SEMI-AUTOMATED DATA INGESTION PIPELINE INSTRUCTIONS

Note: These instructions:

1. *apply to both Private Lift and Private Attribution as they share the same Conversions API Gateway (CAPI-G) pipeline and requirements.*
2. are provided to support your efforts to set up your Private Lift Environment/PCS AWS Infrastructure as detailed in and to complete Step 3 in the PCS Partner Playbook (which describes the work needed to set up and run a private measurement study).

Prerequisites

1. CAPI-G version should be on or after 1.0.2
2. When **setup Private Lift Environment/PCS AWS Infrastructure Setup (Step 2)** in the doc PCS Partner Playbook, semi-auto data ingestion needs to be installed as outlined in the instructions detailed below.
3. The semi-auto pipeline needs at least 10 - 15 min to warm up after deployment.

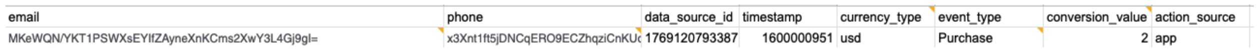
Glossary

- **S3 Data Bucket**: The bucket created or provided when you run the deploy command during the Private Lift Environment. It is the "**-d <S3 data output prefix>**" parameter.
- **S3 Config Bucket**: The bucket created or provided when you run the deploy command during the Private Lift Environment. It is the "**-s <S3 config bucket prefix>**" parameter.

Input - Prepare Your CSV File

Prepare a CSV file containing the following columns detailed in Table 1 below. Please make sure you follow the instructions in the **Description** column.

Example CSV (screenshot):



email	phone	data_source_id	timestamp	currency_type	event_type	conversion_value	action_source
MKeWQN/YKT1PSWXsEYIfZAyneXnKCms2XwY3L4Gj9gI=	x3Xnt1ft5jDNCqERO9ECZhqziCnKU	1769120793387	1600000951	usd	Purchase	2	app

Table 1: Semi-Auto Column Name and Description

Column Name	**Description**
email	• Match key to be used to create Private ID • Format: ○ trim any leading and trailing spaces

	○ in lower-case ○ **SHA-256 hashed** (please check "How to hash email" section below)
device_id	● Match key to be used to create Private ID ● Format: ○ lower case the IDFA/AAID ○ keep the hyphen ● Optional, but highly recommend to include this column if available
client_ip_address	● Match key to be used to create Private ID ● Format: ○ could be either IPv4 or IPv6 ● Optional, but highly recommend to include this column if available
data_source_id	● The Pixel or App id (Must match the data source id in study)
timestamp	● Event timestamp (in **unix time seconds**)
conversion_value	● Conversion value ● **Convert to USD** ● Can be **float** or **integer**
event_type	● Event types (e.g., Purchase, AddToCart, etc.) (Match event name in study)
action_source	● The value is "app", "website", or "others" depending on your data.

NOTE:

1. The file name **CAN NOT** have any spaces or special characters (i.e., any bracket, (,[,{.).
2. The column names should be in lowercase and need to exactly match the names listed above. The order doesn't matter.
3. If you only have one match key for all events, please don't include other columns into the CSV file. For example, if you will only use email, don't add device_id column into the CSV file.
4. If you would like to utilize multi-key matches in the future, please include all the PII data columns into the CSV file.
5. For Windows users, please open the CSV file in a text editor. We noticed there will be format issues (e.g. 1234567890 is automatically changed to 1.23E+10) if you open the CSV file in Excel.

Quick Format/Encryption Check

After you have prepared the CSV file in line with the **Input** section above, please perform the following quick sanity check to make sure your CSV file is ready:

1. Check the header names to make sure they **exactly match** the names (cases, spellings) listed in Table 1 above.
2. Before hashing the email, the raw email should have the leading and trailing spaces trimmed and be in lowercase.
3. Check if the email addresses are hashed.
    a. Check if the email column is SHA256 hashed. The value should look like "30a79640dfd8293d4f4965ec11821f640ca77979ca0a6b365f06372f81a3f602" instead of "123@gmail.com".
    b. Check if the SHA256 process is correct. Please do the same SHA256 process on this fake email address "123@gmail.com", and the expected value should be "30a79640dfd8293d4f4965ec11821f640ca77979ca0a6b365f06372f81a3f602".
4. Check if the timestamp value is in unix time, and in the expected range. Randomly choose one or two values from the timestamp column and convert them to the readable format and make sure they are in the expected time range.

