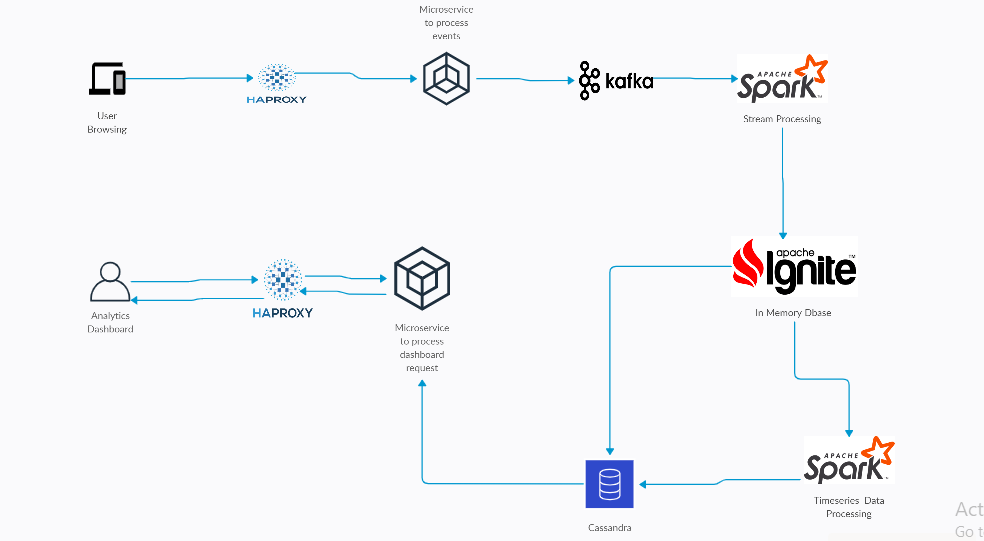
**Design Question**

Design A Google Analytic like Backend System. We need to provide Google Analytic like services to our customers. Please provide a high level solution design for the backend system. Feel free to choose any open source tools as you want.

**Requirements**

1. Handle large write volume: Billions of write events per day.
2. Handle large read/query volume: Millions of merchants wish to gain insight into their business. Read/Query patterns are time-series related metrics.
3. Provide metrics to customers with at most one hour delay.
4. Run with minimum downtime.
5. Have the ability to reprocess historical data in case of bugs in the processing logic.

**Design Solution**



**Design Components**

**HA Proxy**

HA Proxy, which stands for High Availability Proxy, is a popular open source software TCP/HTTP Load Balancer and proxying solution. Its most common use is to improve the performance and reliability of a server environment by distributing the workload across multiple servers. It is used in many high-profile environments, including: GitHub, Imgur, Instagram, and Twitter.

A backend can contain one or many servers in it — generally speaking, adding more servers to your backend will increase your potential load capacity by spreading the load over multiple servers. Increased reliability is also achieved through this manner, in case some of your backend servers become unavailable.

**Apache Kafka**

Apache Kafka is used for building real-time streaming data pipelines that reliably get data between many independent systems or applications.

It allows:

* Publishing and subscribing to streams of records
* Storing streams of records in a fault-tolerant, durable way.
* Kafka stores messages for a specified time and In case of error in processing logic these messages can be replayed by changing topic Index.

Kafka Streams being scalable, highly available and fault-tolerant, and providing the streams functionality (transformations / stateful transformations) are what we need — not to mention Kafka being a reliable and mature messaging system.

Kafka is run as a cluster on one or more servers that can span multiple datacenters spread across geographies. Those servers are usually called brokers.  
  
Kafka Streams is a pretty fast, lightweight stream processing solution that works best if all of the data ingestion is coming through Apache Kafka. The ingested data is read directly from Kafka by Apache Spark for stream processing

# **Apache Spark**

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams.

It provides a high-level abstraction called a discretized stream, or DStream, which represents a continuous stream of data.

DStreams can be created either from input data streams from sources such as Kafka, Flume, and Kinesis, or by applying high-level operations on other DStreams. Internally, a DStream is represented as a sequence of RDDs

# **Cassandra**

# Apache Cassandra is a highly scalable and available distributed database that facilitates and allows storing and managing high velocity structured data across multiple commodity servers without a single point of failure. Cassandra is very good at processing time series data.

The Apache Cassandra is an extremely powerful open source distributed database system that works extremely well to handle huge volumes of records spread across multiple commodity servers. It can be easily scaled to meet sudden increase in demand, by deploying multi-node Cassandra clusters, meets high availability requirements, and there is no single point of failure. Apache Cassandra has best write and read performance.