DHRUV GARG 16 BCE 1190

LARGE SCALE DATA PROCESSING – DIGITAL ASSIGNMENT 2

CASE STUDY: TWITTER DATA STREAMING USING APACHE FLUME

[BEFORE the Twitter data streaming implementation, we understand the concepts of Flume]

Flume: Introduction

Apache Flume is a service for streaming logs into Hadoop. Flume is distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of streaming data into the Hadoop Distributed File System (HDFS). It has a simple and flexible architecture based on streaming data flows; and is robust and fault tolerant with tuneable reliability mechanisms for failover and recovery. YARN coordinates data ingest from Apache Flume and other services that deliver raw data into a Hadoop cluster.

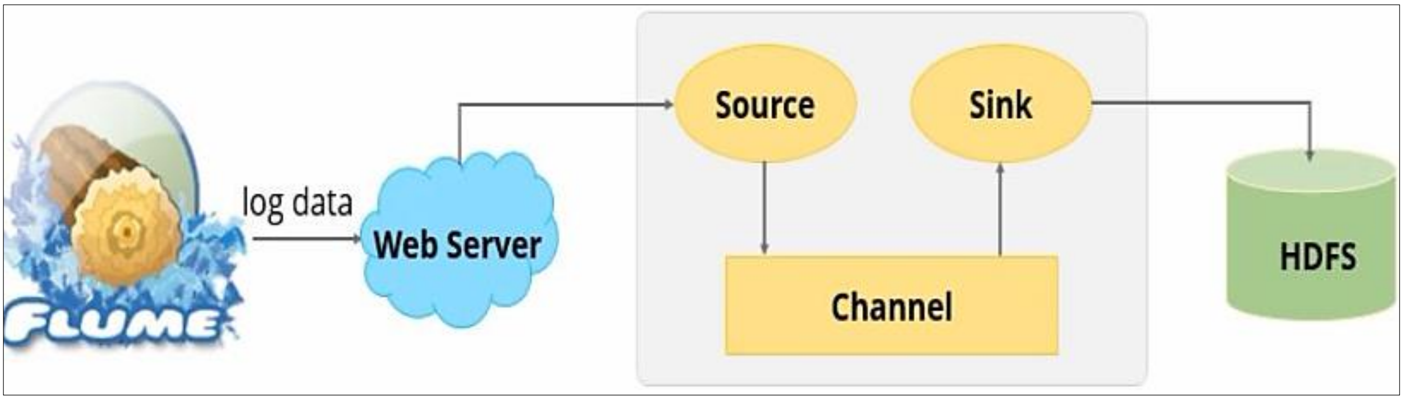
Flume: Features

Flume lets Hadoop users ingest high-volume streaming data into HDFS for storage. Specifically, Flume allows users to:

FEATURES	DESCRIPTION
Stream data	Ingest streaming data from multiple sources into Hadoop for storage and analysis.
Insulate systems	Buffer storage platform from transient spikes, when the rate of incoming data exceeds the rate at which data can be written to the destination.
Guarantee data delivery	Flume NG uses channel-based transactions to guarantee reliable message delivery. When a message moves from one agent to another, two transactions are started, one on the agent that delivers the event and the other on the agent that receives the event. This ensures guaranteed delivery semantics.
Scale horizontally	To ingest new data streams and additional volume as needed.

Enterprises use Flume's powerful streaming capabilities to land data from high-throughput streams in the HDFS. Typical sources of these streams are application logs, sensor and machine data, geo-location data and social media. These different types of data can be landed in Hadoop for future analysis using interactive queries in Apache Hive. Or they can feed business dashboards served ongoing data by Apache HBase.

Example usecase: Flume is used to log manufacturing operations. When one run of product comes off the line, it generates a log file about that run. Even if this occurs hundreds or thousands of times per day, the large volume log file data can stream through Flume into a tool for same-day analysis with Apache Storm or months or years of production runs can be stored in HDFS and analysed by a quality assurance engineer using Apache Hive.



