Apache Flume :

The classic case of Big Data Analytic :

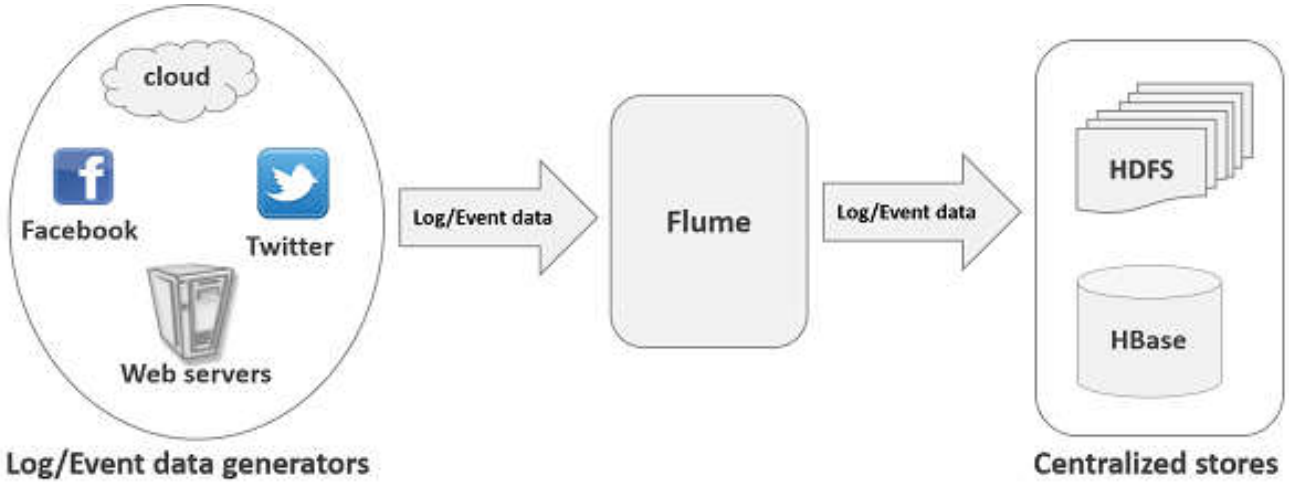
Streaming / Log Data

Generally, most of the data that is to be analyzed will be produced by various data sources like applications servers, social networking sites, cloud servers, and enterprise servers. This data will be in the form of **log files** and **events**.

What Apache Flume :

Apache Flume is a tool/service/data ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log data, events (etc...) from various webserves to a centralized data store.

It is a highly reliable, distributed, and configurable tool that is principally designed to transfer streaming data from various sources to HDFS.

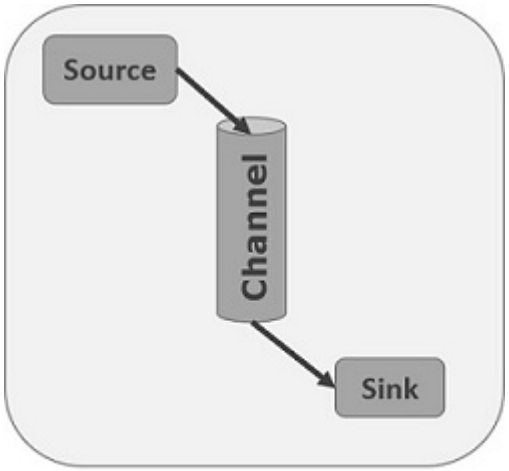


Apache Flume Architecture :

