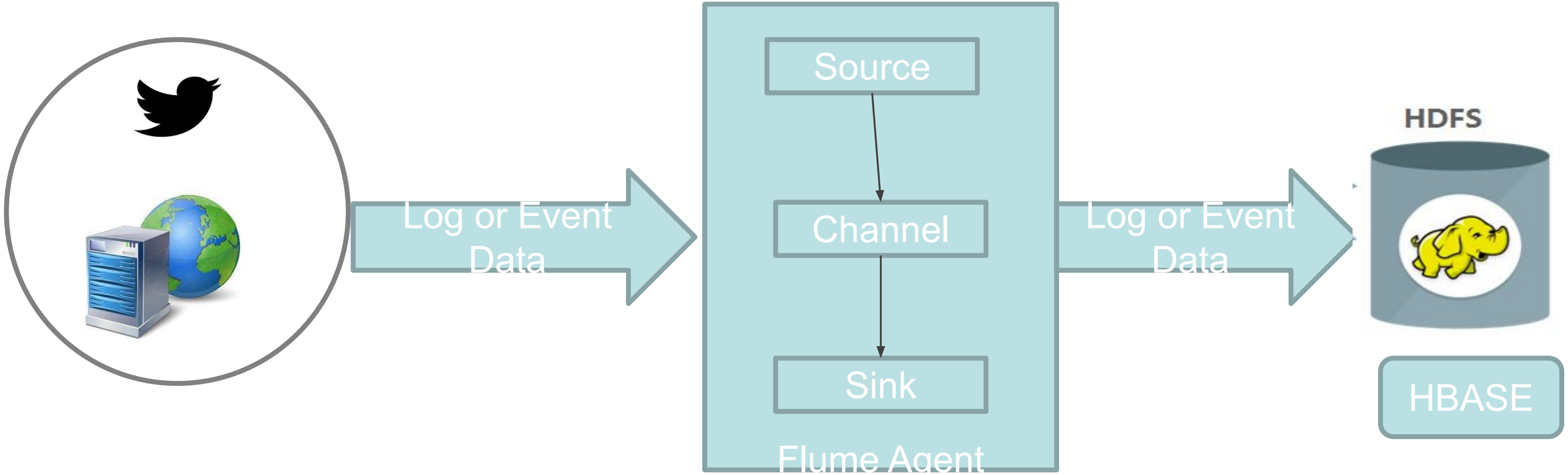
Flume Activity

Reference : Apache Flume

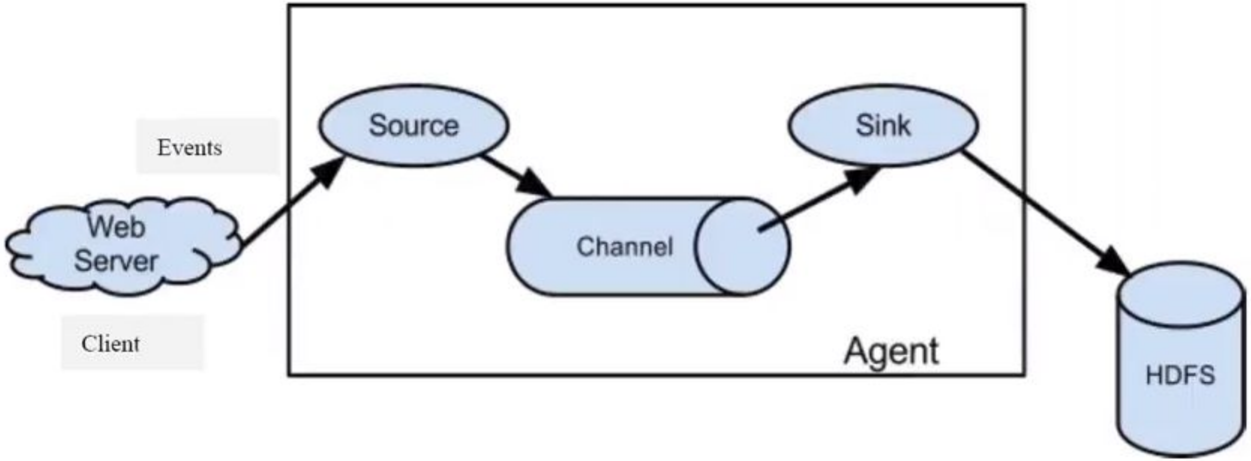
Flume

- Apache Flume is a 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.

