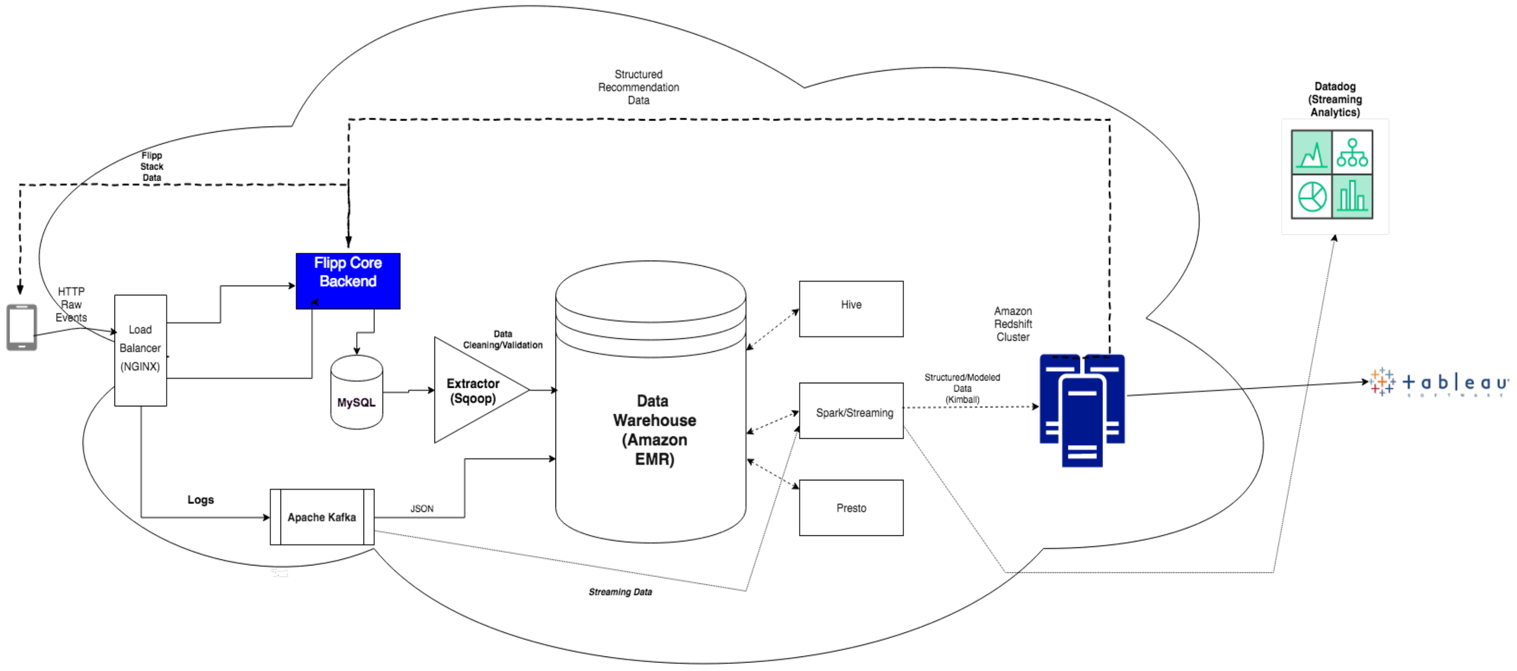
Part 2-System Architecture



Main Component details (*Diagram above available separately in .png format*):

A. **Load Balancer**: I will use NGINX as a load balancer in the AWS cloud to handle the large number of requests from millions of mobile devices. These devices will transmit various different events data such as clickstream, views, clips etc. These events will appear in the form of sys_logs describing every action of the user in raw text format.
B. **Flipp Core**: The LB will forward these requests to the Flipp core backend application cluster and the Kafka cluster. The Flipp core will have its own dedicated DBMS such as MySQL to store user_data,flyer_data,vendor_data etc. The requests forwarded to kafka in the form of messages thus acting as a producer. The flipp core will be based on multiple nodes and each node can be using a WSGI(Web service Gateway Interface) such as *Flask* to define its endpoints to handle incoming requests.
C. **Extractor**: The extractor component will be an important tool in extracting raw user data from the MySQL database. Schemas for the different types of tables/datasets to be exported to the data warehouse will be created and validated here. Furthermore data cleaning/ schema mapping will also be done here. Tools such as *Apache Sqoop* can be housed in this layer to transfer

bulk data from MySQL to HDFS. Alternatively, once data validation and cleaning phase is done, we can write Hadoop MapReduce jobs to output data in the form of *.parquet* or *.avro* files to HDFS depending on the requirement. The various MapReduce jobs can be scheduled by a batch workflow job scheduler such as *Azkaban (*Developed at LinkedIn).

D. **Kafka**: The kafka cluster will have two types of *consumer groups.* One would be the Amazon EMR data warehouse which would consume data from various different topics in the form of JSON. These JSON objects would be populated in Hive in raw/unstructured format. Therefore every topic in kafka would have a raw table with the same name in Hive as well. The second consumer group would be the Spark streaming jobs which would listen to individual topics and post data to a real-time analytics platform such as *DataDog.*

E. **Hive**: This datastore,which will be housed in the Amazon EMR, will contain two types of data: unmodelled/dirty json data from Kafka in the form of tables; the second would be clean data incoming in the form of parquet files from the extractor module present in HDFS. Both the .parquet files and Hive tables would be queryable by PRESTO. The JSON data in the Hive Tables would require further SQL processing to transform them into tables and partitions etc. Therefore one could use the PRESTO wrapper to wrangle kafka and MySQL backup data in the data warehouse together.

F. **Spark**: The spark layer will comprise of two modules: Regular and streaming contexts. The regular Spark Context would take as input data from the .parquet files (which is clean data) and apply low or high level transformations in order to model the data into *dimensions* and *facts* (Kimball's Modelling). Once the facts and dimensions are resolved the output modeled data can be stored separately in an Amazon redshift cluster. This cluster can be used by data analysts(Tableau), scientists to derive insights from the data which can either be used by Flipp backend core to provide content based on recommendation of data to the user's Flipp stack or to predict the overall performance of the app and study user behavior. The streaming context can apply transformations to individual data packets and post these pakcets for real time analytics to *Datadog* or *Kibana.*

G. **Amazon Redshift**: This cluster is where all the clean modelled/relational data will be housed. Data scientists and analysts can use this datastore to monitor performance and run ML algortihms to make predictions. Redshift is ideal because it provides better query performance and parallelization than Hive.