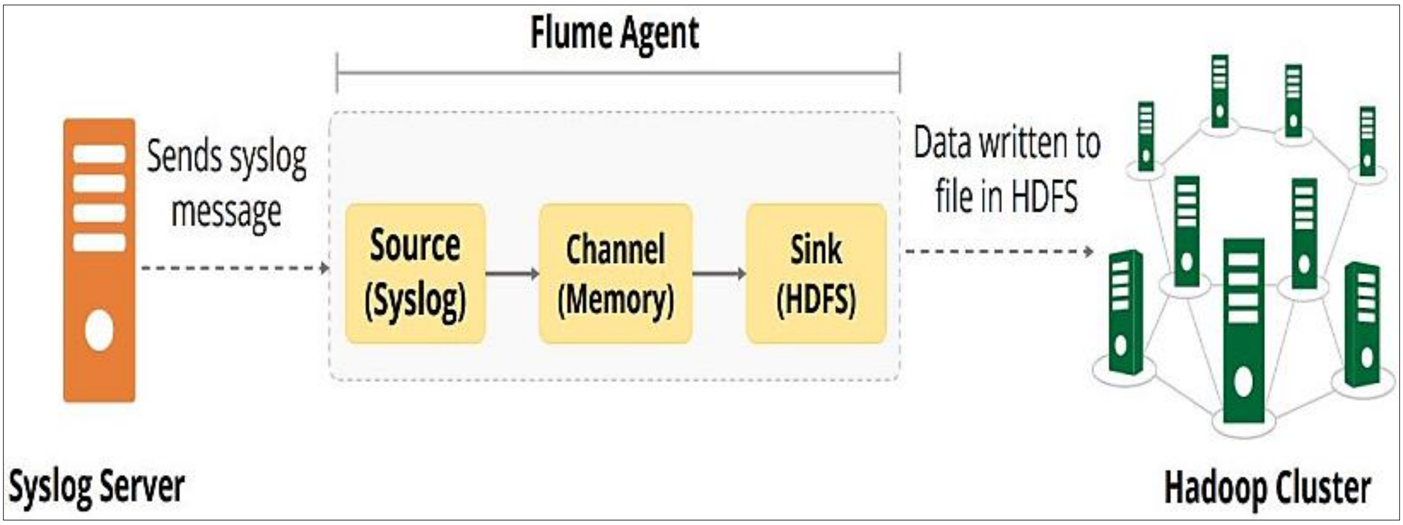
Flume: Working and operations

Flume's high-level architecture is built on a streamlined codebase that is easy to use and extend. The project is highly reliable, without the risk of data loss. Flume also supports dynamic reconfiguration without the need for a restart, which reduces downtime for its agents. The following components make up Apache Flume:

COMPONENT	DEFINITION
Event	A singular unit of data that is transported by Flume (typically a single log entry).
Source	The entity through which data enters into Flume. Sources either actively poll for data or passively wait for data to be delivered to them. A variety of sources allow data to be collected, such as log4j logs and syslogs.
Sink	The entity that delivers the data to the destination. A variety of sinks allow data to be streamed to a range of destinations. One example is the HDFS sink that writes events to HDFS.
Channel	The conduit between the Source and the Sink. Sources ingest events into the channel and the sinks drain the channel.
Agent	Any physical Java virtual machine running Flume. It is a collection of sources, sinks and channels.
Client	The entity that produces and transmits the Event to the Source operating within the Agent.

Flume components interact in the following way:

1. A flow in Flume starts from the **Client**.
2. The **Client** transmits the **Event** to a **Source** operating within the **Agent**.
3. The **Source** receiving this **Event** then delivers it to one or more **Channels**.
4. One or more **Sinks** operating within the same **Agent** drains these **Channels**.
5. **Channels** decouple the ingestion rate from drain rate using the familiar producer-consumer model of data exchange.
6. When spikes in client side activity cause data to be generated faster than can be handled by the provisioned destination capacity can handle, the **Channel** size increases. This allows sources to continue normal operation for the duration of the spike.
7. The **Sink** of one **Agent** can be chained to the **Source** of another **Agent**. This chaining enables the creation of complex data flow topologies.



How is Flume easily scalable?