- Flume Event

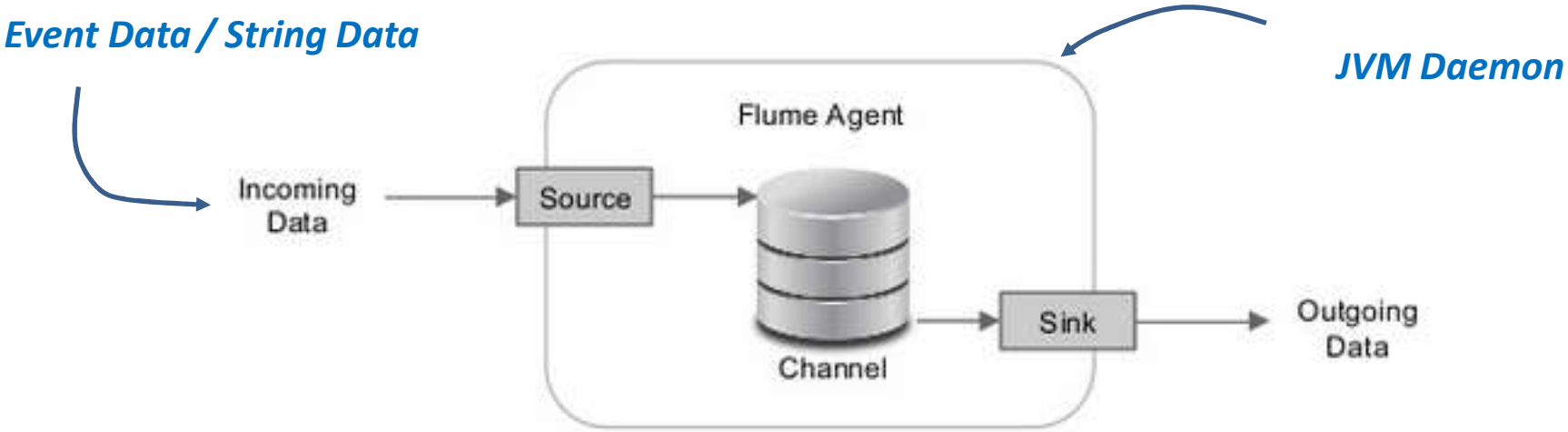
Flume event

- Flume Agent



Flume Agent

Apache Flume Architecture :



