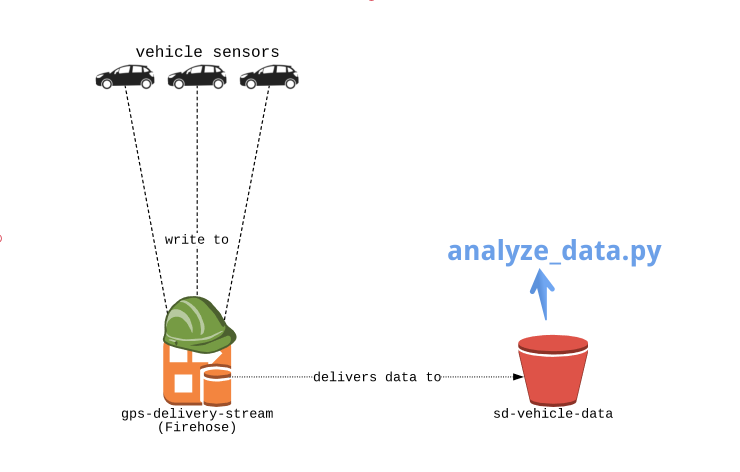
At the end of last chapter, you wrote a method to read data from S3 - analyze\_data. However, you had to execute that method manually.



We're going to write a Lambda function that runs every time Firehose writes a new file to S3. For now, it only prints out the files that were added.

You will be editing the recordReaderS3/lambda\_function.py file in the editor. This contains the handler for our Lambda function.

Don't forget: an object being written to S3 results in an ObjectCreated:Put event. You can always click on the **slides** icon in the top right to see the slides - they'll be pretty helpful!

**Instructions**

**100XP**

* Update the lambda handler method (recordReaderS3/lambda\_function.py) to accept the right arguments.
* Parse the incoming event. If the event is an S3 write event, and is in the "sd-vehicle-data", print the new object key.
* In the terminal, run python3 run\_lambda.py to deploy and test your lambda function.

Code:

# EDIT ME!

# recordReaderS3/lambda\_function.py

# This is lambda handler code that will execute each time an object is added to S3.

import json

def record\_created\_handler(event, context):

# Create a list to store new object keys

written\_objects = []

# Iterate over each S3 event record

for record in event['Records']:

# Get the variables to check for

event\_name = record['eventName']

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

obj\_key = record['s3']['object']['key']

# Verify that the object was written to S3

if event\_name == 'ObjectCreated:Put' and bucket\_name == 'sd-vehicle-data':

# If so, then add it to the writte\_objects list

written\_objects.append(obj\_key)

# Return the collected written objects

return {

"bucket": 'sd-vehicle-data',

"new\_objects": written\_objects

}