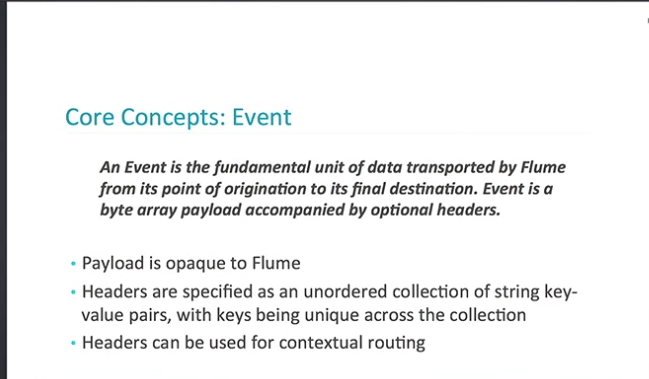
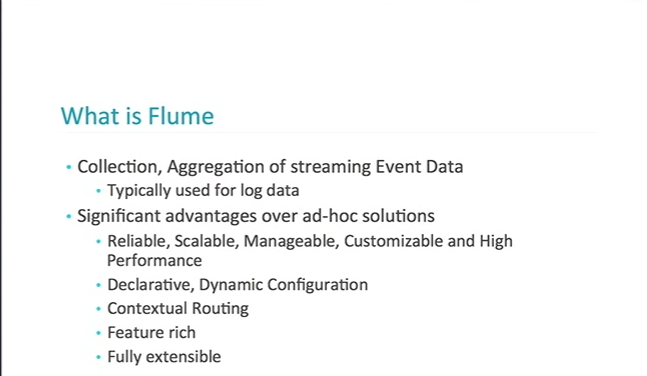
Q1) Explain the need of Flume.

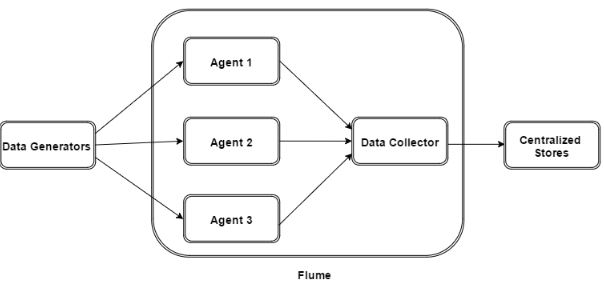
* A company has tons of services running on multiple servers. And lots of data (logs) are produced by them, now we need to analyze them altogether. In order process that logs, we need a reliable, scalable, extensible and manageable distributed data collection service which can perform flow of unstructured data (logs) from one location to another where they will be processed (say in [HDFS](http://goo.gl/26B0d3)). [Apache flume](http://goo.gl/eZZbmE) is an open source data collection service for moving the data from source to destination.
* [Apache Flume](http://data-flair.training/blogs/flume-quickstart-guide-learn-install-flume-copy-data-into-hdfs/) is the most reliable, distributed, and available service for systematically collecting, aggregating, and moving large amounts of streaming data (logs) into the Hadoop Distributed File System ([HDFS](http://goo.gl/AacJiN)). Based on streaming data flows, it has a simple and flexible architecture. It is highly fault tolerant and robust and with tunable reliability mechanisms for fail-over and recovery. Flume allows data collection in batch as well as streaming mode.
* 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.

****

Q2) Explain the working of Flume and its components in brief.

# **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.



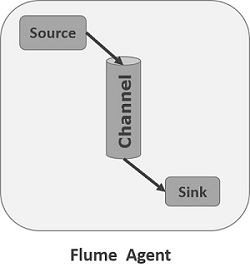
## Flume Event

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 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.

## Additional Components of Flume Agent

What we have discussed above are the primitive components of the agent. In addition to this, we have a few 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.