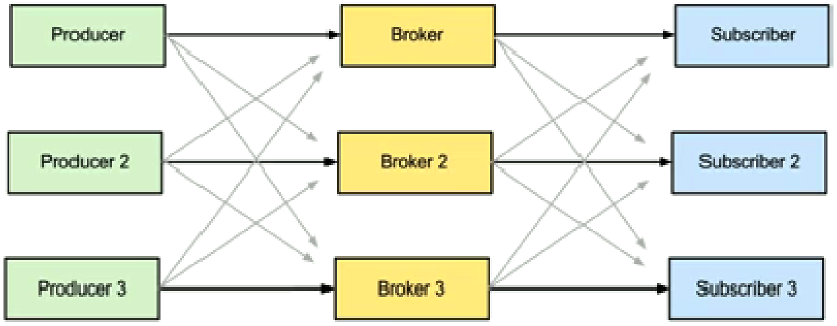
Below comes the description of the each component, its duties and the interaction with each other.

Flume agent on servers: The solution starts from collecting the logs messages from servers. To collect log messages, the flume agent will be installed on those servers. Flume agent is a lightweight java program which runs on JVM. This agent is configured to read the mentioned log file and push the log messages (`event` as per Flume's language) into the Data collector i.e. Kafka broker on configured batch size or time interval.

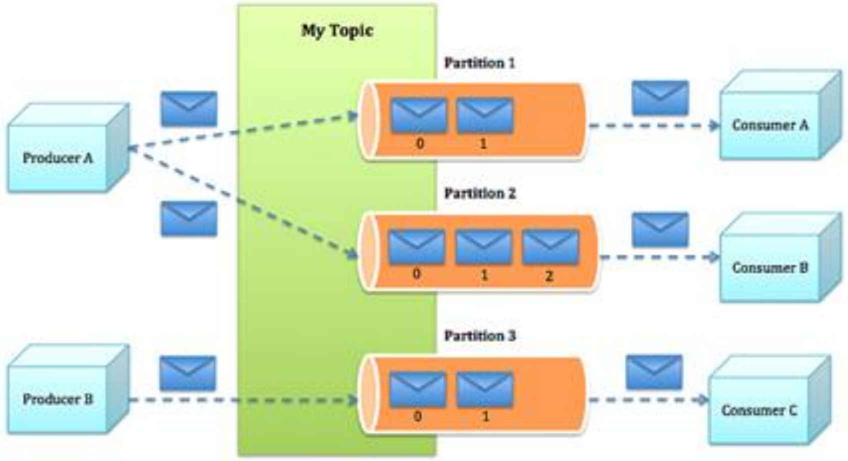
Data Collector/Kafka Broker: This module collects all the log messages pushed by the flume agents and keeps it safe for consumers to consume. In this design, Apache Kafka is chosen as the data collector because of its scalability, durability and low-latency advantages.

Cluster and Load Balancing of Kafka broker: As per the messaging concepts, to collect the log messages, a topic T will be created in the Kafka broker and all the flume agents will logically push the log messages to that single topic. For high availability of the broker and for durability of the messages, we choose to have multi node-multi broker setup

Load balancing on producer side: To gain the optimal load balancing and true parallel processing ability of the multi broker setup, the topic T will be partitioned into n partitions using some key; where n >= number of subscribed consumers and each broker will contain one or more partitions. The flume producers are configured with a partition key so that all the messages from the particular producer will logically be collected by single broker. The key can be subnet value, or a geo location values etc.



Multi Broker Setup with multiple producers and consumers



Topic and Partition Concept - shows parallel producing and consuming ability

Flume Consumer: This layer consists of group of flume consumer agents connected to Kafka broker and subscribed to topic T. These agents together, will read the data from the subscribed topic and syncs into an intermediate storage area which will be of HDFS files.

Load balancing on consumer side*:* To achieve the parallel reading ability, all these agents will be grouped into one consumer group, so that these agents will decide among themselves who will read from which partition and will do load-balancing by themselves. On other side these agents will be configured to sync the consumed log messages into HDFS files.