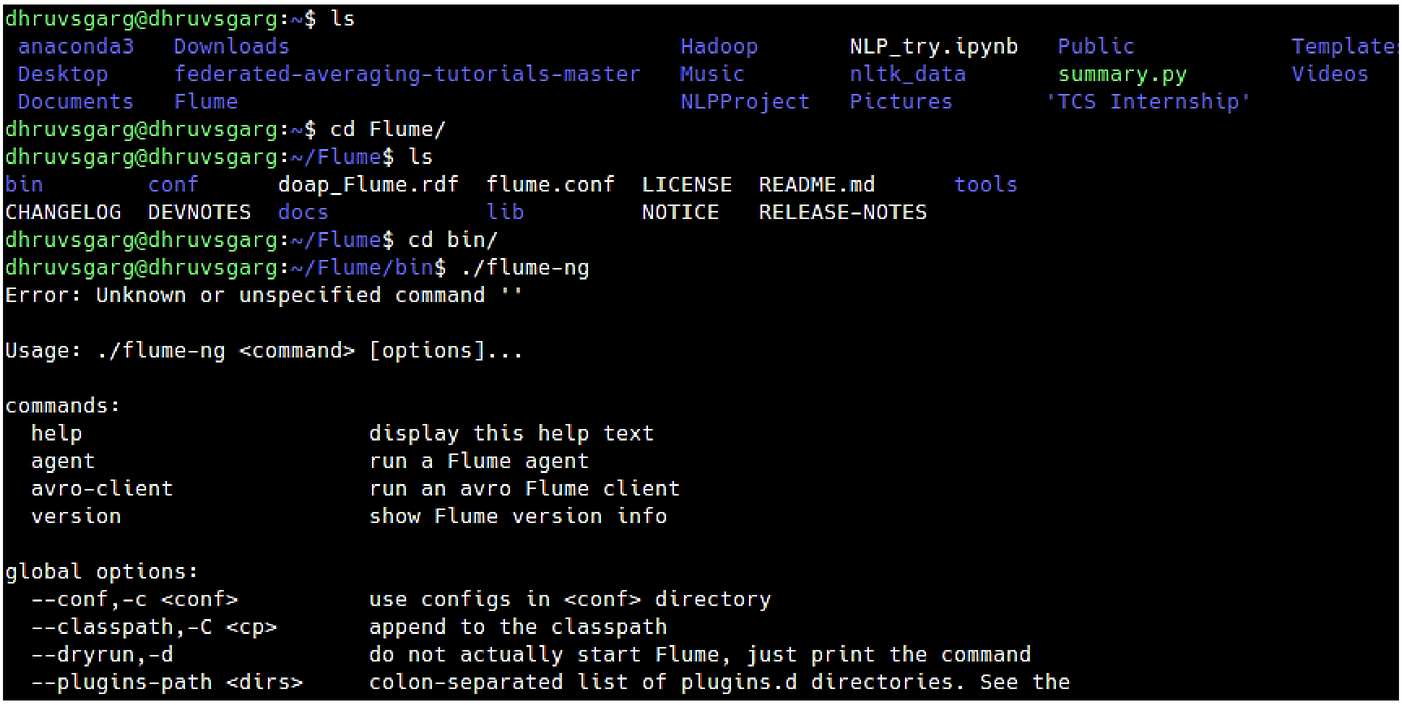
Because Flume's distributed architecture requires no central coordination point. Each agent runs independently of others with no inherent single point of failure, and Flume can easily scale horizontally.

FLUME CASE STUDY: TWITTER DATA STREAMING

STEP 1:

Setting up Apache Flume: We download and install **Flume 1.9**. We also set the appropriate JAVA_HOME path and the FLUME_HOME path in the .bashrc file.

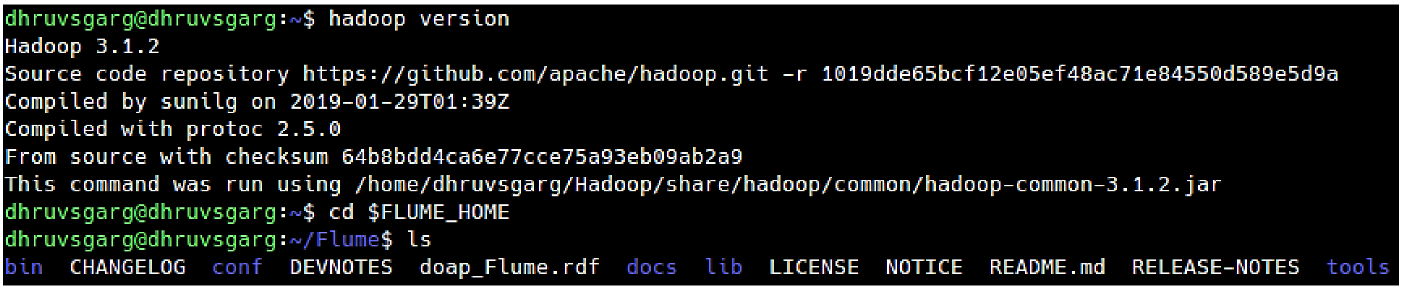
To test if Flume has been installed correctly, we try executing ./flume-ng. On successful installation, we get the following output.



```
dhruvsgarg@dhruvsgarg:~$ ls
 anaconda3   Downloads                               Hadoop       NLP_try.ipynb   Public           Template
 Desktop     federated-averaging-tutorials-master    Music        nltk_data       summary.py       Videos
 Documents   Flume                                   NLPProject   Pictures       'TCS Internship'
dhruvsgarg@dhruvsgarg:~$ cd Flume/
dhruvsgarg@dhruvsgarg:~/Flume$ ls
bin         conf       doap_Flume.rdf  flume.conf  LICENSE  README.md      tools
CHANGELOG   DEVNOTES   docs            lib         NOTICE   RELEASE-NOTES
dhruvsgarg@dhruvsgarg:~/Flume$ cd bin/
dhruvsgarg@dhruvsgarg:~/Flume/bin$ ./flume-ng
Error: Unknown or unspecified command ''

Usage: ./flume-ng <command> [options]...

commands:
  help                        display this help text
  agent                       run a Flume agent
  avro-client                 run an avro Flume client
  version                     show Flume version info

global options:
  --conf,-c <conf>            use configs in <conf> directory
  --classpath,-C <cp>         append to the classpath
  --dryrun,-d                 do not actually start Flume, just print the command
  --plugins-path <dirs>       colon-separated list of plugins.d directories. See the
```

