Tech Stack

Language Used: Python (2.7.13)

Python contains libraries and existing tools which developers feel they need to work from scratch. Python's strengths lie in working with indexed data structures and dictionaries, which are important in ETL operations.

ETL Tool Used: Apache Beam (SDK 2.20.0)

Apache Beam is an open source, unified model for defining both batch and streaming data-parallel processing pipelines. Using one of the open source Beam SDKs, you build a program that defines the pipeline. The pipeline is then executed by one of Beam's supported distributed processing back-ends, which include Apache Apex, Apache Flink, Apache Spark, and Google Cloud Dataflow.

Beam is particularly useful for Embarrassingly Parallel data processing tasks, in which the problem can be decomposed into many smaller bundles of data that can be processed independently and in parallel. Beam can be used for Extract, Transform, and Load (ETL) tasks and pure data integration. These tasks are useful for moving data between different storage media and data sources, transforming data into a more desirable format, or loading data onto a new system.

Solution Design

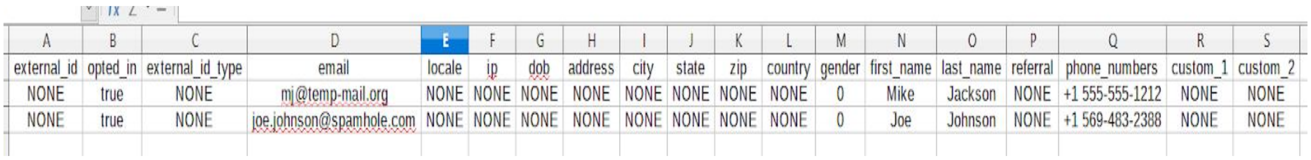
Pipeline Stages

1. Validates the Email ID from the customer table1 provided:
    a. Performs regex check on the mail id
    b. If mail id is a valid mail id according to regex then it is stored in separate location
    c. If mail id is in-valid mail id according to regex then it is stored in separate location
2. Convert data CSV data to JSON
    a. Convert data from csv file Customer1 (validated mail ID data) to json format and save it in a location
3. Convert data CSV data to JSON
    a. Convert data from csv file Customer 2 to json format and save it in a location
4. Join both the json data stored in the Step 2 based on the common key (id)
    a. Perform Left join on data from Customer1 & Customer2
    b. Save the joined data in json format

5. Save the data from json format to the csv format
    a. Save the joined data received from STEP 4 from json to csv format

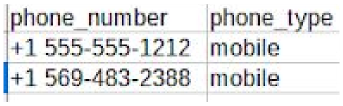
Screenshot of End result

Table 1



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
external_id	opted_in	external_id_type	email	locale	ip	dob	address	city	state	zip	country	gender	first_name	last_name	referral	phone_numbers	custom_1	custom_2
NONE	true	NONE	mj@temp-mail.org	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	0	Mike	Jackson	NONE	+1 555-555-1212	NONE	NONE
NONE	true	NONE	joe.johnson@spamhole.com	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	0	Joe	Johnson	NONE	+1 569-483-2388	NONE	NONE

Table 2



phone_number	phone_type
+1 555-555-1212	mobile
+1 569-483-2388	mobile

Directory structure

```
juhi_gupta@cloudshell:~/Session-M$ tree
├── input                                   -> Dir used for storing Input csv
│   ├── customer1.csv
│   └── customer2.csv
├── invalid                                  -> Dir used for storing data which fails validation check
│   └── customer1-00000-of-00001.csv
├── JoinedTable                             -> Dir used for storing joined table data
│   └── Data-00000-of-00001.json
├── output                                  -> Dir used for storing final output tables
│   ├── table1.csv
│   └── table2.csv
├── output-data1                            -> Clean & validated json data from Customer1 table
│   └── SendToAPI-00000-of-00001
├── output-data2                             -> Clean & validated json data from Customer2 table
│   └── SendToAPI-00000-of-00001
├── src                                     -> Dir used for storing source code
│   └── pipeline.py
└── valid                                   -> Dir used for storing validated customer1 table
   └── customer1-00000-of-00001.csv
```

