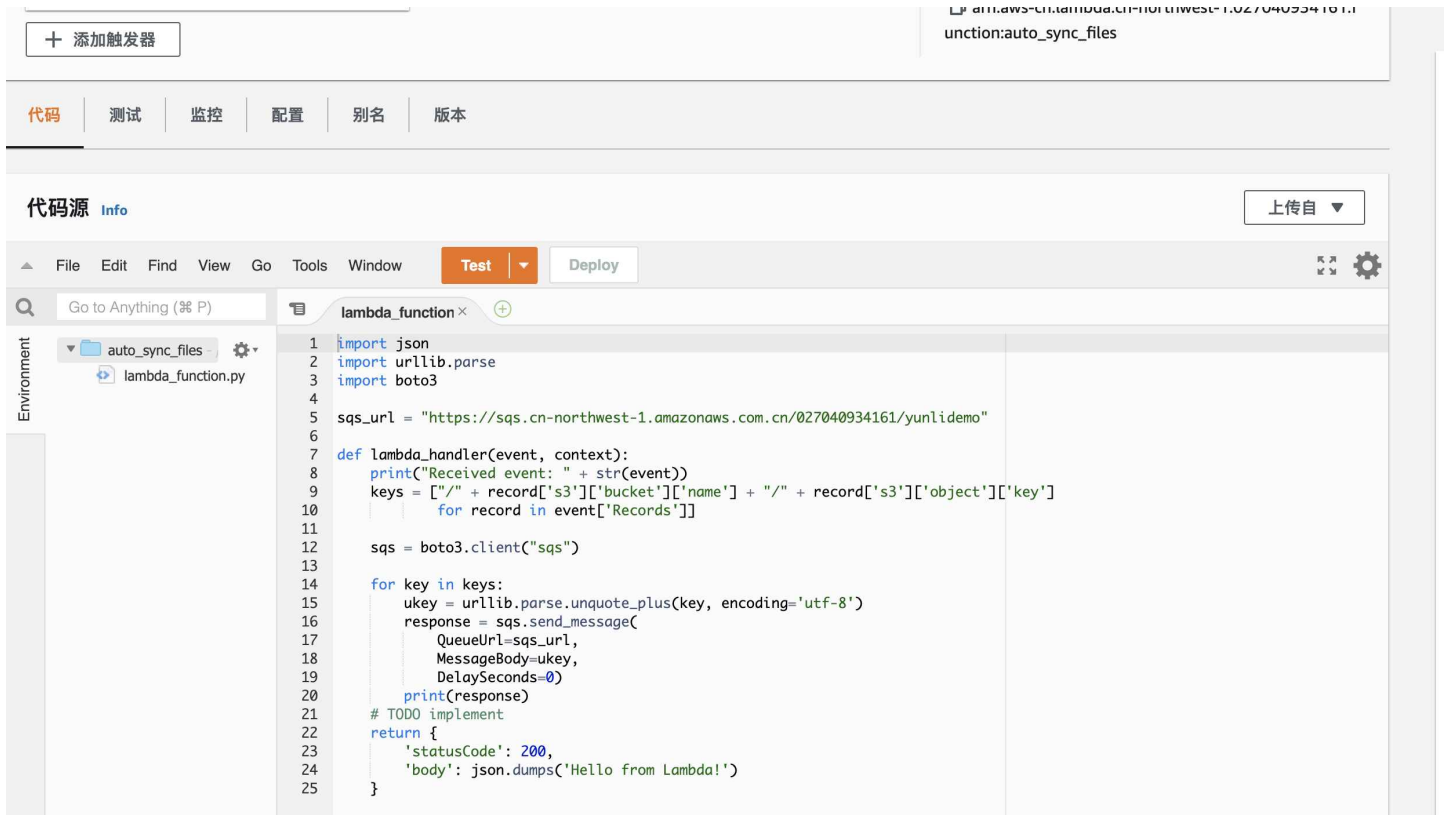
# 需要修改配置的地方=====
# 需要运维同学创建一个 SQS。 sqs的url地址。 和 sync_data.py中的是同一个
sqs_url = "https://sqs.cn-northwest-1.amazonaws.com.cn/027040934161/yunlidemo"
# =====

```

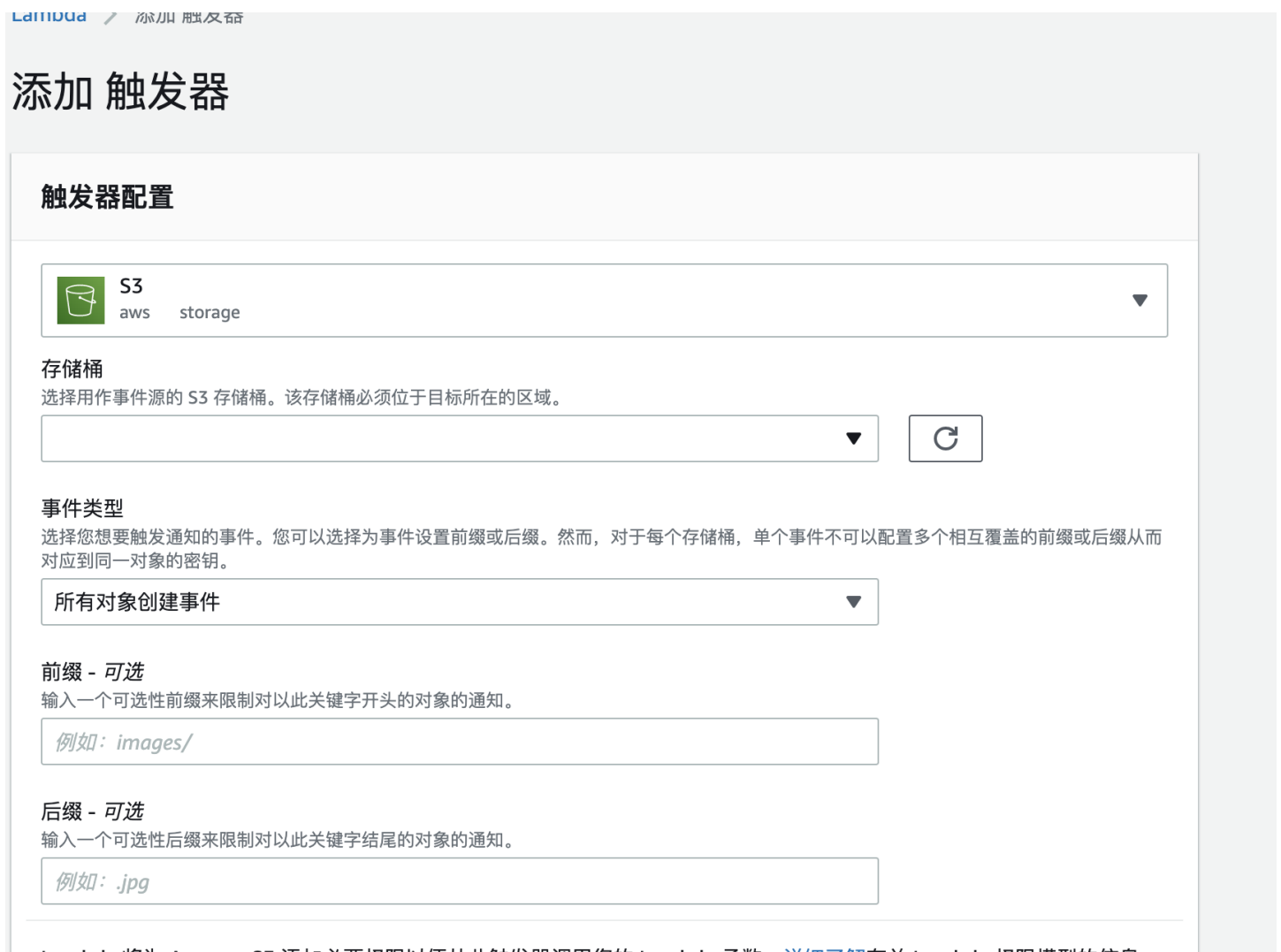
4. 将修改好的代码交给运维同学

部署监听新增数据的lambda函数

1. 请运维同学创建一个名叫 {根据业务或部门自定义前缀}_s3_listener的 lambda函数。例如 data_analytic_log_s3_listener。其实名字不重要，根据贵司的命名规则就好
2. 请将s3_listener.py的代码粘贴到该lambda的函数里。如果不会创建，[请参考这里有创建流程](#)



3. 点击 添加触发器按钮，选择s3作为事件源



4. 存储桶就是要监听的s3桶，请仔细马经理。这个桶应该就是要存放从oss同步过来数据的桶。

根据需要可以添加前缀来过滤一些不需要监听的子文件夹。

5. 添加触发器成功够，请点击右边导航栏里的权限进行权限配置，使用前面创建好的角色 lambda-data-analystist-role

6. 常规配置里，配置超时时间 为 1 分钟

常规配置

触发器

权限