Upload CSV File

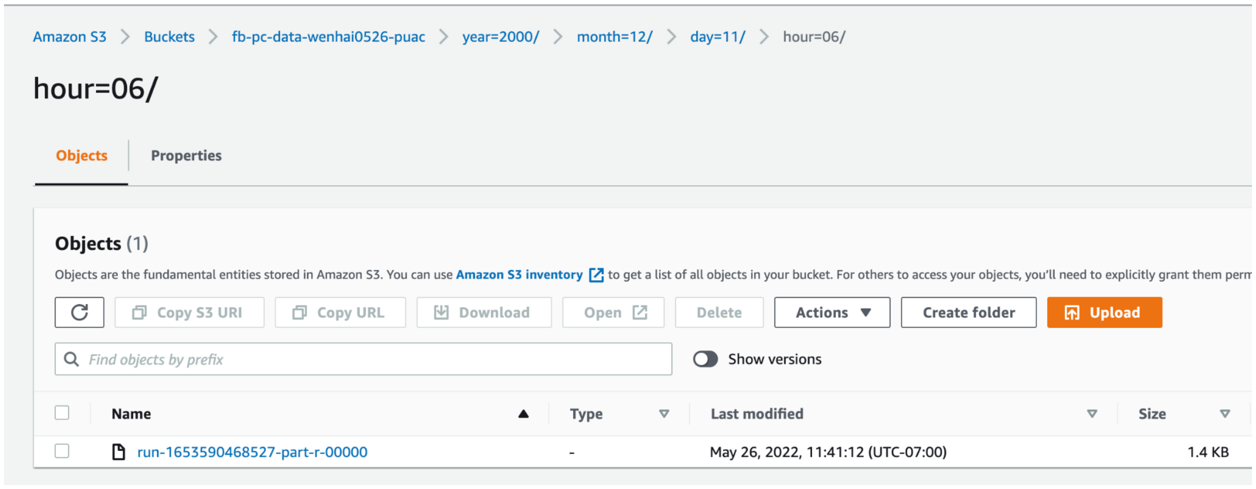
Upload the CSV file to the same **S3 Data Bucket**, under **semi-automated-data-ingestion** directory.

Wait until the data mentioned in the below **Output** section is showing in the S3 bucket.

Output

The re-processed and re-partitioned data will be ready in the same **S3 Data Bucket,** merged with the standard CAPI-G data pipeline storage (**not** under **semi-automated-data-ingestion**). Depending on the data size, it could take 5 - 30 mins to appear in S3.

Sample output (The S3 Data Bucket is "fb-pc-data-1101e2e" in this case.)


Amazon S3 > Buckets > fb-pc-data-wenhai0526-puac > year=2000/ > month=12/ > day=11/ > hour=06/

hour=06/

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them perm

Name	Type	Last modified	Size
run-1653590468527-part-r-00000	-	May 26, 2022, 11:41:12 (UTC-07:00)	1.4 KB

Generate Computation-Ready Input CSV File

After you get the output, there are two more steps to generate the final/Computation Ready input CSV file to run PL or PA.

1. Run AWS Glue Crawler. You have two options:
   a. Wait for at most one hour. The AWS Glue Crawler runs every hour. You could choose to wait for at most one hour.
   b. Trigger the AWS Glue Crawler manually. You can navigate to the AWS console -> AWS Glue -> Crawlers (on the left menu bar) , and run the crawler with the name **mpc-events-crawler-<Tag>**.
2. Follow the "**[Run every time you want to get results for a study]**" section in the PCS Partner Playbook to use AWS Athena to query the data and generate the CSV file.

How to Hash Email

Here is an example of how to perform the sha256 in python 3.x. Equivalent implementations in other languages are also good. Just make sure it works for the following example:

Example email input:

    example@fb.com

Example sha256 output:

    7a1d9f839aa2d4f3f348e8303bfcf699fd7c243baeb55238ee2d1bcd7b80f30e

Python 3.x:

```
        import hashlib
        sha256_output=hashlib.sha256(b'example@fb.com').hexdigest() or
        hashlib.sha256(bytes(str(example@fb.com),
encoding="utf-8")).hexdigest()
```

Presto-SQL:

```
SELECT
   LOWER(
      TO_HEX(
         SHA256(CAST(‘example@fb.com’ AS VARBINARY))
      )
   )
```