STEP 2:

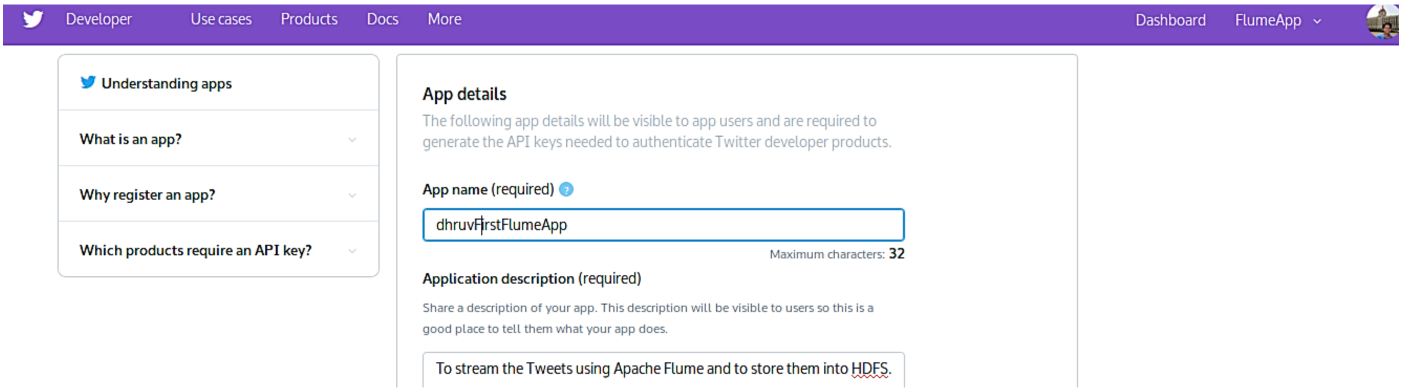
Setting up Hadoop version 3.2: We download and install **Hadoop 3.1.2.** We also set the appropriate JAVA_HOME path and HADOOP_HOME path along with YARN_HOME, etc. We must also manually add content to the various .xml files to set the appropriate configuration settings.



```
dhruvsgarg@dhruvsgarg:~$ hadoop version
Hadoop 3.1.2
Source code repository https://github.com/apache/hadoop.git -r 1019dde65bcf12e05ef48ac71e84550d589e5d9a
Compiled by sunilg on 2019-01-29T01:39Z
Compiled with protoc 2.5.0
From source with checksum 64b8bdd4ca6e77cce75a93eb09ab2a9
This command was run using /home/dhruvsgarg/Hadoop/share/hadoop/common/hadoop-common-3.1.2.jar
dhruvsgarg@dhruvsgarg:~$ cd $FLUME_HOME
dhruvsgarg@dhruvsgarg:~/Flume$ ls
bin  CHANGELOG  conf  DEVNOTES  doap_Flume.rdf  docs  lib  LICENSE  NOTICE  README.md  RELEASE-NOTES  tools
```

STEP 3:

Creating a Twitter application: After logging into Twitter, we create a new application using https://apps.twitter.com. Here I have created an application named **dhruvFirstFlumeApp.**



App details

The following app details will be visible to app users and are required to generate the API keys needed to authenticate Twitter developer products.

