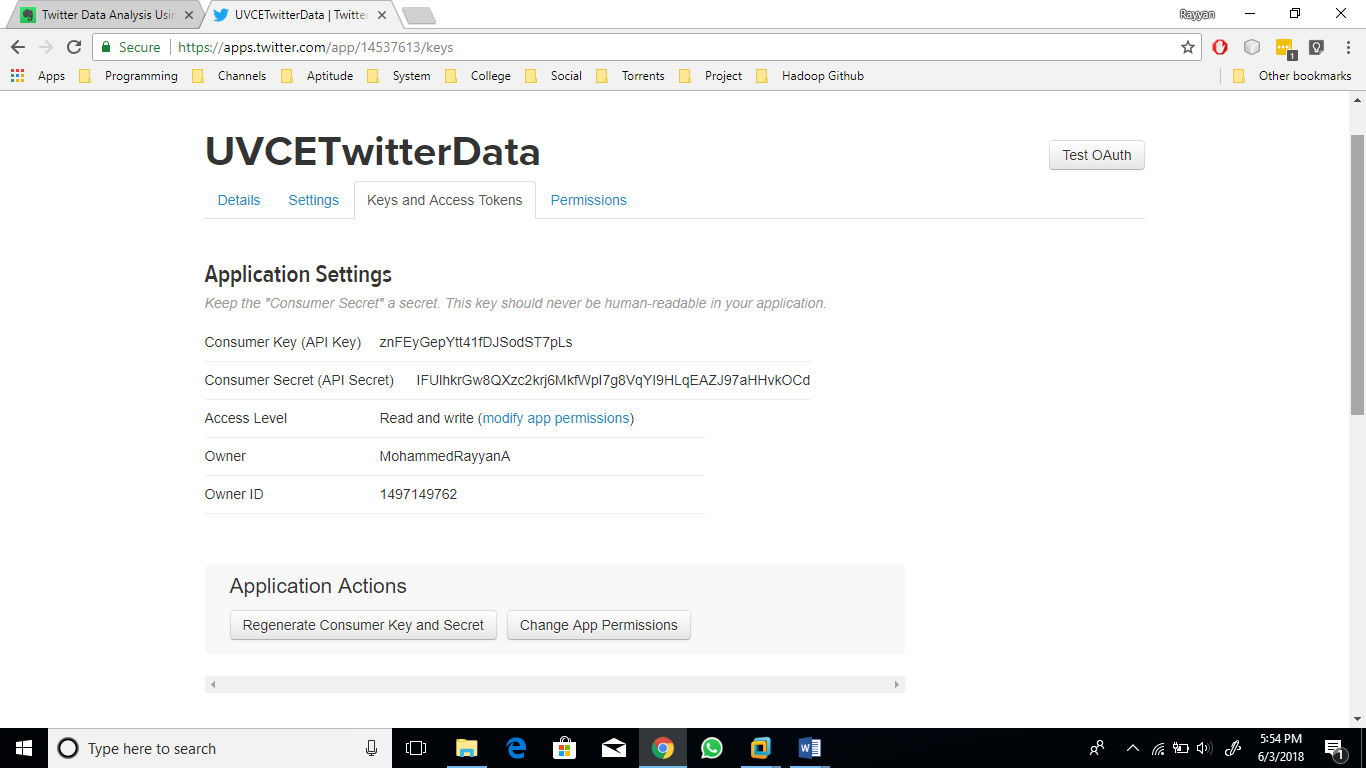
**Election Data Analysis Using Hadoop Framework**