Source
- Accepts incoming Data
- Scales as required
- Writes data to Channel

Channel
- Stores data in the order received

Sink
- Removes data from Channel
- Sends data to downstream Agent or Destination

Interceptors

Channel Selectors
- **Default**
- **Multiplexing**

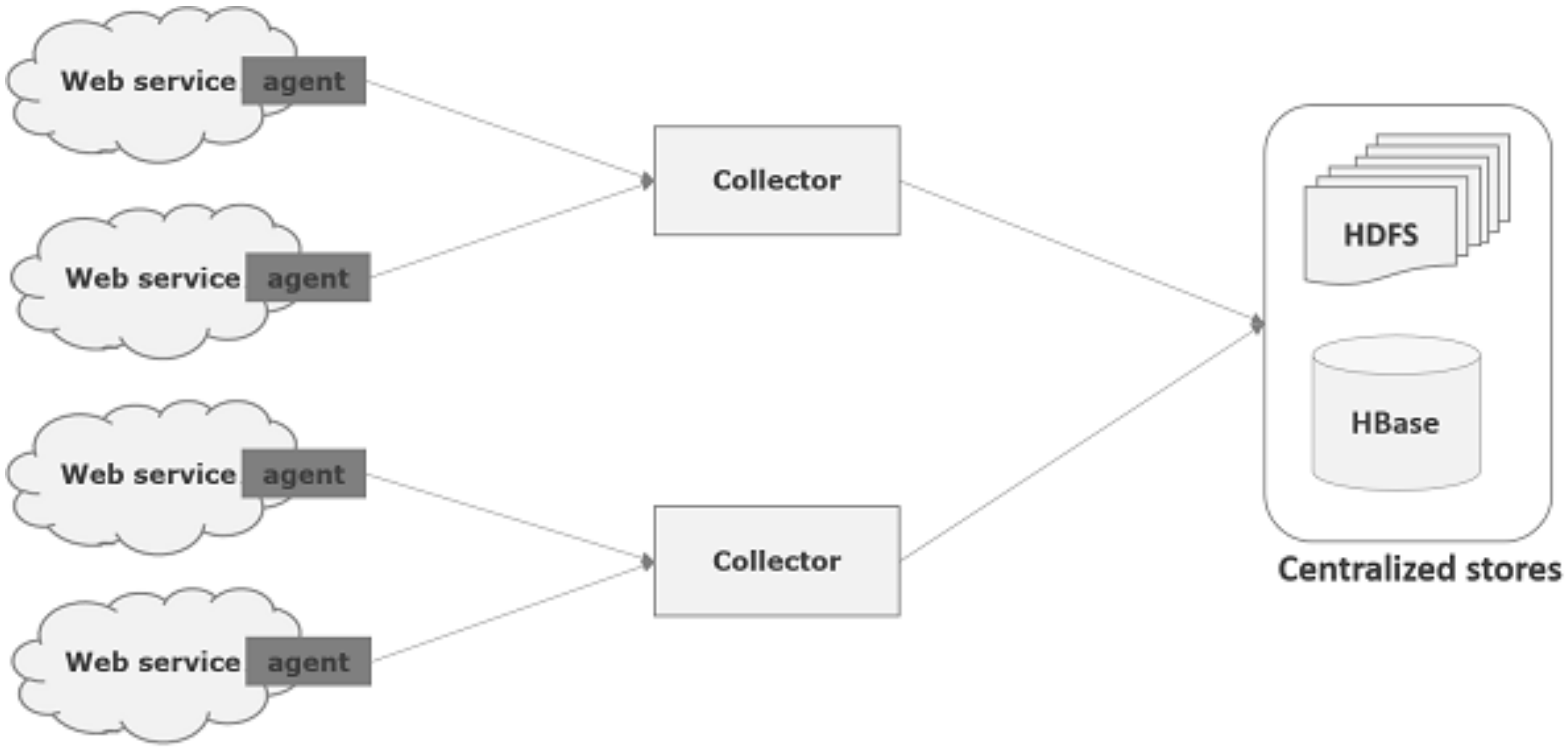
Sink Processors

Flume Agent : Source/Channel/Sink Type

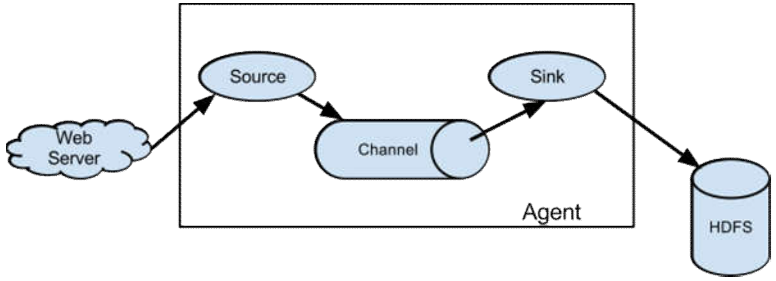
Sources	Channels	Sinks
Avro Source	Memory Channel	HDFS Sink
Thrift Source	JDBC Channel	Hive Sink
Exec Source	Kafka Channel	Logger Sink
JMS Source	File Channel	Avro Sink
Spooling Directory Source	Spillable Memory Channel	Thrift Sink
Twitter 1% firehose Source	Pseudo Transaction Channel	IRC Sink
Kafka Source		File Roll Sink
NetCat Source		Null Sink
Sequence Generator Source		HBaseSink
Syslog Sources		AsyncHBaseSink
Syslog TCP Source		MorphlineSolrSink
Multiport Syslog TCP Source		ElasticSearchSink
Syslog UDP Source		Kite Dataset Sink
HTTP Source		Kafka Sink
Stress Source		
Legacy Sources		
Thrift Legacy Source		
Custom Source		
Scribe Source		

