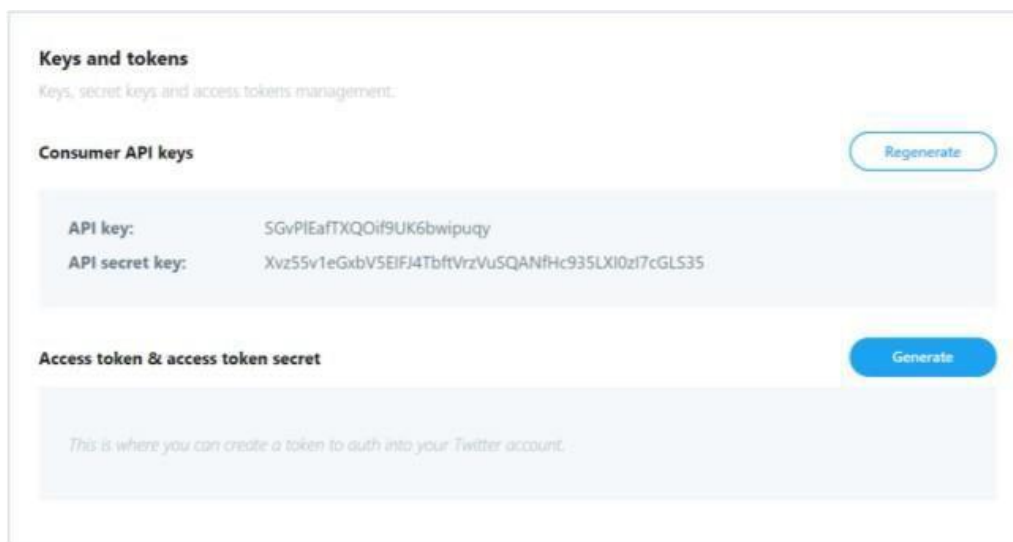


Apache Flume

Apache Flume is an open-source software that helps to store the streaming data into HDFS. A flume agent should be created through which we can stream the data. The following steps describe the process to collect the twitter data.

Process to collect Twitter Data:

1. Create an account in twitter and login with the credentials.
2. Navigate to <https://apps.twitter.com/> and create a new app.
3. Give the Name and the Description on what are you the data, and the website URL is <https://twitter.com/>
4. After the developer account is approved, create an app and get the consumer secret, consumer token, access token and access token secret for your application. To get the consumer secret, consumer token:
(a) Go to Keys and Access Tokens get the consumer key and consumer secret, and then generate Your Access tokens and get the access tokens and access token secret.



The screenshot shows the 'Keys and tokens' section of the Twitter Developer portal. It includes a 'Consumer API keys' section with a 'Regenerate' button and a table displaying the 'API key' (SGvPIEafTXQOif9UK6bwipuzy) and 'API secret key' (Xvz55v1eGxbV5EiFJ4TbftVrzVuSQANfHc935LXI0zl7cGLS35). Below this is an 'Access token & access token secret' section with a 'Generate' button and a placeholder text: 'This is where you can create a token to auth into your Twitter account.'

There are 3 components for a twitter agent namely source, sink and channel.

The flume source connects to Twitter API and receives data in JSON format which in turn stored into HDFS.

Now, create a configuration file for the flume agent by specifying the consumer key, consumer secret, access token, access token secret, keywords and HDFS path.

A sample configuration file with file extension .conf is shown below. It shows all the keys and keywords to be used to collect the twitter data.

LOGGING TO CSX

- Refer <http://computerscience.okstate.edu/computing/csx-logging-on>

Use “hadoop-nn001.cs.okstate.edu” to login into putty. It is a new cluster

GUIDELINES TO STREAM TWITTER DATA USING FLUME

1. Create an account in twitter and login with the credentials.
 - Navigate to <https://apps.twitter.com/> and create a new app.
 - Give the Name and the Description on what are you the data, and the website URL is <https://twitter.com/>
 - Get the consumer secret, consumer token, access token and access token secret for your application. To get the consumer secret, consumer token: Go to Keys and Access Tokens get the consumer key and consumer secret, and then go to Your Access tokens and get the access tokens and access token secret.

2. Create a path in hdfs where data should be collected (same path should be given in conf file)

Command: `hdfs dfs -mkdir /user/your_shortid/file_name/`

Example: `hdfs dfs -mkdir /user/sisheri/JoeBiden_data/`

3. Now, create a configuration file for the flume agent by specifying the consumer key, consumer secret, access token, access token secret, keywords and HDFS path in your local.

Command: `nano file_name.conf`

Example: `nano biden.conf`

Sample Config file:

```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS
TwitterAgent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sources.Twitter.consumerKey = ****
TwitterAgent.sources.Twitter.consumerSecret = ****
TwitterAgent.sources.Twitter.accessToken = ****
```

```
TwitterAgent.sources.Twitter.accessTokenSecret = *****
TwitterAgent.sources.Twitter.keywords = covid
TwitterAgent.sinks.HDFS.channel = MemChannel
TwitterAgent.sinks.HDFS.type = hdfs
TwitterAgent.sinks.HDFS.hdfs.path = hdfs://hadoop-
nn001.cs.okstate.edu:9000/user/sapalad/sample3/%Y/%m/%d/%H
TwitterAgent.sinks.HDFS.hdfs.useLocalTimeStamp = true
TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream
TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text
TwitterAgent.sinks.HDFS.hdfs.batchSize = 100
TwitterAgent.sinks.HDFS.hdfs.rollSize = 0
```

```
TwitterAgent.sinks.HDFS.hdfs.rollCount = 0
TwitterAgent.channels.MemChannel.type = memory
TwitterAgent.channels.MemChannel.capacity = 10000
TwitterAgent.channels.MemChannel.transactionCapacity = 10000
```

Giving '&' at the end will make sure the data collection process runs continuously in the background. nohup.out file is the log file that will be created as we start the process. The data collected will be in JSON format.

4. Check the data in hdfs path created in step-2

Note: Sometimes when downloading a lot of data, the flume agent stops collecting data.

It is recommended to keep checking if data is downloading.

Command to check whether file is downloading or not:-

hdfs dfs -ls /user/userid/(folder name along with date: the format you are using for downloading which is mentioned in the .conf file)

Command to check the count of files.

hdfs dfs -count /username/(folder name)

There is a limit on the number of files. No more data is put into files when the limit is reached. This will also stop data being downloaded.

hdfs dfs -du -s -h /user/csx_username/filename/

Example:

hdfs dfs -du -s -h /user/sapalad/sample/

This command is used to check the size of the files downloaded

Note: -

Please be aware of how many process you are running, always kill the process when your done with downloading the data

Some of the useful commands to check the process which are running in cluster

\$ pgrep -c -u csx_username -f flume

Example: pgrep -c -u sapalad -f flume

This command will give a count of all the processes csx_username is running with flume somewhere in the command line

\$kill -KILL -u csx_username -f flume

Send a kill signal to ALL flume process owned by csx_username , only use this when you want to kill all the remaining process in the cluster