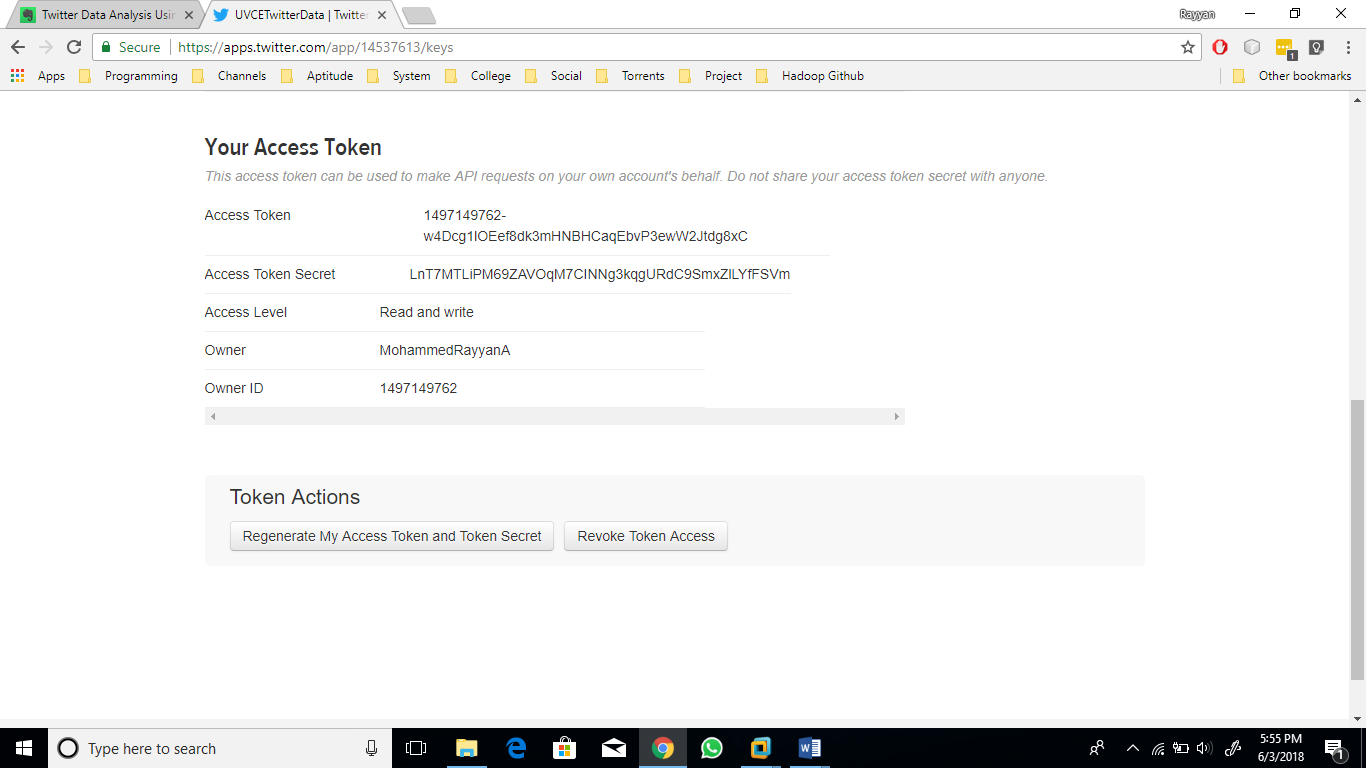
# Creation of Twitter Account

Twitter Developer Account can be created at [Twitter Developers apps Page](https://dev.twitter.com/apps/). In this page we need to provide valid twitter account page in the website field from which we need to get streaming data. If we provide valid details on this page we will get our app created as shown in below screen shots. For security reasons, I have blurred consumer access key, secret values in the below screens.

We need below four values for authenticated by Twitter.

* **Consumer Key (API Key):** znFEyGepYtt41fDJSodST7pLs
* **Consumer Secret (API Secret):** IFUlhkrGw8QXzc2krj6MkfWpI7g8VqYI9HLqEAZJ97aHHvkOCd
* **Access Token :** 1497149762-w4Dcg1lOEef8dk3mHNBHCaqEbvP3ewW2Jtdg8xC
* **Access Token Secret** : LnT7MTLiPM69ZAVOqM7CINNg3kqgURdC9SmxZlLYfFSVm





# Installing Open JDK

**>>sudo apt-get update**

**>>sudo apt-get install openjdk-8-jdk**

**>>java -version**

openjdk version "1.8.0\_72-internal"

OpenJDK Runtime Environment (build 1.8.0\_72-internal-b05)

OpenJDK 64-Bit Server VM (build 25.72-b05, mixed mode)