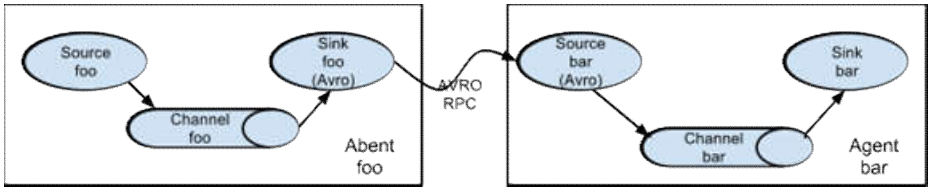
https://www.tutorialspoint.com/apache_flume/apache_flume_configuration.htm

Flume Data Flow Model:

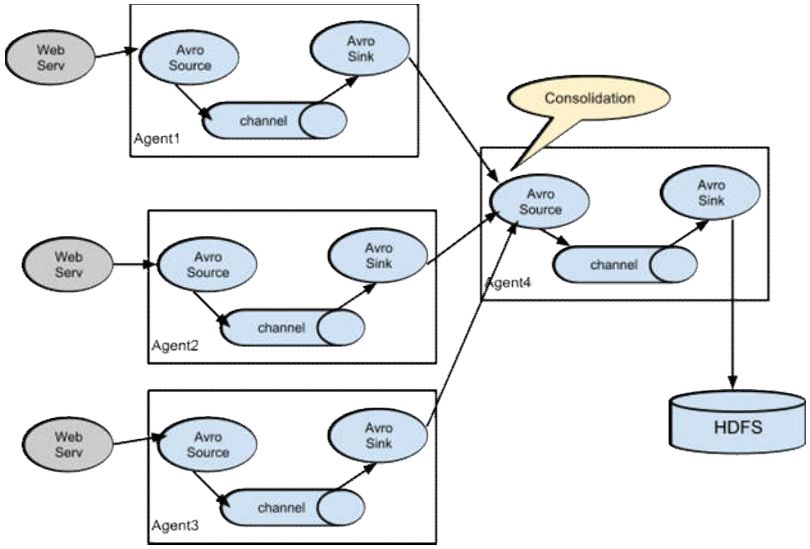


Flume Data Flow Model:

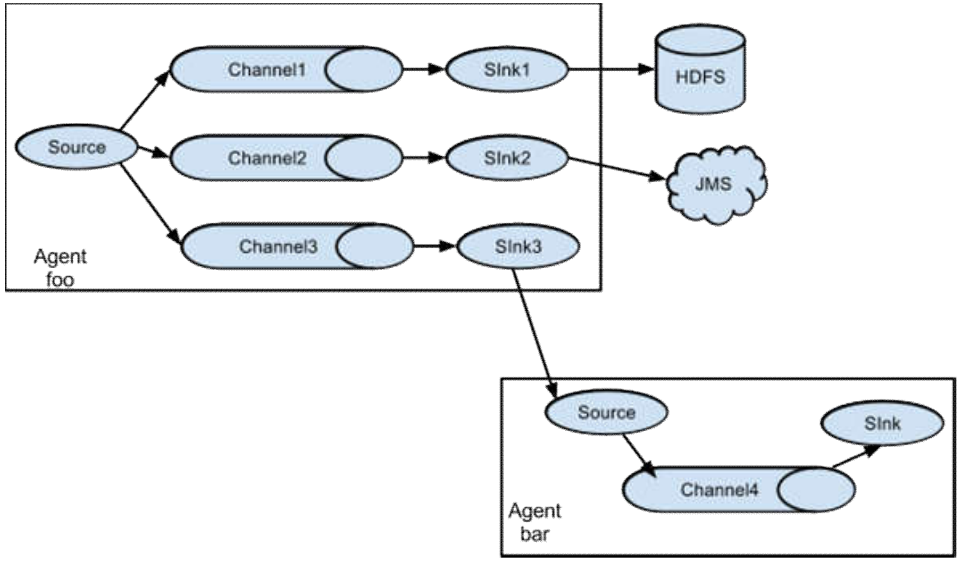




multi-agent



Consolidation



Multiplexing

Flume Properties file :

```
# example.conf: A single-node Flume configuration

# Name the components on this agent

a1.sources = r1

a1.sinks = k1

a1.channels = c1


# Describe/configure the source

a1.sources.r1.type = netcat

a1.sources.r1.bind = localhost

a1.sources.r1.port = 44444
```

```
# Continue:

# Describe the sink

a1.sinks.k1.type = logger


# Use a channel which buffers events

# in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100


# Bind the source and sink to the channel

a1.sources.r1.channels = c1

a1.sinks.k1.channel = c1
```