App name (required)

dhruvFirstFlumeApp

Maximum characters: 32

Application description (required)

Share a description of your app. This description will be visible to users so this is a good place to tell them what your app does.

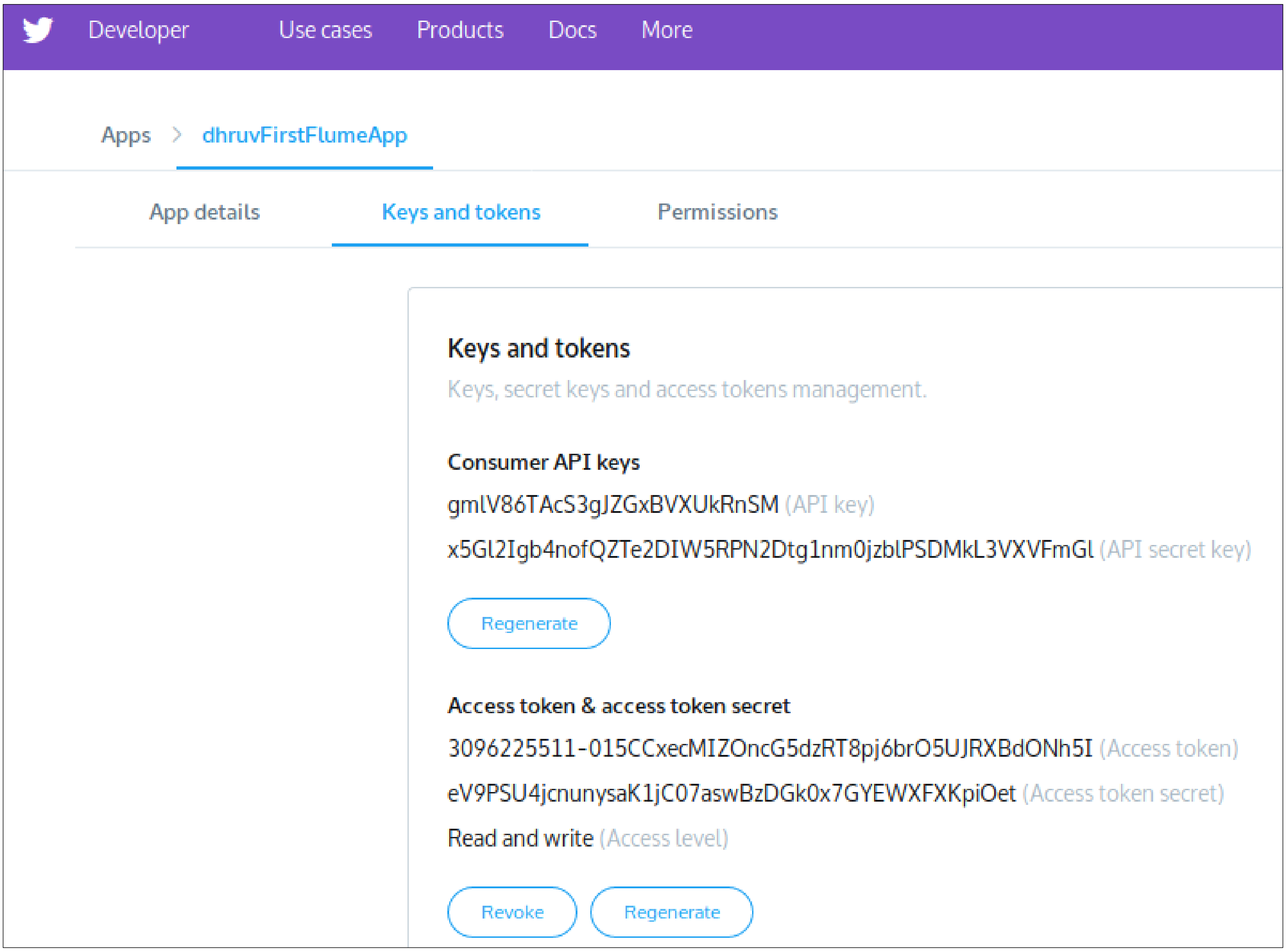
To stream the Tweets using Apache Flume and to store them into HDFS.

STEP 4:

After creating the application, we can **access the keys and tokens** of the application.

