

Session 12: Project 1

Flume Use Case Solution

*Session 12:* Flume Use Case Solution

**Table of Contents**

[1. Introduction](#_gjdgxs)

[2. Problem Solution](#_30j0zll)

[3. Output](https://docs.google.com/document/d/1rcAmMgVWsOF195vB5u1JDytPcrDB9-EudQ7lOYc7E6A/edit#heading=h.3znysh7)

# **Introduction**

This Project will help you to consolidate the concepts learnt in the session.

# **Problem Solution**

**Create a flume agent that streams data from Twitter and stores in the HDFS.**

We all know that Hadoop is a framework which helps in parallel processing using huge datasets called [Big Data](https://acadgild.com/big-data/big-data-development-training-certification). This is just a basic introduction/definition of Hadoop and you can find tons of stuff on Hadoop, over internet. So, instead of discussing more on Hadoop, we will focus on the data which Hadoop will fetch from various sources using Flume.

### **So, what does Flume actually do?**

Apache Flume is a reliable service used for efficiently collecting, aggregating, and moving large amounts of log data.

The next question on your mind would be, “How does Flume do this?”

*Read on to find out the answer for your question.*

### **How Flume helps Hadoop to get data from live streaming?**

Flume allows the user to do the following:

* Stream data into Hadoop from multiple sources for analysis.
* Collect high-volume web logs in real-time.
* It acts as a buffer when the rate of incoming data exceeds the rate at which the data can be written. Thereby preventing data loss.
* Guarantees data delivery.
* Scales horizontally (connects commodity system in parallel) to handle additional data volume.

### **Essential Components Involved in Getting Data from a Live-Streaming Source**

There are 3 major components, namely: Source, Channel, and Sink, which are involved in ingesting data, moving data and storing data, respectively.



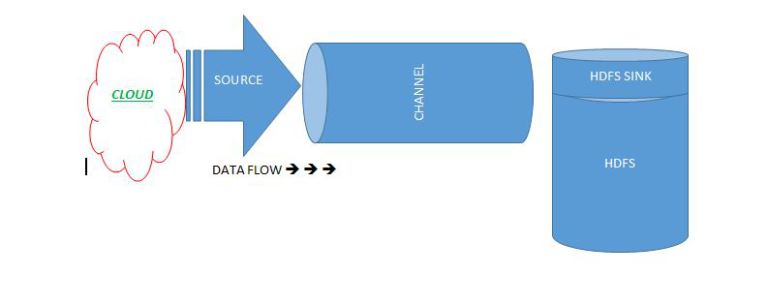
## **Fascinated by latest technology?**

Subscribe to our blog and we'll keep you updated with latest technologies like AngularJS, Machine Learning, Big Data, Spark, Android, Analytics, NodeJS, Cloud Computing and many more!

Subscribe, It’s Free!

Below is the breakdown of the parts applicable in this scenario:

* Event – A singular unit of data that is transported by Flume (typically a single log entry).
* Source – The entity through which data enters into the Flume. Sources either actively samples the data or passively waits for data to be delivered to them. A variety of sources such as log4j logs and syslogs, allows data to be collected.
* Sink – The unit that delivers the data to the destination. A variety of sinks allow data to be streamed to a range of destinations. Example: HDFS sink writes events to the HDFS.
* Channel – It is the connection between the Source and the Sink. The Source ingests Event into the Channel and the Sink drains the Channel.
* Agent – Any physical Java virtual machine running Flume. It is a collection of Sources, Sinks and Channels.
* Client – It produces and transmits the Event to the Source operating within the Agent



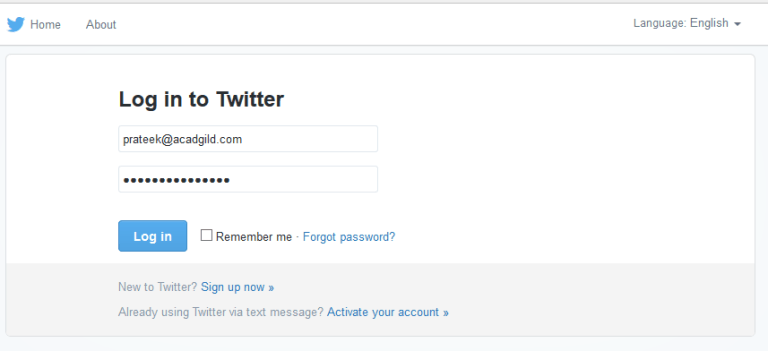
### **How to Take Input Data from Twitter to the HDFS?**