Flume Properties file : [กิจกรรม]

 # แปลง Configuration file เป็นรูป Architecture :

Flume Installation :

- Download & Installation

  $ wget http://www-eu.apache.org/dist/flume/1.7.0/apache-flume-1.7.0-bin.tar.gz

  $ tar vxf apache-flume-1.7.0-bin.tar.gz

  $ mv apache-flume-1.7.0-bin /usr/local/flume

  $ chown -R /usr/local/flume

- Environment Variable

  $nano .bashrc

  export FLUME_HOME=/usr/local/flume

  export PATH=$PATH:$FLUME_HOME/bin

  export CLASSPATH=$CLASSPATH:$FLUME_HOME/lib/*

Flume Installation :

- Configuration Flume : /usr/local/flume/conf

  - flume-conf.properties.template

  - flume-env.sh.template

  - flume-env.ps1.template

  - log4j.properties

-- สร้าง Config. file สำที่จะใช้งาน จากการ copy template ที่มีอยู่

:/usr/local/flume/conf$cp flume-conf.properties.template flume-conf.properties

:/usr/local/flume/conf$cp flume-env.sh.template flume-env.sh

Run Flume Agent : Hand-On#1

@ Flume Agent , receive some message on telnet and print out to screen on screen console

Start Flume Agent:

$ bin/flume-ng agent --conf conf --conf-file </path/config-name.conf> --name <Agent Name> -Dflume.root.logger=INFO,console

Run Flume Agent : Hand-On#1

@ Flume Agent , receive some message on telnet and print out to screen on screen console

Start Flume Agent:

$ bin/flume-ng agent --conf conf --conf-file example.conf --name a1 -Dflume.root.logger=INFO,console

UbuntuServer [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
ters:{} } }} channels:{c1=org.apache.flume.channel.MemoryChannel{name: c1}} }
17/05/28 20:52:04 INFO node.Application: Starting Channel c1
17/05/28 20:52:04 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: CHAN
NEL, name: c1: Successfully registered new MBean.
17/05/28 20:52:04 INFO instrumentation.MonitoredCounterGroup: Component type: CHANNEL, name: c1 star
ted
17/05/28 20:52:04 INFO node.Application: Starting Sink k1
17/05/28 20:52:04 INFO node.Application: Starting Source r1
17/05/28 20:52:04 INFO source.NetcatSource: Source starting
17/05/28 20:52:04 INFO source.NetcatSource: Created serverSocket:sun.nio.ch.ServerSocketChannelImpl[
/127.0.0.1:44444]
17/05/28 20:52:34 INFO sink.LoggerSink: Event: {
1 0D          Hello world!. }
17/05/28 20:59:44 INFO sink.LoggerSink: Event: {
4 6F 20 6D 75 I love you to mu }
17/05/28 21:00:06 INFO sink.LoggerSink: Event: {
              i. }
17/05/28 21:00:09 INFO sink.LoggerSink: Event: {
              i. }
17/05/28 21:00:10 INFO sink.LoggerSink: Event: {
              i. }
17/05/28 21:00:11 INFO sink.LoggerSink: Event: {
              u. }
17/05/28 21:00:12 INFO sink.LoggerSink: Event: {
              u. }
17/05/28 21:00:14 INFO sink.LoggerSink: Event: { headers:{} body: 75 0D
              u. }
17/05/28 21:00:15 INFO sink.LoggerSink: Event: { headers:{} body: 69 0D
              i. }
17/05/28 21:00:20 INFO sink.LoggerSink: Event: { headers:{} body: 78 0D
              x. }
17/05/28 21:00:22 INFO sink.LoggerSink: Event: { headers:{} body: 79 0D
              y. }
17/05/28 21:01:01 INFO sink.LoggerSink: Event: { headers:{} body: 48 65 6C 6C 6F 20 77 6F 72 6C 64 2
1 0D          Hello world!. }
```

auoychai@ubtserver: /usr/local/flume/bin