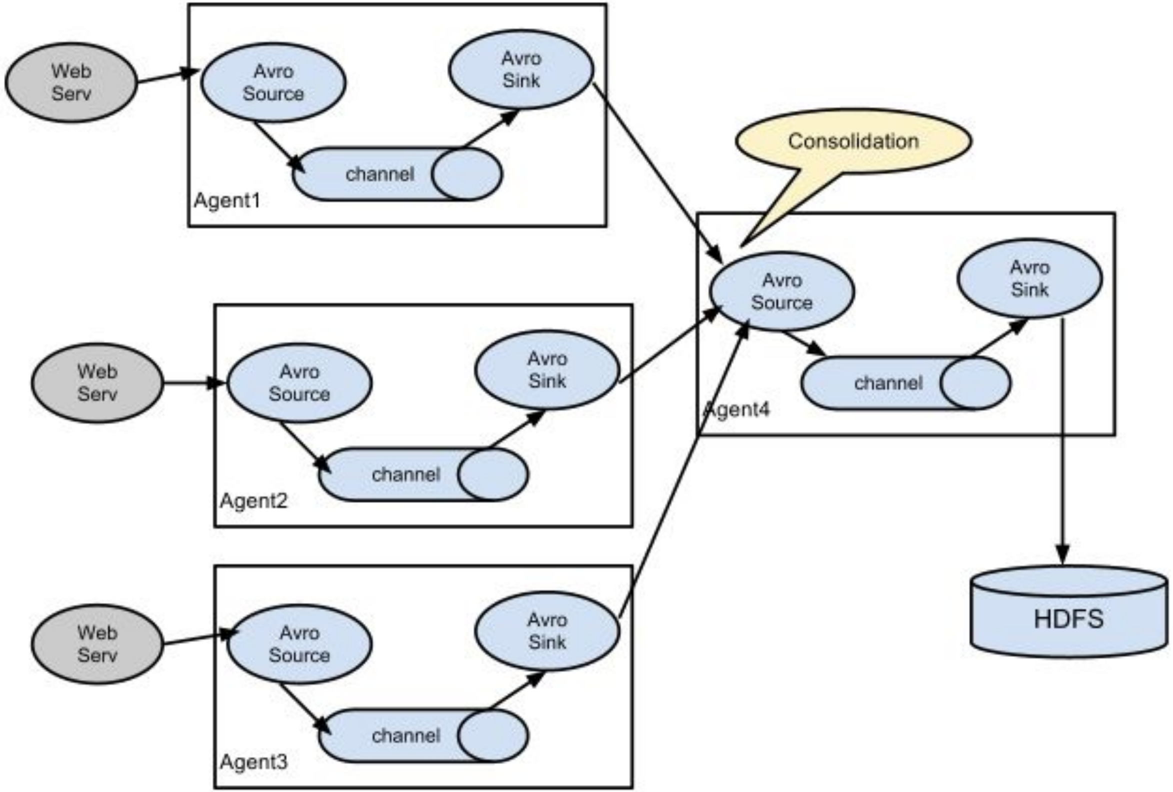
Components


Log or Event Data
Source
Channel
Sink
Flume Agent
Log or Event Data
HDFS
HBASE

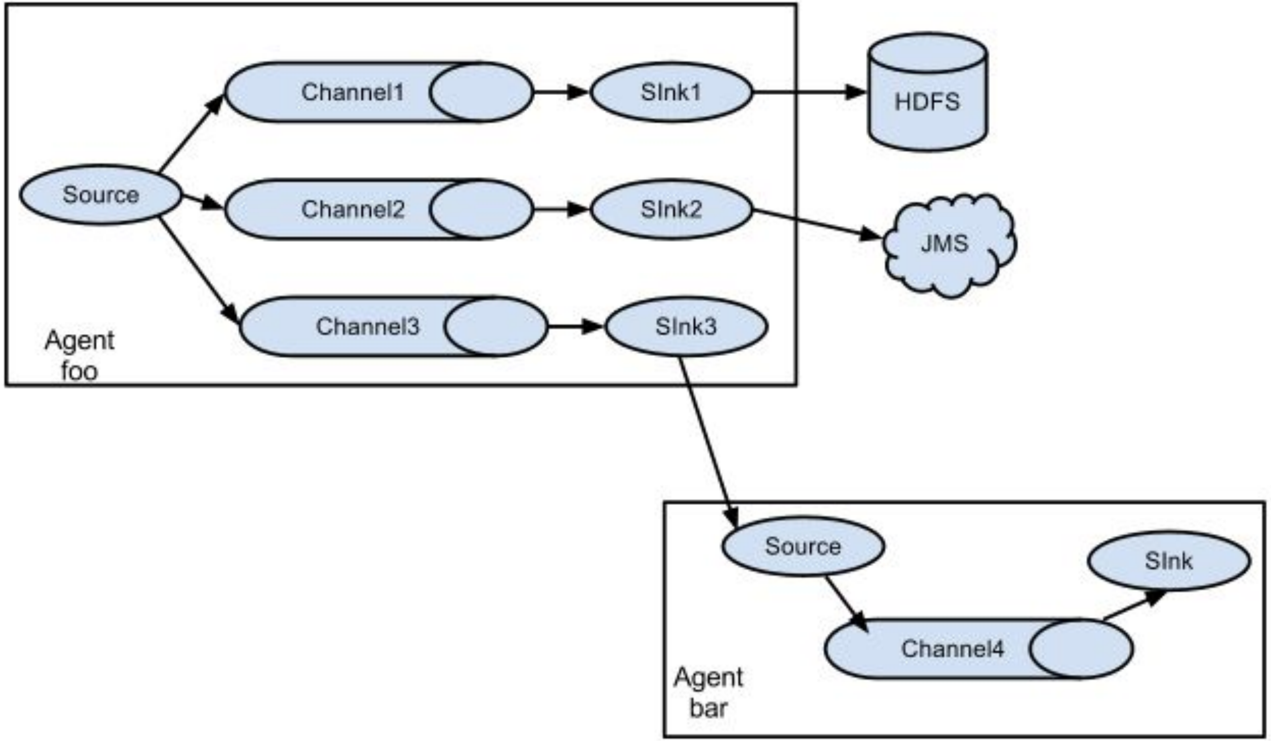
Broader explanation



Consolidation

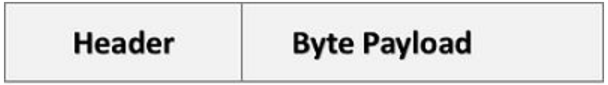


Multiplexing



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 −



Header	Byte Payload

Flume event

Flume contd..

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.
- Agent contains three main components namely, **source**, **channel**, and **sink.**

Contd..

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 source etc.

Contd..

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

Contd..

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

Flume Conf. Properties

In the Flume configuration file, we need to –

1. Name the components of the flume agent.
2. Describe/Configure the Source.
3. Describe/Configure the Channel.
4. Describe/Configure the Sink.
5. Bind the Source and the Sink to the Channel.

Step 1 :

Naming the Components of the Agent.

- Give a name to the flume Agent and let it be like

  Ex: flumeAgent,appagent,uberagent etc.

- Give the for the components

  <Agent_Name>.sources = <Source_Name>

  <Agent_Name>.channels = <Channel_Name>

  <Agent_Name>.sinks = <Sink_Name>

Step2:

Describe the Source:

Each Source will have different list of Properties that is needed to be defined here .

<Agent_name>.sources.<Source_Name>.type = <value>

<Agent_name>.sources.<Source_Name>.property = <value>

<Agent_name>.sources.<Source_Name>.property 1= <value>