# Installing Hadoop

## Disable IPv6

**>>sudo apt-get install vim**

**>>sudo gedit /etc/sysctl.conf**

    # disable ipv6

    net.ipv6.conf.all.disable\_ipv6 = 1

    net.ipv6.conf.default.disable\_ipv6 = 1

    net.ipv6.conf.lo.disable\_ipv6 = 1

**>>cat /proc/sys/net/ipv6/conf/all/disable\_ipv6** ... (should return zero)

## Adding a dedicated Hadoop User

**>>sudo addgroup hadoop**

**>>sudo adduser --ingroup hadoop hduser**

## Install SSH

**>>sudo apt-get install ssh**

## Give hduser Sudo Permission

**>>sudo adduser hduser sudo**

## Setup SSH Certificates

**>>su hduser**

**>>ssh-keygen -t rsa -P ""**

**>>cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys**

**>>ssh localhost**

## Download Hadoop

**>>su hduser**

**>>sudo  wget** <http://www-us.apache.org/dist/hadoop/common/hadoop-2.9.0/hadoop-2.9.0.tar.gz>

**>>tar xvzf hadoop-2.9.0.tar.gz**

**>>cd hadoop-2.9.0**

**>>sudo mkdir /usr/local/hadoop**

**>>sudo mv \* /usr/local/hadoop**

## Set up the Configuration files

**>> sudo gedit ~/.bashrc**

        #HADOOP VARIABLES START

        export JAVA\_HOME=/usr/lib/java/jdk-9.0.4

        export HADOOP\_INSTALL=/usr/local/hadoop

        export PATH=$PATH:$HADOOP\_INSTALL/bin

        export PATH=$PATH:$HADOOP\_INSTALL/sbin

        export HADOOP\_MAPRED\_HOME=$HADOOP\_INSTALL

        export HADOOP\_COMMON\_HOME=$HADOOP\_INSTALL

        export HADOOP\_HDFS\_HOME=$HADOOP\_INSTALL

        export YARN\_HOME=$HADOOP\_INSTALL

        export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_INSTALL/lib/native

        export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_INSTALL/lib"

        #HADOOP VARIABLES END

**>> Sudo gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh**

        export JAVA\_HOME=/usr/lib/java/jdk-9.0.4

**>>sudo mkdir -p /app/hadoop/tmp**

**>>sudo chown hduser:hadoop /app/hadoop/tmp**

**>>sudo gedit /usr/local/hadoop/etc/hadoop/core-site.xml**

        <property>

              <name>hadoop.tmp.dir</name>

                <value>/app/hadoop/tmp</value>

                    <description>A base for other temporary directories.</description>

               </property>

        <property>

              <name>fs.default.name</name>

              <value>[hdfs://localhost:54310](hdfs://localhost:54310/)</value>

              <description>The name of the default file system.  A URI whose

              scheme and authority determine the FileSystem implementation.  The

              uri's scheme determines the config property (fs.SCHEME.impl) naming

              the FileSystem implementation class.  The uri's authority is used to

              determine the host, port, etc. for a filesystem.</description>

        </property>

**>>cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml**

**>>sudo gedit /usr/local/hadoop/etc/hadoop/mapred-site.xml**

            <property>

              <name>mapred.job.tracker</name>

              <value>localhost:54311</value>

              <description>The host and port that the MapReduce job tracker runs

              at.  If "local", then jobs are run in-process as a single map

              and reduce task.

              </description>

        </property>

**>>sudo mkdir -p /usr/local/hadoop\_store/hdfs/namenode**

**>>sudo mkdir -p /usr/local/hadoop\_store/hdfs/datanode**

**>>sudo chown -R hduser:hadoop /usr/local/hadoop\_store**

**>>sudo gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml**

           <property>

              <name>dfs.replication</name>

              <value>1</value>

              <description>Default block replication.

              The actual number of replications can be specified when the file is created.

              The default is used if replication is not specified in create time.