Before we discuss how to take the input data from Twitter to HDFS, let’s look at the necessary pre-requisites:

* Twitter account
* Install Hadoop/Start Hadoop

### **Step-by-step Tutorial: Data Streaming from Twitter to HDFS**

Step 1: Open a Twitter account

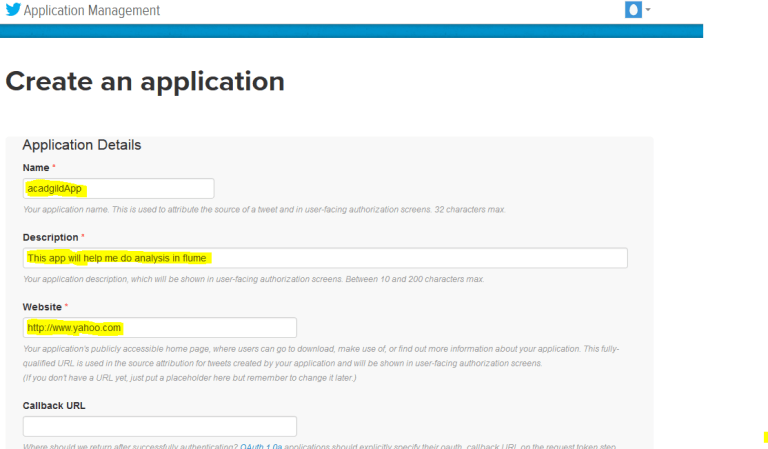


Step 2: Go to the following link and click on ‘create app’.

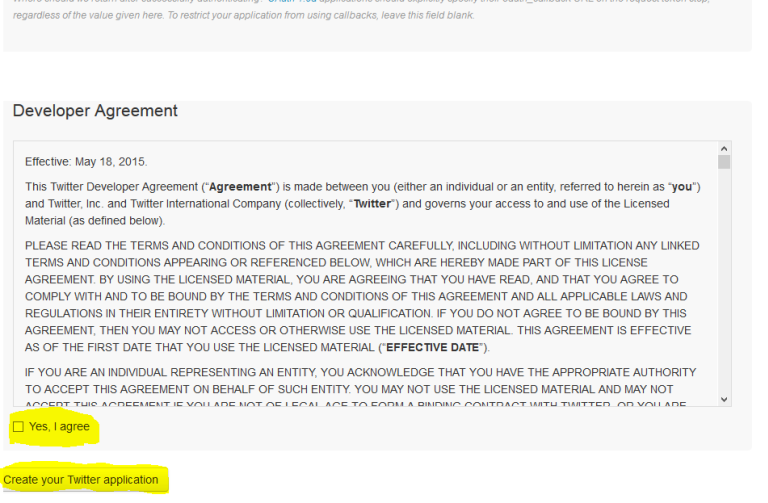
<https://apps.twitter.com/app>



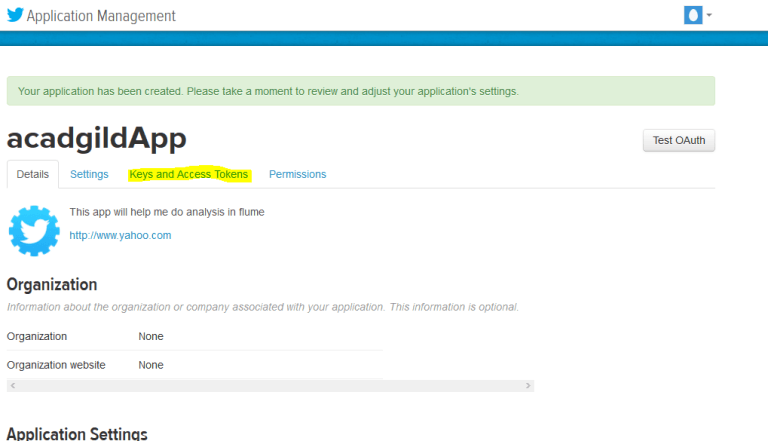
Step 3: Fill in the necessary details.