Steps to setup & execute Pipeline:

1. Install all the dependency packages

   ```
   pip install -r requirements.txt
   ```

2. Go to the appropriate dir

   ```
   cd src
   ```

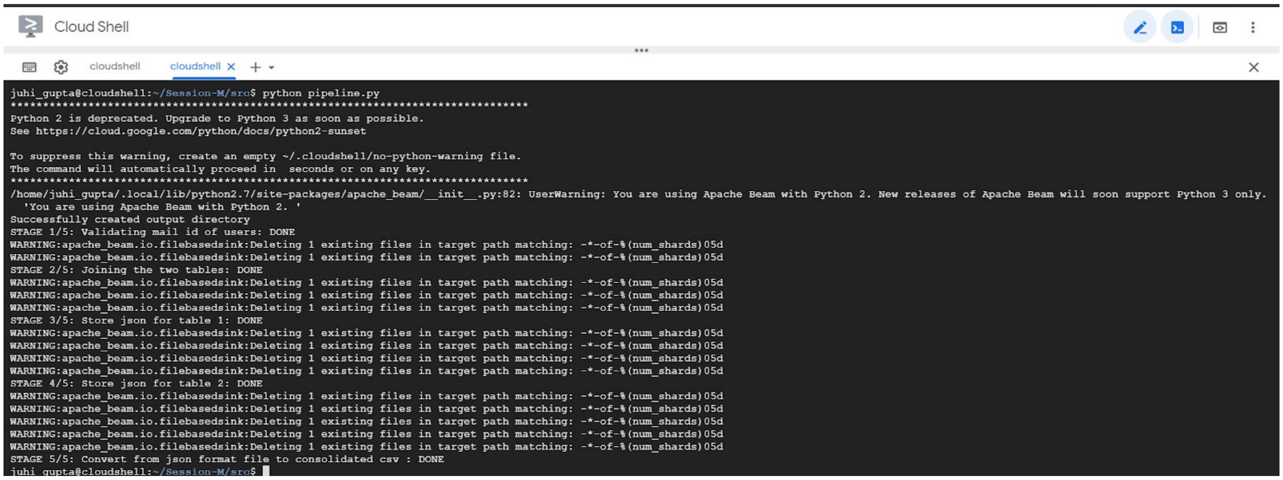
3. Delete old staging and output dirs (if any)

   ```
   rm -rf  "../output-data1" "../output-data2" "../valid" "../JoinedTable" "../invalid" "../output"
   ```

4. Run command to start the execution of pipeline

   ```
   python pipeline.py
   ```

   Screenshot:



```
juhi_gupta@cloudshell:~/Session-M/src$ python pipeline.py
********************************************************************************
Python 2 is deprecated. Upgrade to Python 3 as soon as possible.
See https://cloud.google.com/python/docs/python2-sunset

To suppress this warning, create an empty ~/.cloudshell/no-python-warning file.
The command will automatically proceed in  seconds or on any key.
********************************************************************************
/home/juhi_gupta/.local/lib/python2.7/site-packages/apache_beam/__init__.py:82: UserWarning: You are using Apache Beam with Python 2. New releases of Apache Beam will soon support Python 3 only.
  'You are using Apache Beam with Python 2. '
Successfully created output directory
STAGE 1/5: Validating mail id of users: DONE
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
STAGE 2/5: Joining the two tables: DONE
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
STAGE 3/5: Store json for table 1: DONE
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
STAGE 4/5: Store json for table 2: DONE
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
WARNING:apache_beam.io.filebasedsink:Deleting 1 existing files in target path matching: -*-of-%(num_shards)05d
STAGE 5/5: Convert from json format file to consolidated csv : DONE
juhi_gupta@cloudshell:~/Session-M/src$ 
```

5. Check the output dir for the final output
   a. Check output dir for the two csv files for the required two tables