Assumptions:

1. My system will deal with data for only **one** client. The same system can be replicated for all clients.
2. In RTB, "USER INFO" refers to the **client** who is bidding for the ad space, "AUCTION DETAILS" would have USER_ID, SPACE_ID, AMOUNT, ad targeting criteria would have value for 3 metrics (age, location, gender)
3. AdvertiseX is **Demand Service Platform**, hence it does not bother will supply data.

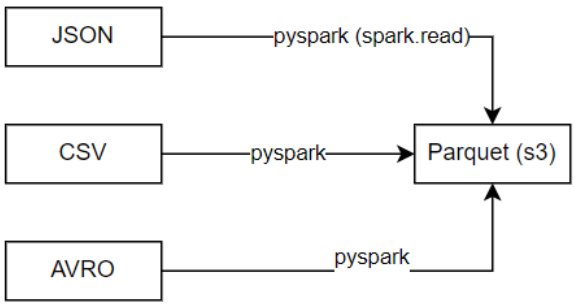
Infrastructure:

1. AWS **S3** buckets to store **Parquet** files.
2. AWS **EC2** (**Ubuntu**) for software installations (**Airflow, Python, Spark, Kafka consumer, Presto**)

Data Ingestion:

Use **Python** scripts in Apache **Airflow** to schedule and orchestrate the batch data ingestion process.

We can use **pyspark** to read JSON, CSV and AVRO and write them to **parquet** files into our **s3** bucket



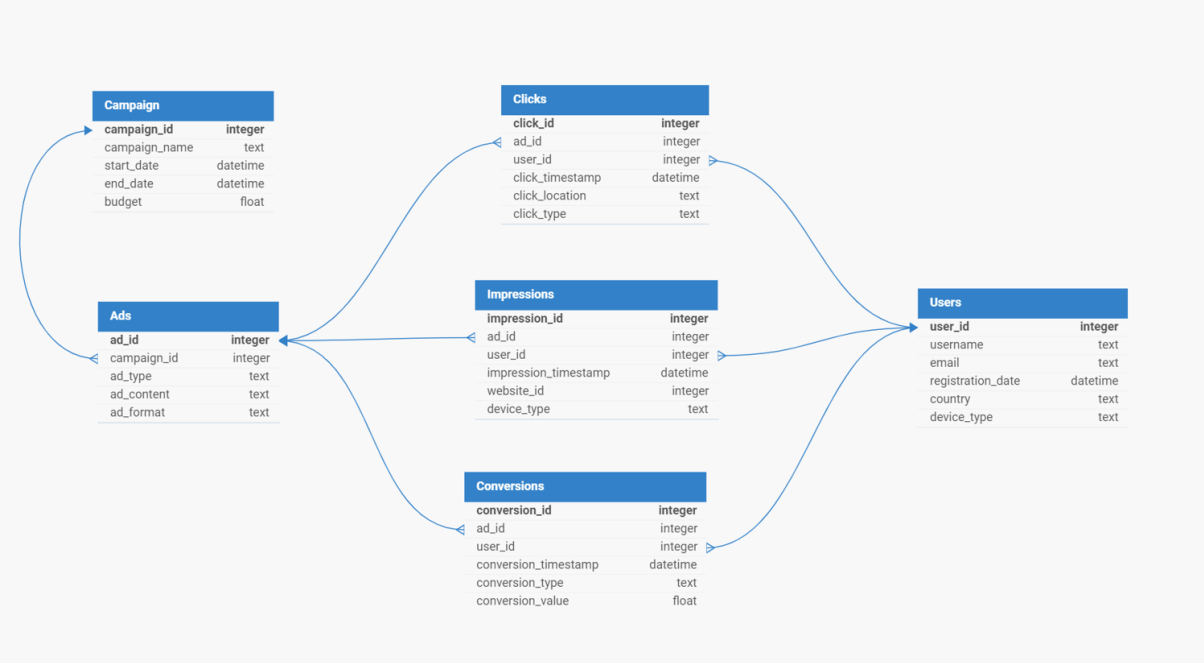
We can have a **Kafka** cluster set up and spark can read from respective Kafka stream for **real time** data ingestion.

Data Processing:

Once data is ingested into Parquet files, we can use **SQL** via **presto** to query to files. We can do data validation, filtering and deduplication on top of the base tables to create **golden standard tables**. Once the gold standard tables are created, we can have **recon queries** on top of them to ensure **data integrity.**

Data Storage and Query Performance:

To enable optimal storage and quick return of data for analytics and decision-making, we have to come up with an optimal data model which can cater to multiple analytics use cases. One such data model can be -



Error Handling and Monitoring:

We can use **Airflow** to **mail** us for any failure in our data ingestion and ETL pipelines. Airflow can also be used to send **mails based on results from recon queries**. We can create dataframes on top of any recon results which do not match base data. Whenever length of dataframe is greater than 0, the mail will be sent out.