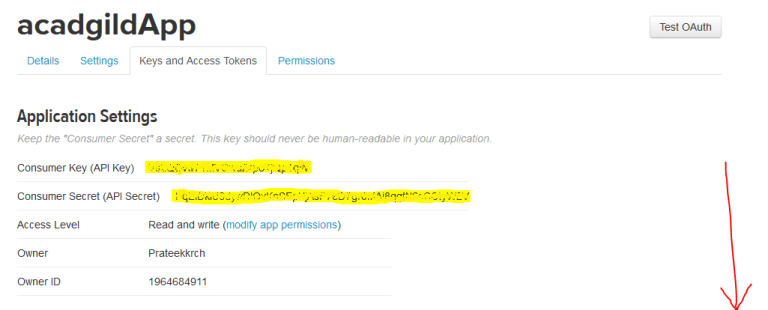
Step 4: Accept the agreement and click on ‘create your Twitter application’.



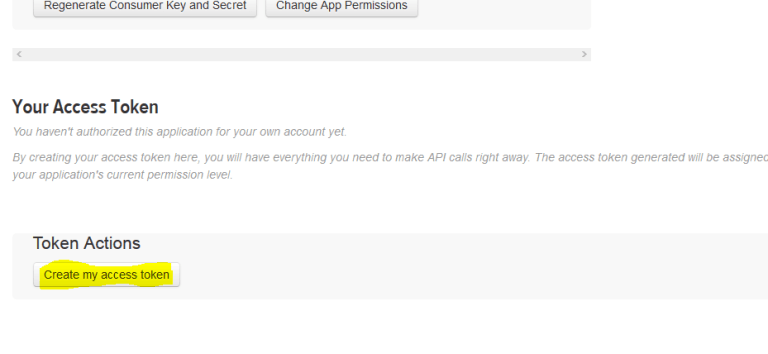
Step 5: Go to ‘Keys and Access Token’ tab.



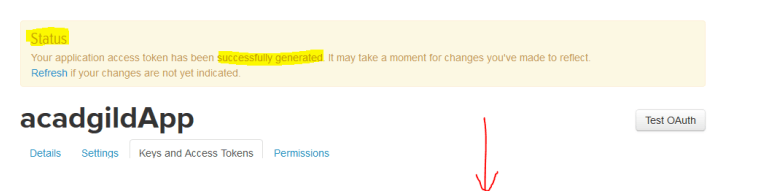
Step 6: Copy the consumer key and the consumer secret.



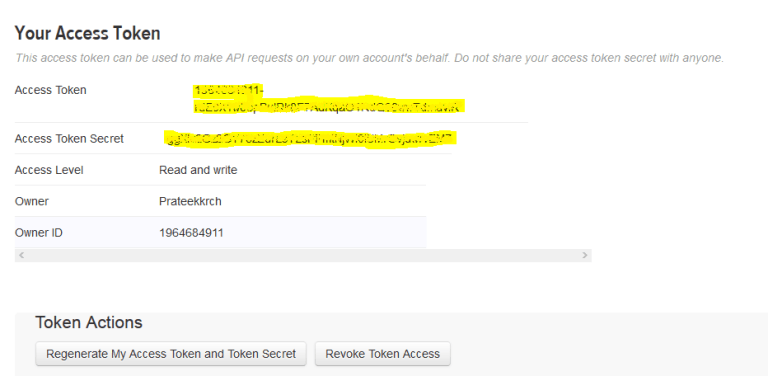
Step 7: Scroll down further and click on ‘create my access token’.



You will now receive a message that says that you have successfully generated your application access token.



Step 8: Copy the Access Token and Access token Secret.



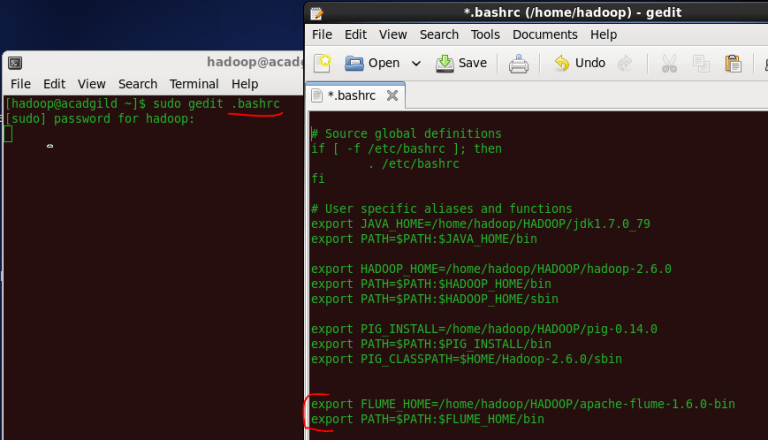


Step 9: Download flume tar file from below link and extract it .

