First of all, this [article](https://towardsdatascience.com/make-data-acquisition-easy-with-aws-lambda-python-in-12-steps-33fe201d1bb4) was very helpful. The parts that this article didn’t cover well are highlighted in yellow.

# A simple lambda example

Huayi Wei

4/24/2020

**My goal:** scrape data from google populartimes every 10 minutes into an S3 bucket from 6am to 9pm EST.

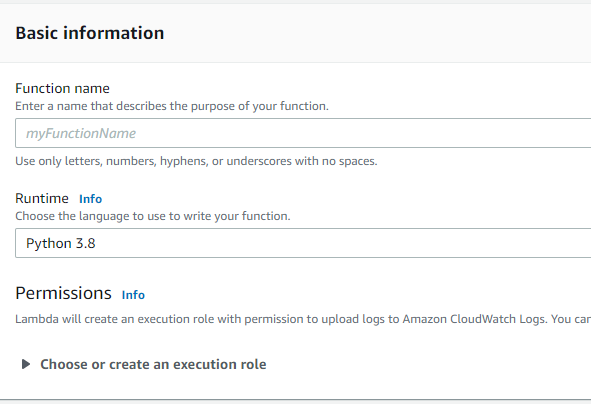
# Step 0: Sign into your aws account as a root user.

## Step1: Create an S3 bucket.

* I named mine “pop-time-data”
* Ignore all security setting for now

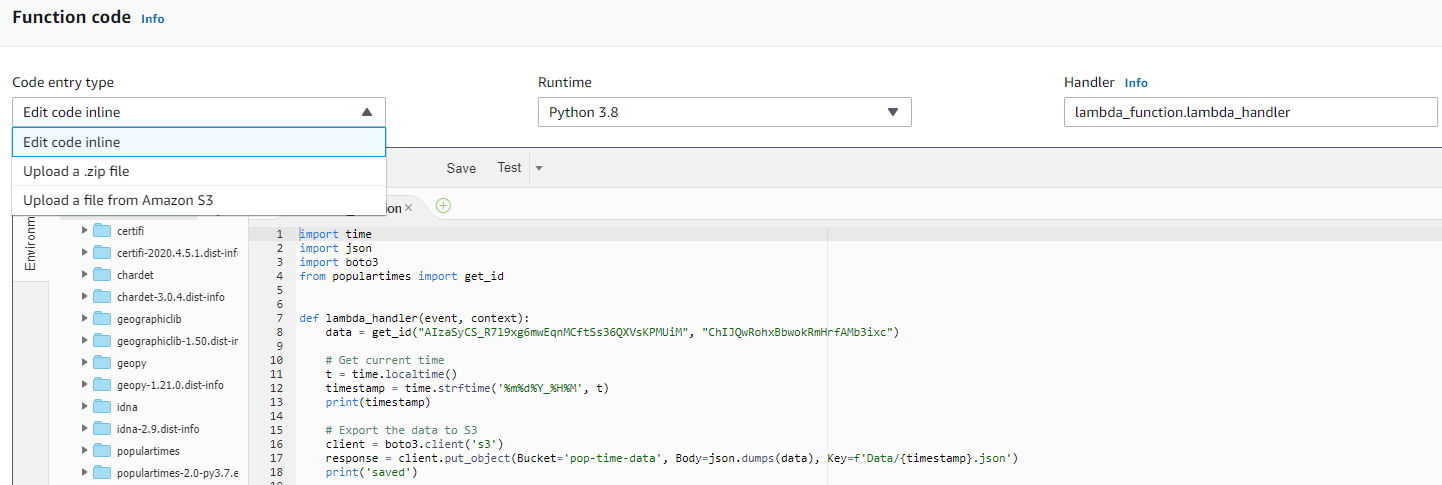
## Step2: Create a Lambda function.

Lambda-> create function



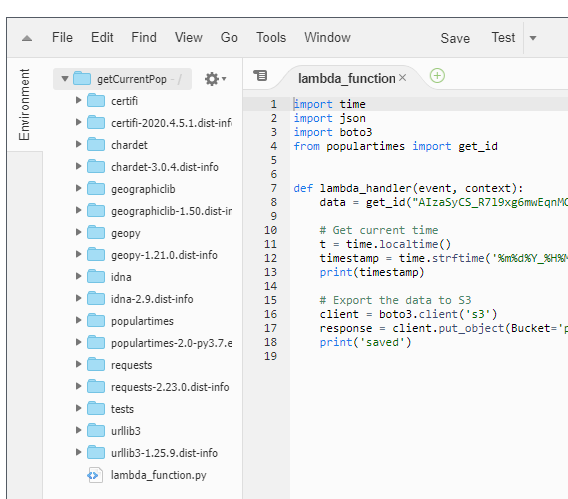
* I named mine “getCurrentPop”
* I choose Python 3.8 for my Runtime, as I’m using python 3
* Ignore Permissions for now

## Step3: Upload your Code

* Under “configuration”, go to “Function code”. 
* You can simply paste a code there if you don’t have dependencies. But if you do, then, you need to zip up all your dependencies into a .zip file.
* For sample code, refer to the file “lambda\_function.py”.
* In any case, a file named “lambda\_function.py” that contains a function called “lambda\_handler” needs to be there.
  + Why? On the right upper corner of the above screen shot, there’s a box called “Handler”. The function contained in this box is what lambda will be calling.
  + So if you file is called “othername.py” and you function is called “other\_handler”, then you need to change the entry into this box into “othername.other\_handler”.
* I used the .zip method because my code depends on the package “populartimes”.
* If you have a linux system, pip install that package into the folder that contains your “lambda\_function.py” file and then zip it up.