Note : Refer Flume Docs for Properties of Source

Step3:

Describing the Channel

<Agent_Name>.channels.<Channel_Name>.type = <Value>

<Agent_Name>.channels.<Channel_Name>.property = <Value>

<Agent_Name>.channels.<Channel_Name>.property = <Value>

Note : Refer Flume Docs for Channel Properties

Step4:

Describe the Sink:

- Each sink will have a separate list of properties. The property named “type” is common to every Sink.
- Along with the property “type”, it is needed to provide values to all the **required** properties
- <Agent_Name>.sinks.<Sink_Name>.type = <Value>
- <Agent_Name>.sinks.<Sink_Name>.property = <Value>

Note: Refer Flume Docs for the Sink Properties to be defined

Step 5:

Binding the Sources and Sinks to Channels

<Agent_Name>.sources.<Source_Name>.channels = <Channel_Name>

<Agent_Name>.sinks.<Sink_Name>.channels = <Channel_Name>

Contd..

- Call the Following on the bash Shell

```
flume-ng agent --conf ./conf/ -f
conf/flume-conf.properties
-Dflume.root.logger=DEBUG,console
-n flumeAgent
```

Let’s begin !

This is what you need to do.

1. Log on to the Cluster.
2. Create below directories on Linux
   - Uber
3. Copy the files as below
   - Uber/Hist
   - Uber/Streaming

Flume to Ingest the Data

- Create the Flume Configuration Properties file in the linux .
- Create a folder “conf “ and inside the conf place the flumeconf properties file.

Get the flume running !

```
flume-ng agent --conf /conf/ -f
conf/flumeconf.properties
-Dflume.root.logger=DEBUG,console -n
flumeAgent
```