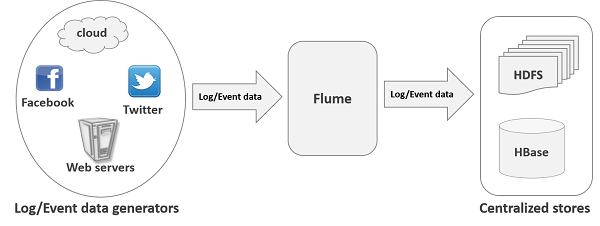
**Explain the need of Flume.**

* Using Apache Flume we can store the data in to any of the centralized stores (HBase, HDFS).
* When the rate of incoming data exceeds the rate at which data can be written to the destination, Flume acts as a mediator between data producers and the centralized stores and provides a steady flow of data between them.
* Flume provides the feature of **contextual routing**.
* The transactions in Flume are channel-based where two transactions (one sender and one receiver) are maintained for each message. It guarantees reliable message delivery.
* Flume is reliable, fault tolerant, scalable, manageable, and customizable.

**Explain the working of Flume and its components in brief.**

Apache Flume is a tool/service/data ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log files, events (etc...) from various sources to a centralized data store.

Flume is a highly reliable, distributed, and configurable tool. It is principally designed to copy streaming data (log data) from various web servers to HDFS.



Core Concepts

* Event: A byte payload with optional string headers that represent the unit of data that Flume can transport from it’s point of origination to it’s final destination.
* Flow: Movement of events from the point of origin to their final destination is considered a data flow, or simply flow. This is not a rigorous definition and is used only at a high level for description purposes.
* Client: An interface implementation that operates at the point of origin of events and delivers them to a Flume agent. Clients typically operate in the process space of the application they are consuming data from. For example, Flume Log4j Appender is a client.
* Agent: An independent process that hosts flume components such as sources, channels and sinks, and thus has the ability to receive, store and forward events to their next-hop destination.
* Source: An interface implementation that can consume events delivered to it via a specific mechanism. For example, an Avro source is a source implementation that can be used to receive Avro events from clients or other agents in the flow. When a source receives an event, it hands it over to one or more channels.
* Channel: A transient store for events, where events are delivered to the channel via sources operating within the agent. An event put in a channel stays in that channel until a sink removes it for further transport. An example of channel is the JDBC channel that uses a file-system backed embedded database to persist the events until they are removed by a sink. Channels play an important role in ensuring durability of the flows.
* Sink: An interface implementation that can remove events from a channel and transmit them to the next agent in the flow, or to the event’s final destination. Sinks that transmit the event to it’s final destination are also known as terminal sinks. The Flume HDFS sink is an example of a terminal sink. Whereas the Flume Avro sink is an example of a regular sink that can transmit messages to other agents that are running an Avro source.