There are 4 important application settings:

1. Consumer Key (API Key)
2. Consumer Secret (API Secret)
3. Access Token
4. Access Token Secret.



Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

gmlV86TAcS3gJZGxBVXUkRnSM (API key)

x5Gl2Igb4nofQZTe2DIW5RPN2Dtg1nm0jzblPSDMkL3VXVFmGl (API secret key)

Access token & access token secret

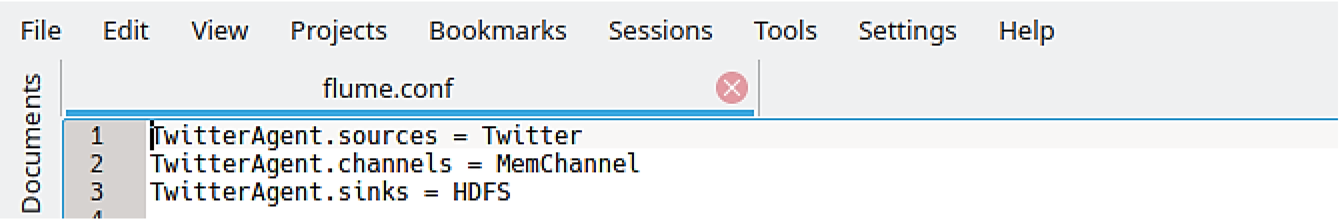
3096225511-015CCxecMIZOncG5dzRT8pj6brO5UJRXBdONh5I (Access token)

eV9PSU4jcnunysaK1jC07aswBzDGk0x7GYEWXFXKpiOet (Access token secret)

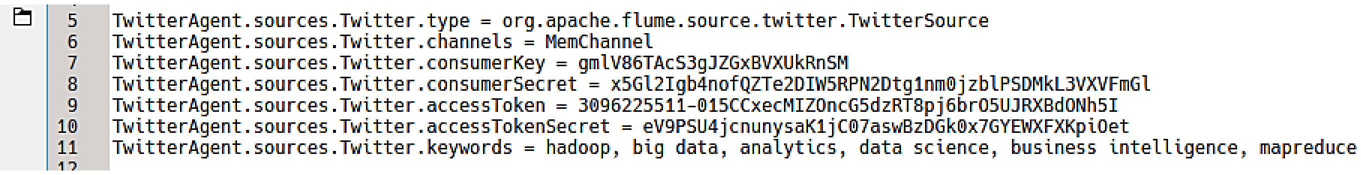
Read and write (Access level)

STEP 5:

We create a **flume.conf file** in Flume’s root directory as shown below. As discussed in the Flume Architecture, we configure our Source, Sink and Channel. In this application, the Source is Twitter, from where we are streaming our data and our Sink is HDFS, where we are writing the data.

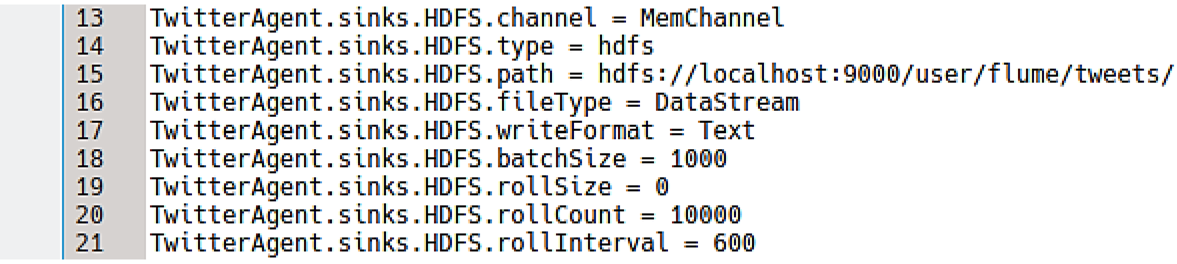


```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS
```



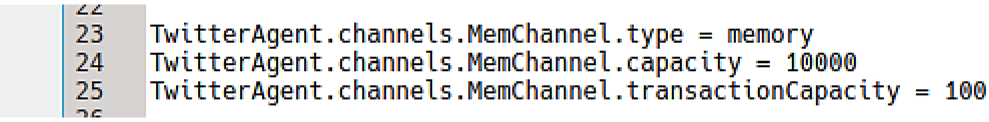
```
TwitterAgent.sources.Twitter.type = org.apache.flume.source.twitter.TwitterSource
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sources.Twitter.consumerKey = gmlV86TAcS3gJZGxBVXUkRnSM
TwitterAgent.sources.Twitter.consumerSecret = x5Gl2Igb4nofQZTe2DIW5RPN2Dtg1nm0jzblPSDMkL3VXVFmGl
TwitterAgent.sources.Twitter.accessToken = 3096225511-015CCxecMIZOncG5dzRT8pj6brO5UJRXBdONh5I
TwitterAgent.sources.Twitter.accessTokenSecret = eV9PSU4jcnunysaK1jC07aswBzDGk0x7GYEWXFXKpiOet
TwitterAgent.sources.Twitter.keywords = hadoop, big data, analytics, data science, business intelligence, mapreduce
```