```
auoychai@ubtserver:/usr/local/flume/bin$ telnet localhost 44444
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello World!
OK
```

Run Flume Agent : Hand-On#2

create HDFS directory

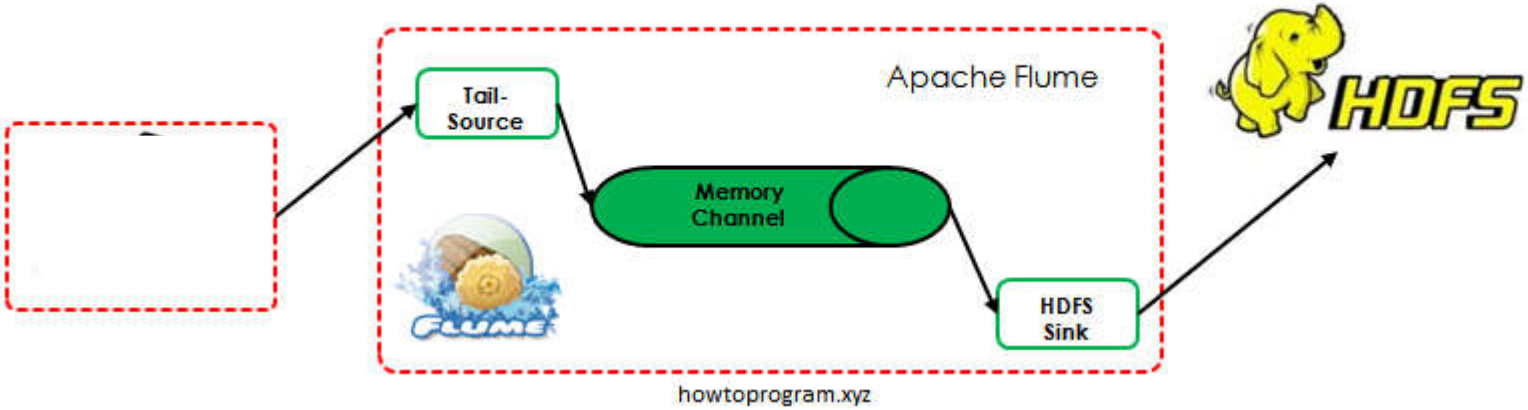
/user/flume/logs

/user/flume/events

/user/flume/tweets

Run Flume Agent : Hand-On#2

@ Read Message on telnet interface and keep result to HDFS



Run Flume Agent : Hand-On#2

@ Read Message on telnet interface and keep result to HDFS

