edureka!

Big Data & Hadoop

Course Topics

- ✓ Module 1
  - ✓ Understanding Big Data
  - ✓ Hadoop Architecture
- ✓ Module 2
  - ✓ Hadoop Cluster Configuration
  - ✓ Data loading Techniques
  - ✓ Hadoop Project Environment
- ✓ Module 3
  - ✓ Hadoop MapReduce framework
  - ✓ Programming in Map Reduce
- ✓ Module 4
  - ✓ Advance MapReduce
  - ✓ MRUnit testing framework
- ✓ Module 5
  - ✓ Analytics using Pig
  - ✓ Understanding Pig Latin
- ✓ Module 6
  - ✓ Analytics using Hive
  - ✓ Understanding HIVE QL
- ✓ Module 7
  - ✓ Advance Hive
  - ✓ NoSQL Databases and HBASE
- ✓ Module 8
  - ✓ Advance HBASE
  - ✓ Zookeeper Service
- ✓ Module 9
  - ✓ **Hadoop 2.0 – New Features**
  - ✓ **Programming in MRv2**
- ✓ Module 10
  - ✓ Apache Oozie
  - ✓ Real world Datasets and Analysis
  - ✓ Project Discussion

Flume

Flume is described as a system for the retrieval and distribution of logs, meaning line-oriented textual data.

It is not a generic data-distribution platform; in particular, don't look to use it for the retrieval or movement of binary data.

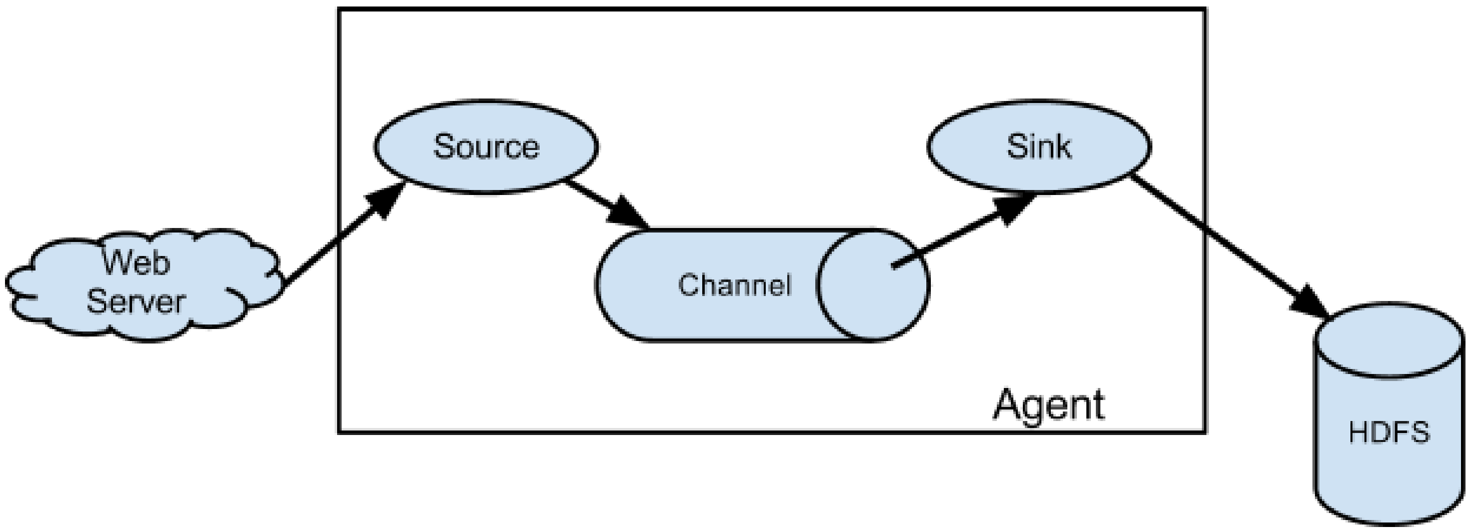
Majority of the data processed in Hadoop matches this description.

Flume can be used to transport massive quantities of event data including but not limited to network traffic data, social-media-generated data, email messages and pretty much any data source possible.

Flume

A Flume event is defined as a unit of data flow having a byte payload and an optional set of string attributes.

A Flume agent is a (JVM) process that hosts the components through which events flow from an external source to the next destination.



Flume

The agent needs to know what individual components to load and how they are connected in order to constitute the flow.

This is done by listing the names of each of the sources, sinks and channels in the agent, and then specifying the connecting channel for each sink and source.

Defining the flow

You need to list the sources, sinks and channels for the given agent, and then point the source and sink to a channel.

```
# list the sources, sinks and channels for the agent

<Agent>.sources = <Source>

<Agent>.sinks = <Sink>

<Agent>.channels = <Channel1> <Channel2>

# set channel for source

<Agent>.sources.<Source>.channels = <Channel1> <Channel2> ...

# set channel for sink

<Agent>.sinks.<Sink>.channel = <Channel1>
```

Network traffic onto HDFS

Defining the flow

For example agent named myagent is reading data from an Netcat and sending it to HDFS via a memory channel. The config file tohdfs.config could look like:

```
myagent.sources = netsource
myagent.sinks = hdfssink
myagent.channels = memorychannel

myagent.sources.netsource.channels = memorychannel
myagent.sinks.hdfssink.channel = memorychannel
```

Configuring individual components

```
myagent.sources.netsource.type = netcat
myagent.sources.netsource.bind = localhost
myagent.sources.netsource.port = 3000

myagent.sinks.hdfssink.type = hdfs
myagent.sinks.hdfssink.hdfs.path = /flume
myagent.sinks.hdfssink.hdfs.filePrefix = log
myagent.sinks.hdfssink.hdfs.rollCount = 3

myagent.channels.memorychannel.type = memory
myagent.channels.memorychannel.capacity = 1000
myagent.channels.memorychannel.transactionCapacity = 100
```

Channel Capacity This is the maximum capacity number of events of the channel.

Channel Transaction Capacity. This is the max number of events stored in the channel per transaction.

Configuring individual components

```
$ bin/flume-ng agent --conf conf --conf-file toHDFS.conf --name myagent
```

The agent argument tells Flume to start an agent, which is the generic name for a running Flume process involved in data movement.

The conf directory path

The particular configuration file for the process we are going to launch

The name of the agent within the configuration file

END