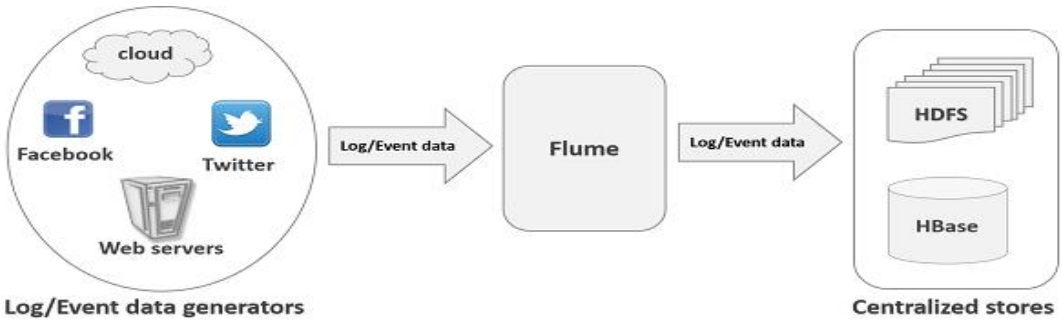
# APACHE FLUME

## Apache Flume - Introduction

### What is Flume?

- 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.




## Applications of Flume

- Assume an e-commerce web application wants to analyze the customer behavior from a particular region.
- To do so, they would need to move the available log data in to Hadoop for analysis.
- Here, Apache Flume comes to our rescue.
- Flume is used to move the log data generated by application servers into HDFS at a higher speed.

## Advantages of Flume

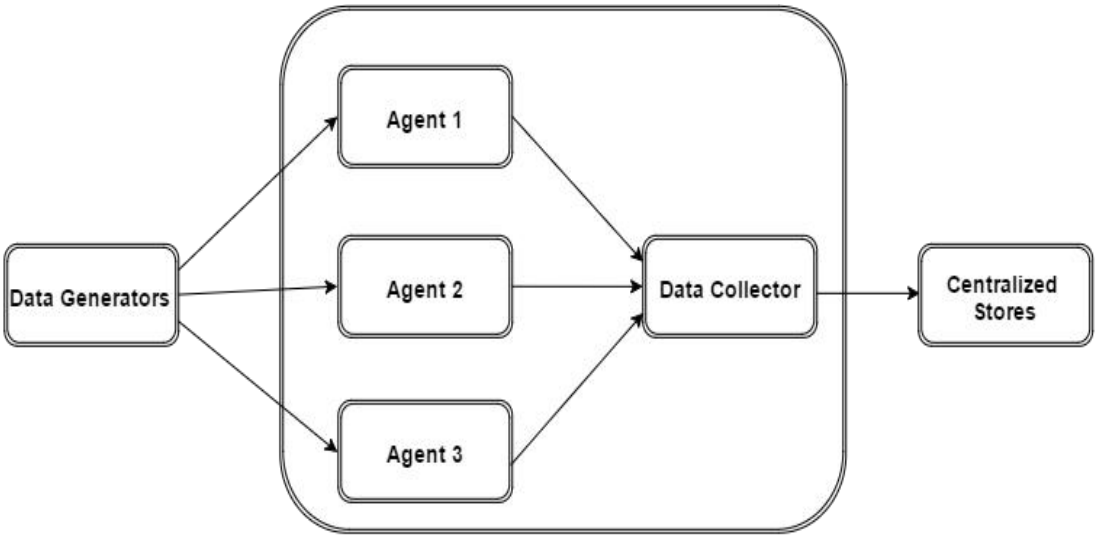
- Using Apache Flume we can store the data in to any of the centralized stores (HBase, HDFS).
- When the rate of incoming data exceeds the rate at which data can be written to the destination, Flume acts as a mediator between data producers and the centralized stores and provides a steady flow of data 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.

## Features of Flume

- Using Flume, we can get the data from multiple servers immediately into Hadoop.
- Along with the log files, Flume is also used to import huge volumes of event data produced by social networking sites like Facebook and Twitter, and e-commerce websites like Amazon and Flipkart.
- Flume supports a large set of sources and destinations types.
- Flume supports multi-hop flows, fan-in fan-out flows, contextual routing, etc.
- Flume can be scaled horizontally.

## Apache Flume - Architecture

- The following illustration depicts the basic architecture of Flume.
- As shown in the illustration,
  - data generators (such as Facebook, Twitter) generate data which gets collected by individual Flume agents running on them.
  - Thereafter, a data collector (which is also an agent) collects the data from the agents which is aggregated and pushed into a centralized store such as HDFS or HBase.




## Flume Event

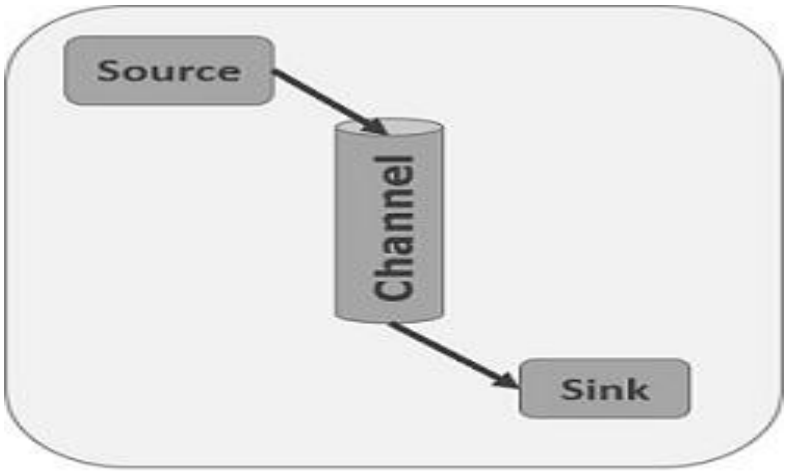
- An event is the basic unit of the data transported inside Flume.
- It contains a payload of byte array that is to be transported from the source to the destination accompanied by optional headers.
- A typical Flume event would have the following structure −