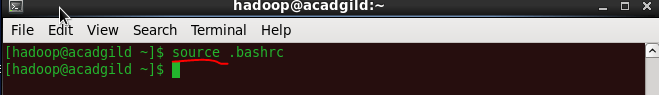
<https://drive.google.com/drive/u/0/folders/0B1QaXx7tpw3SWkMwVFBkc3djNFk>

Extract the flume.tar file and update the path of extracted file in .bashrc

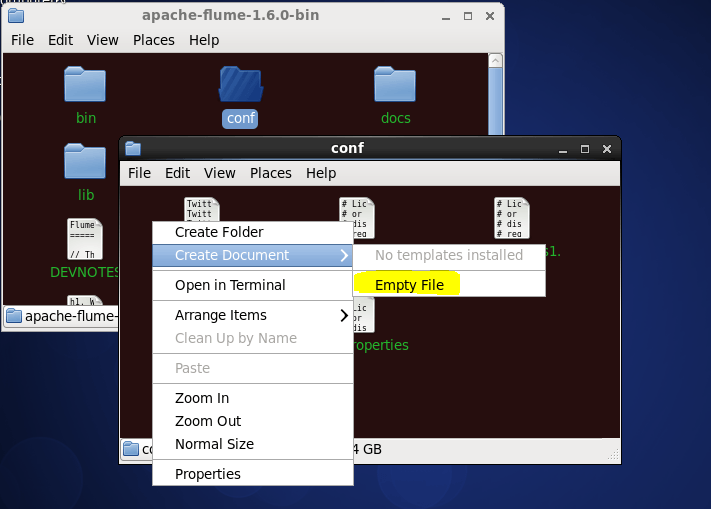
NOTE: keep the path same as where the extracted file exists.



Update the bashrc file with source command.



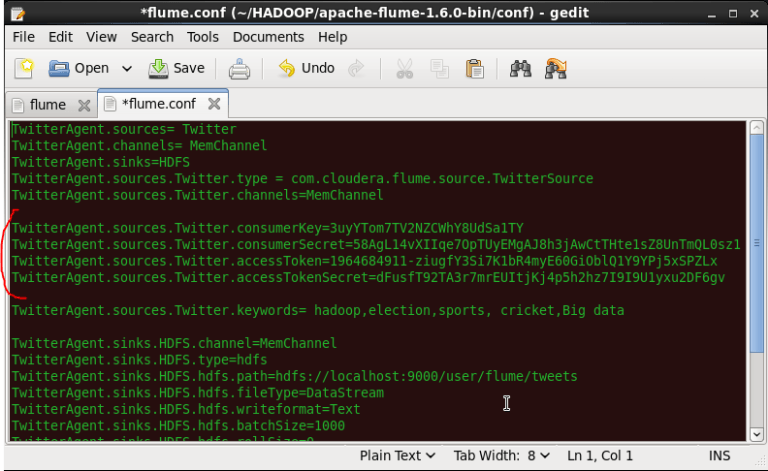
Step 10: Create a new file inside the ‘conf’ directory inside the Flume-extracted directory.



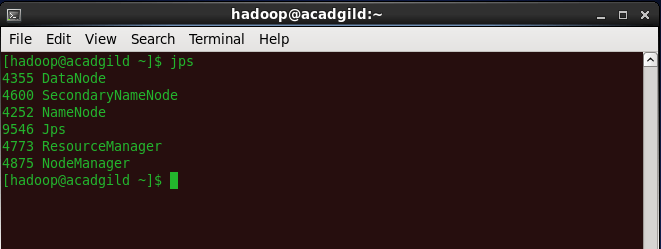
Step 11: Copy the information from the below link and paste it inside the newly created file.

<https://github.com/prateekATacadgild/Session-12_Project-1_Flume-Files/blob/master/acadgild.conf>

Step 12: Change the twitter api keys with the keys generated as shown in the step no 6 and step number 8.

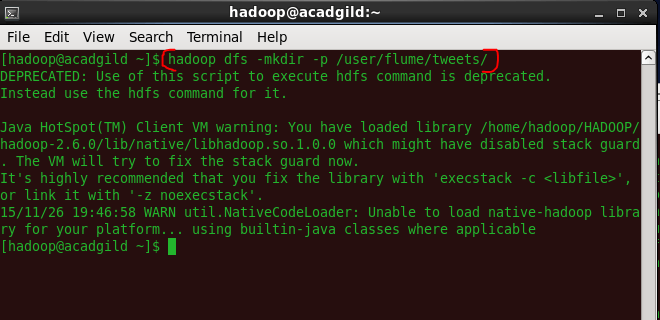


Step 13: Open the terminal to check for all Hadoop daemons running, by using the ‘jps’ command.



