**Intercom Event Processing Framework**

**Purpose:**

The purpose of this project is to build a framework to process the events logs from the S3 buckets. Some events can be broken or incomplete. Parse all the correct events and process it toload into Redshift.

**Infrastructure:**

I have used Databricks Community Cloud for this project work which provide the following:

* 1 Node development cluster
* Apache Spark 2.4.5
* Scala 2.11
* Databricks File System (DBFS) [This can be replaced with S3 buckets]

<https://community.cloud.databricks.com/>

**Design Consideration:**

* I have replaced the S3 bucket with other storage system due to the unavailability of AWS account. The same code will work out of S3 bucket as well with extra authentication to access the event logs.
* I have used SPARK for processing due to the fact that it provides Distributed In-Memory Computing and supports APIs for Scala, Python and SQL.
* Spark DataFrames provides extra functionalities like:
  + Table structure to parse the JSON
  + In built "\_corrupt\_record" column which can store all the unparsed records from JSON
  + SQL can be written over DataFrame to quickly analyse the records
  + Partitioning and other spark configurations to optimize and scale up the solution
* Final output is stored in DataBricks File System for testing. We can upload it into the Redshift table or save it in S3 bucket.

**Solution Notebook:** The solution notebook can be accessed using the below URL. This notebook can be imported to any spark cluster and execute.

[**https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/2269326428961072/3499935794714464/7487126396143688/latest.html**](https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/2269326428961072/3499935794714464/7487126396143688/latest.html)

**Process/Code Flow:**

S3 Bucket

Apache Spark

JSON Load

Spark DataFrame

Filter

Corrupt

Record

NO

Parse strings using SQL

Spark DataFrame

YES

Flatten Spark DataFrame

Flatten Spark DataFrame

Add missing columns

(This will make the columns same in both DF)

UNION

Final Event Spark DataFrame