For example, cd to the folder, and do: “python3 -m pip install populartimes --target .”

* + This may also work on macs. On my ancient windows laptop this didn’t. So I end up using an EC2 instance just to make this zip file.
* Once you have the zip, choose “Upload a .zip file” and upload it to lambda.
* Now you’ll see a folder with the name of you zip that contains all your zipped files.
  + CRITICAL STEP: select all those files in that folder and drag them out so that they are under the root directory (which should be the name of your lambda). Delete the emptied folder.

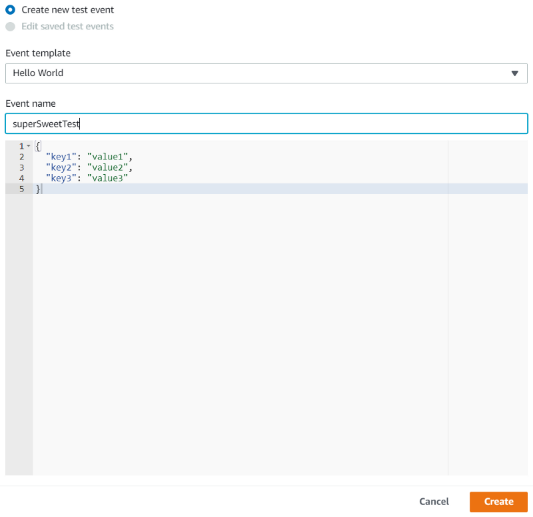


### Step4: Add permission

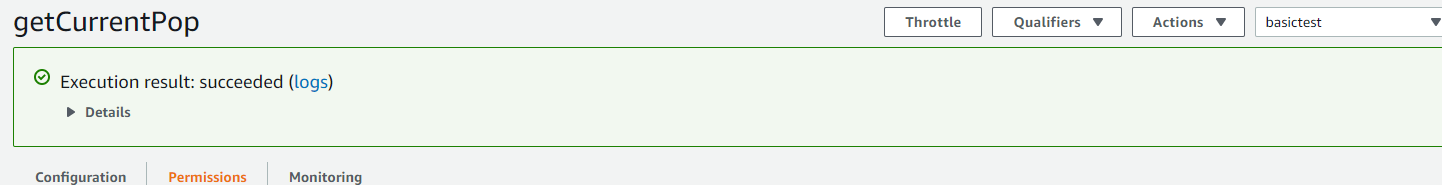
* Go to “Identity and Access Management (IAM)”, select “Roles”
* Click ‘Create Role’, select lambda on the next page, select ‘Next: permissions’, search for “AmazonS3FullAccess”, skip ‘Tags’, name your role and hit ‘Create role’.
* Go back to your lambda function. Go to “Permissions”, under “Execution role” click “edit” and select the role you just created.

### Step5: Create a test

* At the right upper corner of the page, click on “Test”, and create a new test. (I used the default).

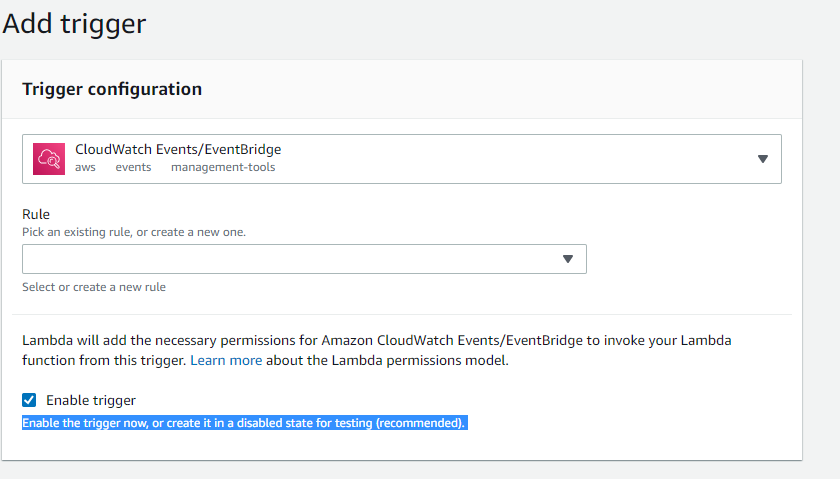


* Now, hit the test button. If it didn’t pass, then trouble shoot. Otherwise, you’ll see below pic and find a new file saved in your S3 bucket.



* + If you run into any permission issue, go back to the zip file on your computer. Unzip it. Delete the zip file. And zip the unzipped folder. Upload again.

## Step 6: Schedule it!

* Under “Configuration”, select “+Add trigger”, select CloudWatch
* 
* Under ‘Rule’, select ‘Create a New Rule’. Now, use cron to schedule yours. The AWS cron is based on UTC, which is 4hours faster than EST. Their cron also has slightly different fields. More details [here](https://docs.aws.amazon.com/lambda/latest/dg/services-cloudwatchevents-expressions.html).
  + For my purpose, I added two cron jobs, one for 6am-8pm (which is 10am to 0 in UTC), and the other one is 8pm-9pm (which is 0 to 1am in UTC).
    - 6am-8pm: cron(0/10 10-0 \* \* ? \*)
    - 8pm-9pm: cron(0/10 0-1 \* \* ? \*). *Note: it collects starting form 8pm but the last collection is at 8:50pm.*
* Once all the rules are added. You can sit back and wait for data to appear in your bucket!