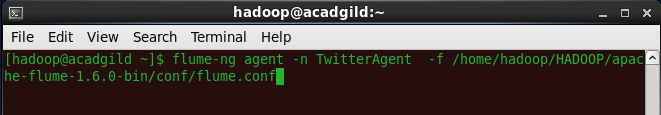
Step 14: Using the below command, create a directory inside HDFS where Twitter data will be stored.

Hadoop dfs –mkdir –p /user/flume/tweets

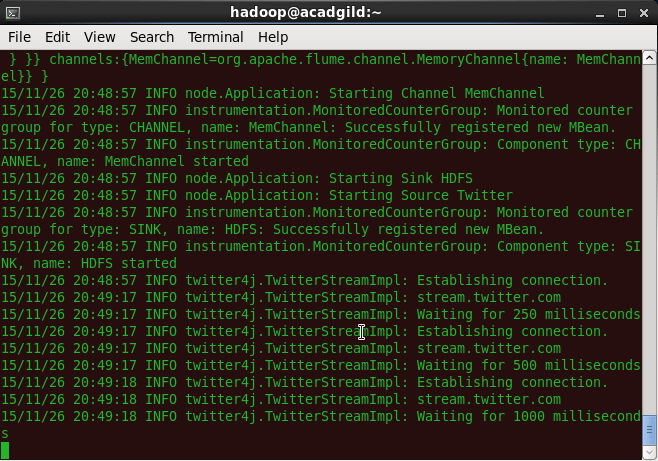


Step 15: For fetching data from Twitter, give the below command in the terminal.

flume-ng agent -n TwitterAgent -f <location of created/edited conf file>



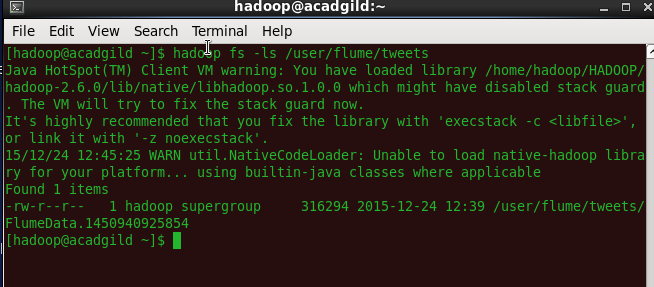
This will start fetching data from Twitter and send it to the HDFS.



To stop fetching data, press *‘Ctrl+c’*. This will end the process of fetching the data.

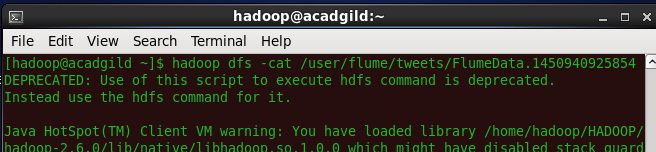
Step 16: To check the contents of the Tweets folder, use the following command:

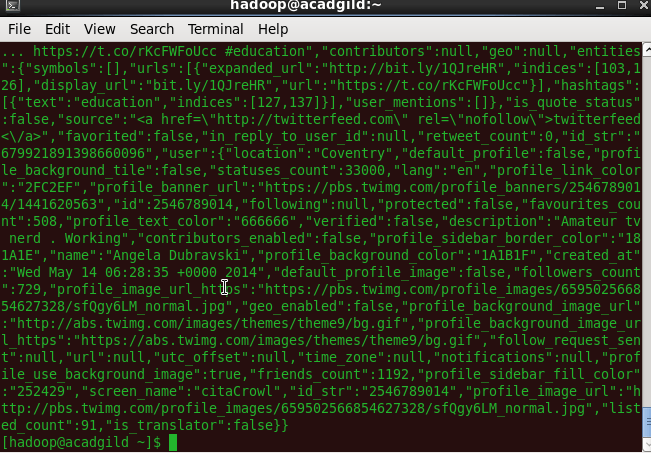
hadoop dfs –ls /user/flume/tweets



Step 17: To see the data inside this file, type the following command:

hadoop dfs –cat /us er/flume/tweets/<flumeData file name>





We have completed the action of fetching live-streaming data from Twitter and loaded it to the HDFS, using Flume.

# **3. Output**

N/A