● Explain the need of Flume.

• 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.

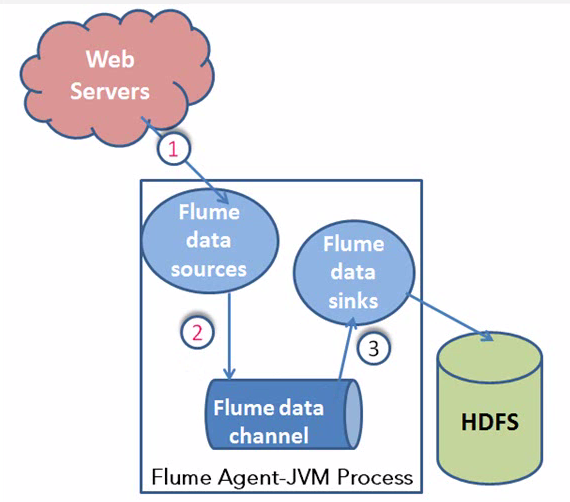
• The usual destination (or sink in Flume parlance) is HDFS. However, Flume is flexible enough to write to other systems, like HBase or Solr.

EG WE WANT TO ANALYSE LOG FILES OR TWEEWS (REGION OR TIME FROM WHERE IT IS COMING)

FLUME IS MAINLY USED FOR PULLING LOG OR STREAMING DATA OR EVENT DATA AND PUTTING INTO HDFS FOR ANALYSIS

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

COMPONENTS



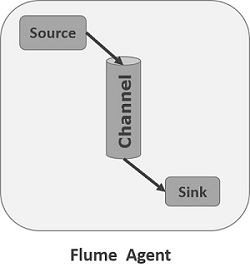
1.FLUME AGENT

### 2. Interceptors

### 3.Sink Processors

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**



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

**Note** − A flume agent can have multiple sources, sinks and channels. We have listed all the supported sources, sinks, channels in the Flume configuration chapter of this tutorial.

### 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.