Source properties: In source configuration we are passing the Twitter source type as org.apache.flume.source.twitter.TwitterSource. Then, we are passing all the four tokens which we received from Twitter. At last in source configuration we are passing the keywords on which we are going to fetch the tweets.



```
TwitterAgent.sinks.HDFS.channel = MemChannel
TwitterAgent.sinks.HDFS.type = hdfs
TwitterAgent.sinks.HDFS.path = hdfs://localhost:9000/user/flume/tweets/
TwitterAgent.sinks.HDFS.fileType = DataStream
TwitterAgent.sinks.HDFS.writeFormat = Text
TwitterAgent.sinks.HDFS.batchSize = 1000
TwitterAgent.sinks.HDFS.rollSize = 0
TwitterAgent.sinks.HDFS.rollCount = 10000
TwitterAgent.sinks.HDFS.rollInterval = 600
```

Sink properties: In the sink configuration, we set the type, path, file type, batch size and roll parameters.



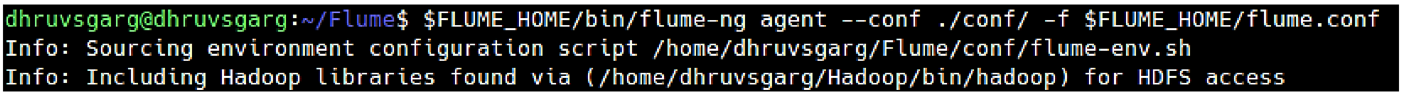
```
TwitterAgent.channels.MemChannel.type = memory
TwitterAgent.channels.MemChannel.capacity = 10000
TwitterAgent.channels.MemChannel.transactionCapacity = 100
```

Channel properties: In the channel properties, we set the memory, capacity and the capacity of each transaction.

STEP 6:

After setting up all the required tools, we are **set for execution**. We perform the Twitter Data Streaming execution using the following command:

$FLUME_HOME/bin/flume-ng agent --conf ./conf/ -f $FLUME_HOME/flume.conf

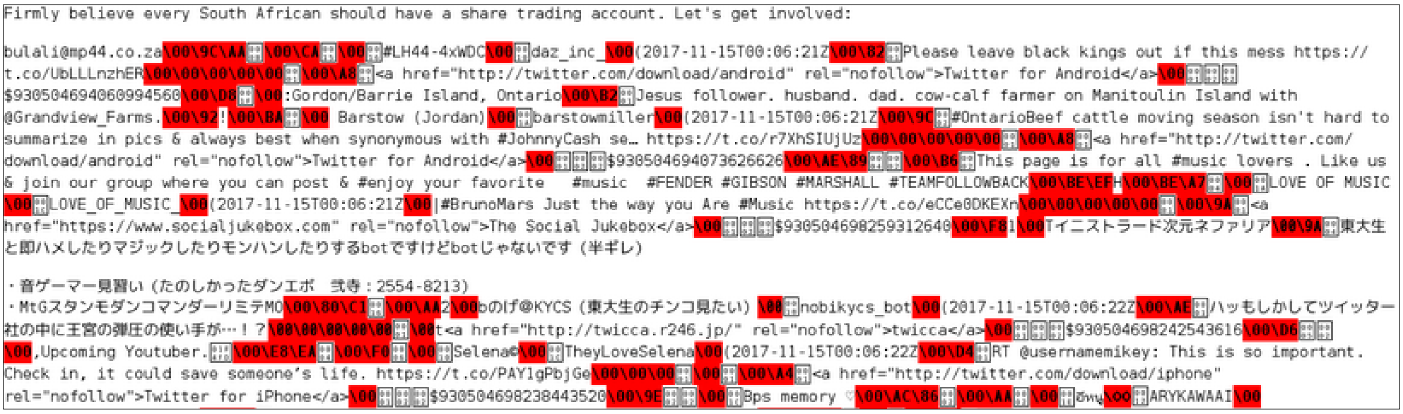


```
dhruvsgarg@dhruvsgarg:~/Flume$ $FLUME_HOME/bin/flume-ng agent --conf ./conf/ -f $FLUME_HOME/flume.conf
Info: Sourcing environment configuration script /home/dhruvsgarg/Flume/conf/flume-env.sh
Info: Including Hadoop libraries found via (/home/dhruvsgarg/Hadoop/bin/hadoop) for HDFS access
```

We stop the execution by entering Ctrl+C after some time.