目标

常规配置 [Info](#)

描述

-

内存

128 MB

超时

1 分钟 0 秒

编辑

7. 保存并部署


部署同步新增数据的lambda

1. 请运维同学创建一个名叫 {根据业务或部门自定义前缀}_s3_to_athena的 lambda函数。例如 data_analystic_log_s3_to_athena.其实名字不重要，根据贵司的命名规则就好
2. 请将代码文件sync_data.py中的代码粘贴到该lambda函数中
3. 添加触发器，选择SQS. SQS队列名称是前面创建的那个，可以直接在下拉框里选择

Lambda > 添加 触发器

添加 触发器

触发器配置

 SQS

aws queue

▼

SQS 队列

SQS

选择或输入 SQS 队列的 ARN。

Q

↻

批处理大小

单个批次中要检索的消息最大数量。

10

▶ 其他设置 - optional

要从 SQS 触发器中进行读取，您的执行角色必须具有适当权限。

☒

取消

添加

4. 添加触发器成功够，请配置权限，使用前面创建好的角色 lambda-data-analystist-role

5. 配置并发，并发配置为 1

并发

编辑

函数并发	预留并发
使用预留并发	1

6. 常规配置里，配置超时时间 为15分钟

常规配置 Info

编辑

描述	内存	超时
-	128 MB	15 分钟 0 秒

7. 保存并部署

测试

- 1. 可以手动在被监听的s3桶中，按oss同步的文件夹路径，上传一个或多个规定格式的tar.gz的包
- 2. 大概等三四分钟，查询一下目标表里的数据是否有新增

！ 请务必根据自身环境修改各种配置变量和review code

！ 部署成功过，该项目是prototyping, 请在环境中大规模反复测试

增强&修正

防止数据重复消费

由于dth或者s3 trigger都是通过 putObject 方式在s3上添加新文件，所以如果数据源重复添加相同的文件，会出现重复消费

更新步骤：

创建dynamoDB table.

- 1. 创建dynamoDB table. 假设表名为：yunli-s3-athena-status
- 2. 创建表的过程中，分区键 key 为 keyetag
- 3. 点击保存

创建更新角色权限

1. 在iam上对当前lambda函数使用的角色添加对新建的dynamoDB table 读写权限

修改代码中的配置参数

1. 马经理需要对代码中这两个配置修正（以前修正的地方也要保留）
2. s3_listener.py 和 sync_data.py都有这两个新增参数

```
dynamodb_status_table = 'yunli-s3-athena-status'
dynamodb_status_table_key = 'keyetag'
# =====

def lambda_handler(event, context):
```

3. 更新两个lambda函数
4. 测试