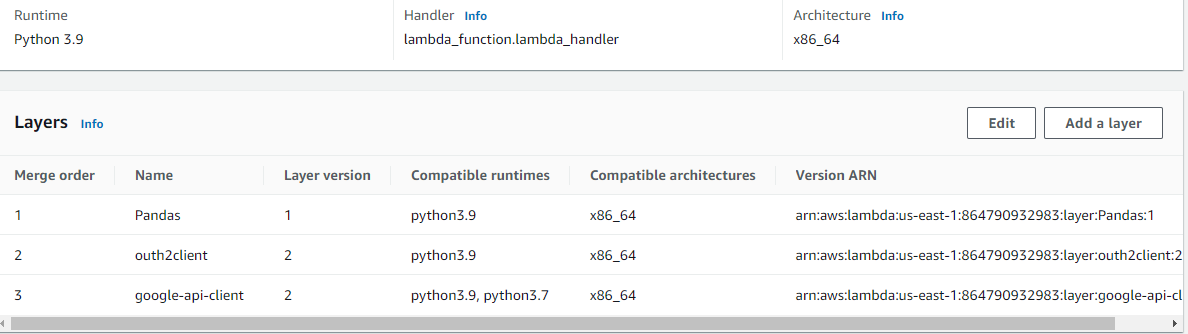
**STEPS TO CREATE LAMBDA FUNCTION FOR GOOGLE ANALYTICS**

**Prerequisites.**

# 1. Python version 3.7

1. **Lambda layers-** To add google-api-client, auth2client, and pandas to the lambda layer, we must have created a lambda layer.

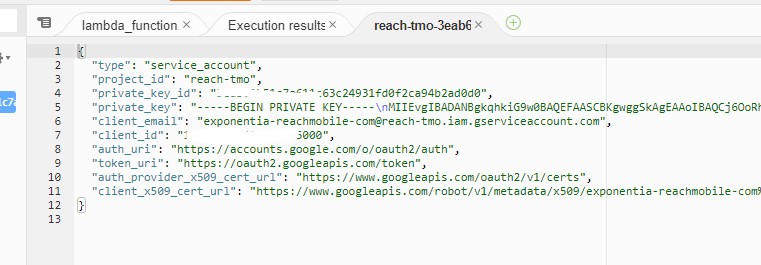


1. Lambda should have Amazons3FullAccess Access for reading and writing files to s3



1. There Should be a “service-account json” file in a root folder of the lambda function for each client, that has Read Only Permission On all tables.

**A file for a service account is shown below**



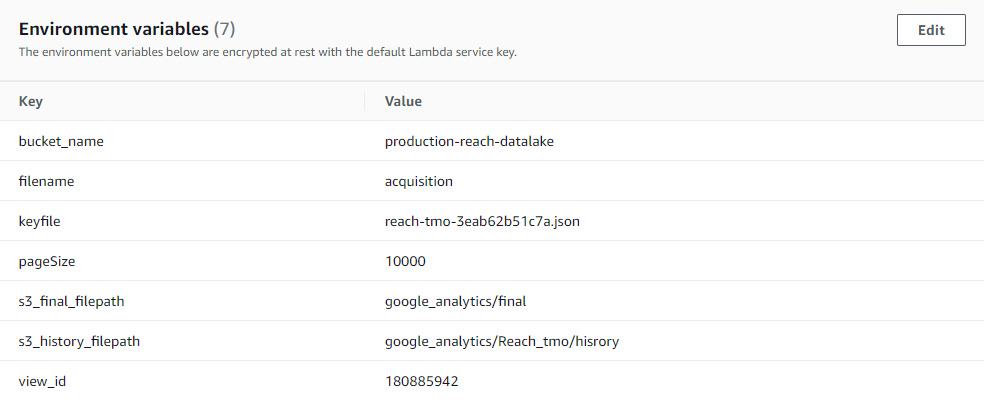
**Getting to know the code**

# 1. Environment variable

It is common source code for all clients, so all we need to change the environment variables according to each client.

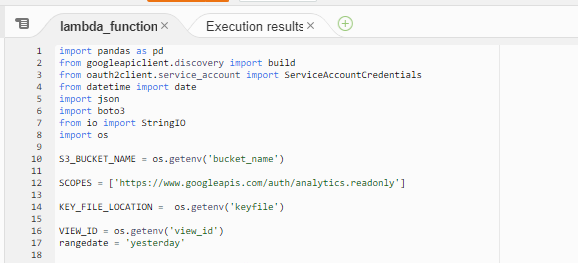
# Here is the list of environment variable

* **keyfile**- Name of the credential file.
* **bucket\_name-**  S3 bucket name where the final file will be stored.
* **filenam**e**-** Final file name for merging records.
* **s3\_final\_filepath** - Final file location on S3
* **s3\_history\_filepath** - History file location on S3 for ex google-analytics/<client name>/history
* **pageSize-** Size of the records fetched per request from the API
* **view\_id-** view id google analytics project.



## 2. Source code

* Imported all the required package and created rangedate for yesterday default value.



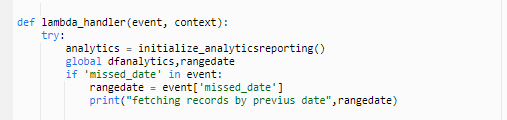
* Update the date object for missed date, if the event object contains the key (**missed\_date**) containing date in a format of (**yyyy-mm-dd**) to fetch records for given date. And initializing the analytics-reporting object using service account.json file

Passing missed date in event object

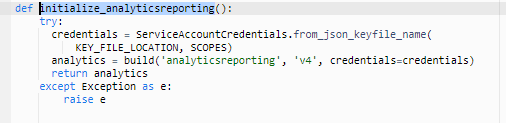


Updating the rangedate if missed date available in event,

And invoking the initialize\_analyticsreporting function.



Defining the initialize\_analyticsreporting function to create analytics report.



* Calling handle\_report the fetch the records from google analytics, this function will iterate for all the pages.

Calling handle\_report function to get data.

Defining the handle\_report function it will call the api for each page until pagetoken equals None



Generating the dynamic body against each call based on rangedate and pagetoken 

* We will upload the csv file to the s3 Inside history path that can be found in the environment variable and append the same records to the final file.

# Uploading file to history folder



##  Updating the final file

In the below function, we will check if the file already exists in S3, if it does, it will update the file, otherwise it will create a new one.