STEP 7:

After some time, the **execution completes** and the data has been streamed into HDFS from Twitter.



```
Firmly believe every South African should have a share trading account. Let's get involved:

bulali@mp44.co.za\00\9C\AA□\00\CA□\00□#LH44-4xWDC\00□daz_inc_\00(2017-11-15T00:06:21Z\00\82□Please leave black kings out if this mess https://
t.co/UbLLLnzhER\00\00\00\00\00□\00\A8□<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>\00□□□
$930504694060994560\00\D8□\00:Gordon/Barrie Island, Ontario\00\B2□Jesus follower. husband. dad. cow-calf farmer on Manitoulin Island with
@Grandview_Farms.\00\92!\00\BA□\00 Barstow (Jordan)\00□barstowmiller\00(2017-11-15T00:06:21Z\00\9C□#OntarioBeef cattle moving season isn't hard to
summarize in pics & always best when synonymous with #JohnnyCash se… https://t.co/r7XhSIUjUz\00\00\00\00\00□\00\A8□<a href="http://twitter.com/
download/android" rel="nofollow">Twitter for Android</a>\00□□□$930504694073626626\00\AE\89□□\00\B6□This page is for all #music lovers . Like us
& join our group where you can post & #enjoy your favorite   #music  #FENDER #GIBSON #MARSHALL #TEAMFOLLOWBACK\00\BE\EFH\00\BE\A7□\00□LOVE OF MUSIC
\00□LOVE_OF_MUSIC_\00(2017-11-15T00:06:21Z\00|#BrunoMars Just the way you Are #Music https://t.co/eCCe0DKEXn\00\00\00\00\00□\00\9A□<a
href="https://www.socialjukebox.com" rel="nofollow">The Social Jukebox</a>\00□□□$930504698259312640\00\F81\00Tイニストラード次元ネファリア\00\9A□東大生
と即ハメしたりマジックしたりモンハンしたりするbotですけどbotじゃないです（半ギレ）

・音ゲーマー見習い（たのしかったダンエボ　弐寺：2554-8213）
・MtGスタンモダンコマンダーリミテMO\00\80\C1□\00\AA2\00bのげ@KYCS（東大生のチンコ見たい）\00□nobikycs_bot\00(2017-11-15T00:06:22Z\00\AE□ハッもしかしてツイッター
社の中に王宮の弾圧の使い手が…！？\00\00\00\00\00□\00t<a href="http://twicca.r246.jp/" rel="nofollow">twicca</a>\00□□□$930504698242543616\00\D6□□
\00,Upcoming Youtuber.□\00\E8\EA□\00\F0□\00□Selena©\00□TheyLoveSelena\00(2017-11-15T00:06:22Z\00\D4□RT @usernamemikey: This is so important.
Check in, it could save someone's life. https://t.co/PAY1gPbjGe\00\00\00□\00□\00\A4□<a href="http://twitter.com/download/iphone"
rel="nofollow">Twitter for iPhone</a>\00□□□$930504698238443520\00\9E□□\00□8ps memory ♡\00\AC\86□\00\AA□\00□ðwų\00□ARYKAWAAI\00
```

RESULT

Thus Flume was successfully used for Twitter Data Streaming and the output was stored in HDFS.

OTHER APPLICATIONS OF APACHE FLUME

- Flume is used in e-commerce companies to analyse the customer behaviour of different regions.
- It is used to feed huge log data generated by application servers into HDFS at a higher speed.
- Flume is used in online analytics application hence it is a backbone for real-time event processing.
- Flume efficiently feed log data files from multiple web servers into a centralized store i.e HDFS or HBase.
- Flume can process data in-flight using interceptors. These can be very useful for data masking or filtering.
- Flume is distributed in nature i.e agents can be installed on many machine.
- Flume is able to collect real-time and batch mode data from multiple servers and immediately feed into Hadoop.
- Flume is also used to import huge volumes of event data (instantaneous values) produced by social networking sites like Facebook and Twitter, and e-commerce websites like Amazon and Flipkart.