1). Create output Directory :

- HDFS: /user/flume/events
- FOS:/var/log/flume-ng

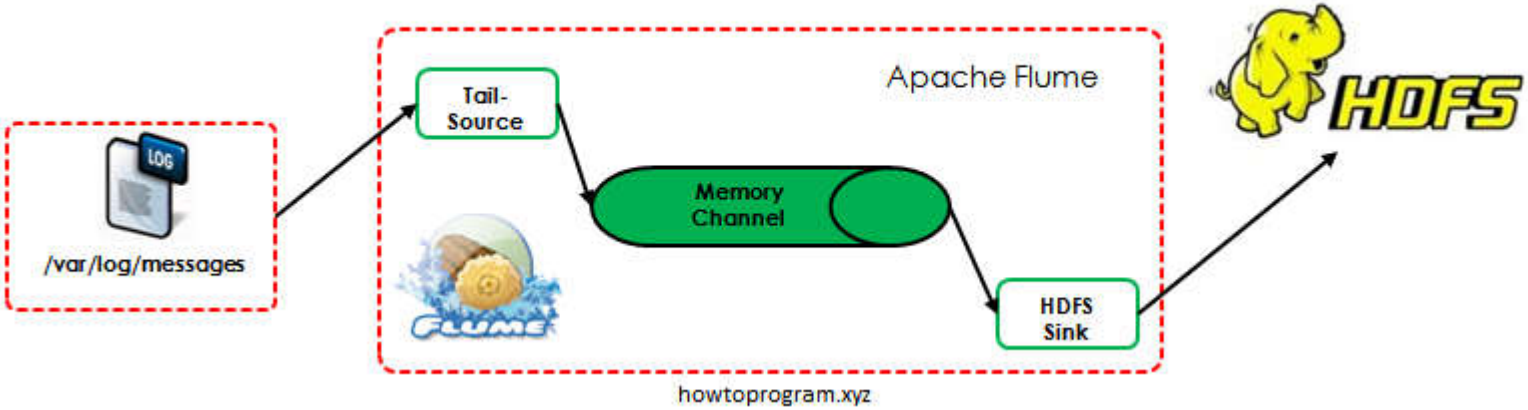
2). To receive the hadoop log file and write to os file sysem and hadoop hdfs file.

- Hadoop Log file location : /var/log/hadoop/hadoop-auoychai-namenode-ubtserver.log

3). Set Flume Configuration:

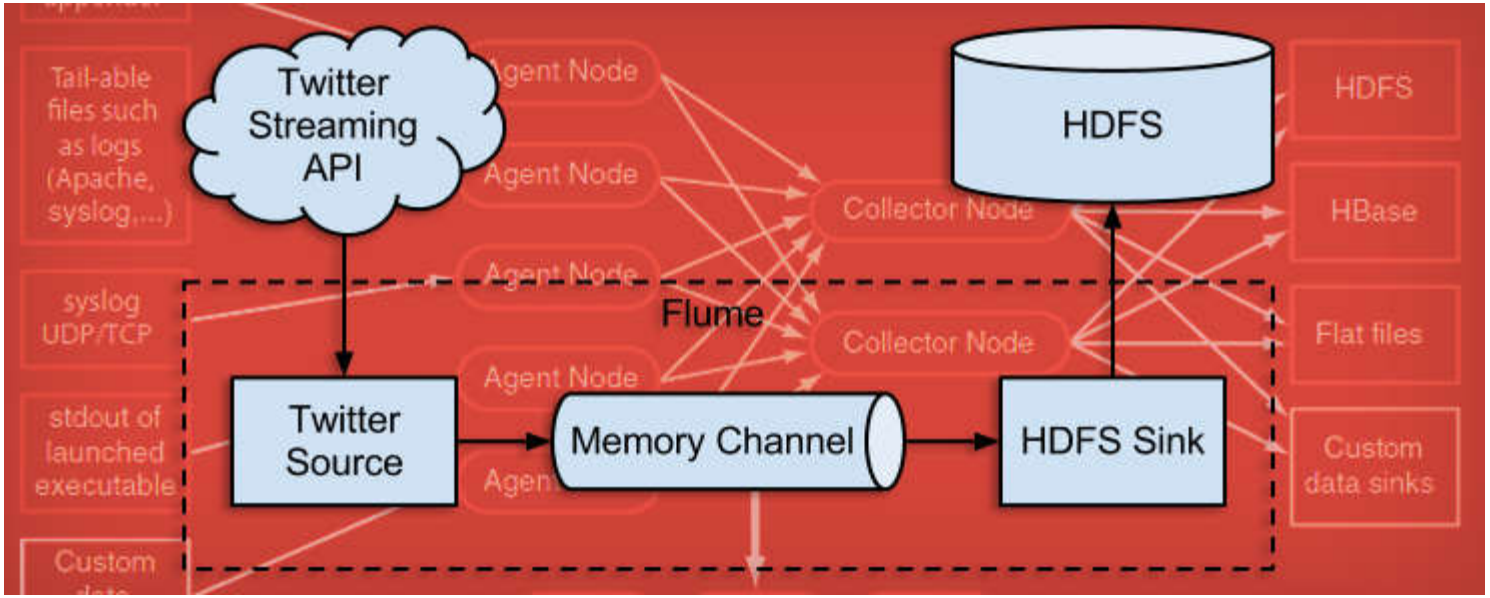
Run Flume Agent : Hand-On#2

@ Read log file and sent result to HDFS



