**Q1Explain the need of Flume?**

**A1**

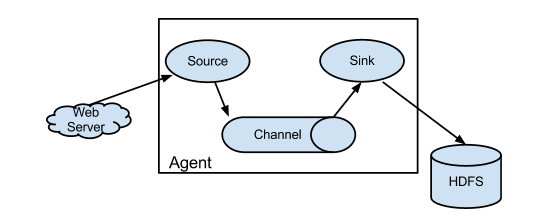
* Often it is assumed that the data is already in HDFS, or can be copied there in bulk. However, there are many systems that don’t meet this assumption.
* They produce streams of data that we would like to aggregate, store, and analyze using Hadoop — and these are the systems that Apache Flume is an ideal fit for.
* Flume is designed for high-volume ingestion into Hadoop of event-based data.
* The canonical example is using Flume to collect logfiles from a bank of web servers, then moving the log events from those files into new aggregated files in HDFS for processing.
* The usual destination (or sink in Flume parlance) is HDFS. However, Flume is flexible enough to write to other systems, like HBase or Solr.

**Q2 Explain the working of Flume and its components in brief?**

**A2**

**Working of Flume**

Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application.

****

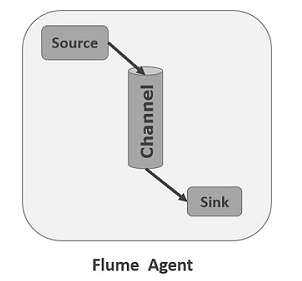
**Flume Event**

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.



## 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.**



### **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.