Flume event

**Flume Agent**

- An agent is an independent daemon process (JVM) in Flume.
- It receives the data (events) from clients or other agents and forwards it to its next destination (sink or agent). Flume may have more than one agent. Following diagram represents a Flume Agent




Flume Agent

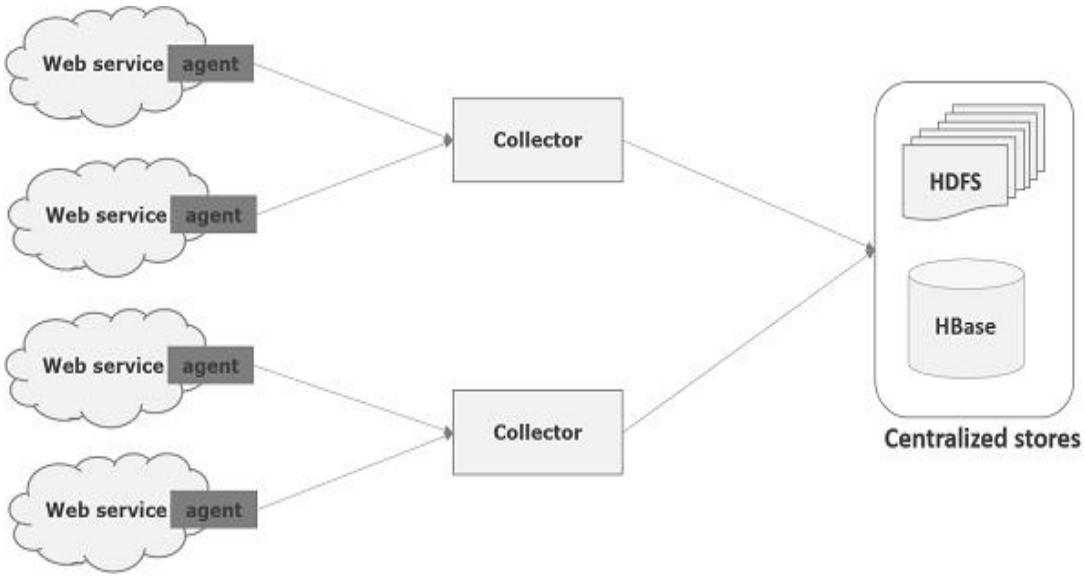
- As shown in the diagram a Flume Agent contains three main components namely, source, channel, and sink.
- **Source**
- A source is the component of an Agent which receives data from the data generators and transfers it to one or more channels in the form of Flume events.
- Apache Flume supports several types of sources and each source receives events from a specified data generator.
- Example − Avro source, Thrift source, twitter 1% source etc.
- **Channel**
- A channel is a transient store which receives the events from the source and buffers them till they are consumed by sinks. It acts as a bridge between the sources and the sinks.

- These channels are fully transactional and they can work with any number of sources and sinks.
- Example − JDBC channel, File system channel, Memory channel, etc.
- **Sink**
- A sink stores the data into centralized stores like HBase and HDFS. It consumes the data (events) from the channels and delivers it to the destination. The destination of the sink might be another agent or the central stores.
- Example − HDFS sink

- **More components** that play a vital role in transferring the events from the data generator to the centralized stores.
- **Interceptors**
- Interceptors are used to alter/inspect flume events which are transferred between source and channel.
- **Channel Selectors**
- These are used to determine which channel is to be opted to transfer the data in case of multiple channels. There are two types of channel selectors
  - Default channel selectors − These are also known as replicating channel selectors they replicates all the events in each channel.
  - Multiplexing channel selectors − These decides the channel to send an event based on the address in the header of that event.
- **Sink Processors**
- These are used to invoke a particular sink from the selected group of sinks. These are used to create failover paths for your sinks or load balance events across multiple sinks from a channel.

## Apache Flume - Data Flow

- Flume is a framework which is used to move log data into HDFS. Generally events and log data are generated by the log servers and these servers have Flume agents running on them. These agents receive the data from the data generators.
- The data in these agents will be collected by an intermediate node known as Collector. Just like agents, there can be multiple collectors in Flume.
- Finally, the data from all these collectors will be aggregated and pushed to a centralized store such as HBase or HDFS. The following diagram explains the data flow in Flume.




- **Multi-hop Flow**
- Within Flume, there can be multiple agents and before reaching the final destination, an event may travel through more than one agent. This is known as multi-hop flow.
- **Fan-out Flow**
- The dataflow from one source to multiple channels is known as fan-out flow. It is of two types −
  - Replicating − The data flow where the data will be replicated in all the configured channels.
  - Multiplexing − The data flow where the data will be sent to a selected channel which is mentioned in the header of the event.
- **Fan-in Flow**
- The data flow in which the data will be transferred from many sources to one channel is known as fan-in flow.
- **Failure Handling**
- In Flume, for each event, two transactions take place: one at the sender and one at the receiver. The sender sends events to the receiver. Soon after receiving the data, the receiver commits its own transaction and sends a received signal to the sender. After receiving the signal, the sender commits its transaction. (Sender will not commit its transaction till it receives a signal from the receiver.)