Run Flume Agent : Hand-On#3

@ Fetching Twitter Data

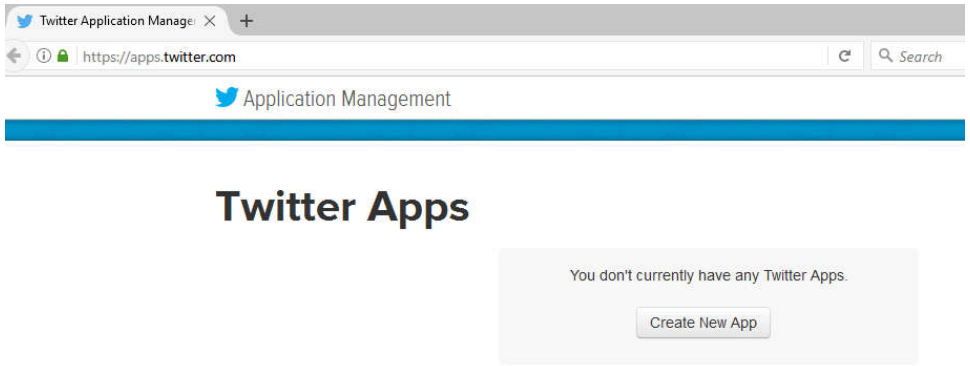


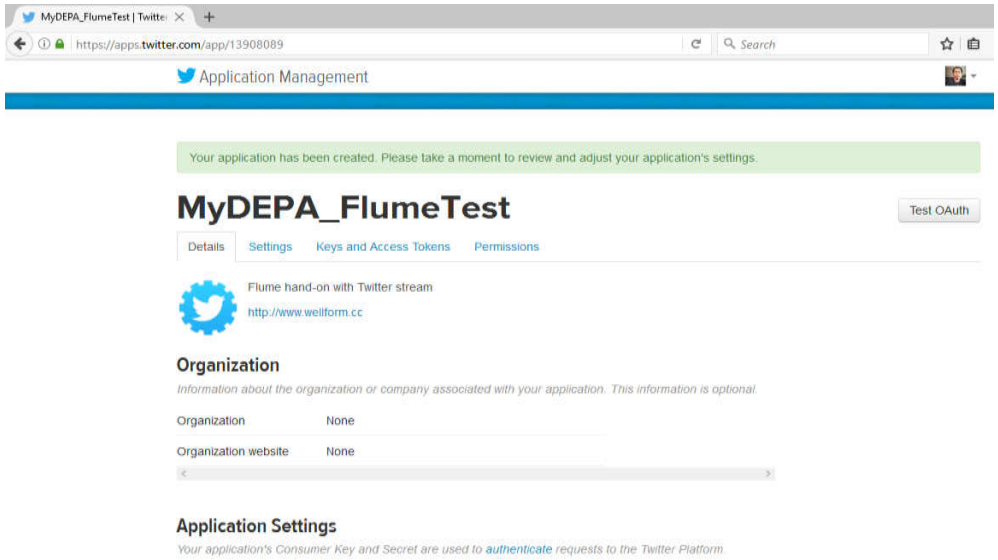
https://cdnp-2f3a.kxcdn.com/blog/wp-content/uploads/2015/06/Flume-Installation-and-Streaming-Twitter-Data-Using-Flume.png

Run Flume Agent : Hand-On#3

1). create a Twitter application

https://apps.twitter.com/





Run Flume Agent : Hand-On#3

1). Create HDFS directory

/user/twitter_data


The End
Big data
Shift
Del
Prt