              </description>

        </property>

        <property>

               <name>dfs.namenode.name.dir</name>

               <value>file:/usr/local/hadoop\_store/hdfs/namenode</value>

        </property>

        <property>

               <name>dfs.datanode.data.dir</name>

               <value>file:/usr/local/hadoop\_store/hdfs/datanode</value>

        </property>

**>>sudo chmod 777 -R /usr/local/hadoop/**

## Format Hadoop filesystem

   >> **hadoop namenode -format**

Close all Terminal

## Starting Hadoop

**>>su hduser**

**>>hadoop fs -mkdir /user/hduser/flume**

**>>hadoop fs -chown hduser:hadoop /user/hduser  
>>sudo chown -R hduser:hadoop /usr/local/hadoop/**

**>>cd /usr/local/hadoop/sbin**

**>>start-dfs.sh**

**>>start-yarn.sh**

## Testing if it is working

**>>jps**

19889 DataNode  
20705 Jps  
19779 NameNode  
20074 SecondaryNameNode  
20270 ResourceManager  
20382 NodeManager

(Or)

**>>netstat -plten | grep java**

Web Browser: [http://localhost:9000/](http://localhost:50070/)

## Stopping Hadoop

**>>stop-dfs.sh**

stop-yarn.sh

# Flume Installation

## Download flume:

>>**sudo wget** <http://www-eu.apache.org/dist/flume/1.8.0/apache-flume-1.8.0-bin.tar.gz>

## Unzip flume tar file

**>>tar -xvzf apache-flume-1.8.0-bin.tar.gz**

**>>cd /usr/local**

**>>sudo mkdir flume**

**>>sudo mv \* /usr/local/flume**

**>>sudo gedit ~/.bashrc**

#flume variables

export FLUME\_HOME=/usr/local/flume

PATH=$PATH:$FLUME\_HOME/bin

export CLASSPATH=$CLASSPATH:$FLUME\_HOME/lib/\*

## Download jar file

flume-source-1.0-SNAPSHOOT.jar

**>>sudo mv flume-sources-1.0-SNAPSHOT.jar /usr/local/flume/lib**

## In flume/conf

**>>sudo cp flume-env.sh.template flume-env.sh**

## Modify flume-env.sh

**>>sudo vim flume-env.sh**

export JAVA\_HOME=/usr/lib/java/jdk1.8.0\_161

FLUME\_CLASSPATH="/usr/local/flume/lib/flume-sources-1.0-SNAPSHOT.jar"

## Create flume-twitter.conf file in conf folder and paste given lines

**>>sudo gedit flume-twitter.conf**

## Twitter:

### Keys and Access Tokens:

Consumer Key (API Key): znFEyGepYtt41fDJSodST7pLs

Consumer Secret (API Secret): IFUlhkrGw8QXzc2krj6MkfWpI7g8VqYI9HLqEAZJ97aHHvkOCd

Access Token 1497149762-w4Dcg1lOEef8dk3mHNBHCaqEbvP3ewW2Jtdg8xC

Access Token Secret LnT7MTLiPM69ZAVOqM7CINNg3kqgURdC9SmxZlLYfFSVm

//Conf code

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 = znFEyGepYtt41fDJSodST7pLs

TwitterAgent.sources.Twitter.consumerSecret = IFUlhkrGw8QXzc2krj6MkfWpI7g8VqYI9HLqEAZJ97aHHvkOCd

TwitterAgent.sources.Twitter.accessToken = 1497149762-w4Dcg1lOEef8dk3mHNBHCaqEbvP3ewW2Jtdg8xC

TwitterAgent.sources.Twitter.accessTokenSecret = LnT7MTLiPM69ZAVOqM7CINNg3kqgURdC9SmxZlLYfFSVm

TwitterAgent.sources.Twitter.keywords = spark, flink

TwitterAgent.sinks.HDFS.type = hdfs

TwitterAgent.sinks.HDFS.hdfs.path = <hdfs://localhost:9000/user/Hadoop/twitter_data/>

TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream

TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text

TwitterAgent.sinks.HDFS.hdfs.batchSize = 1000

TwitterAgent.sinks.HDFS.hdfs.rollSize = 0

TwitterAgent.sinks.HDFS.hdfs.rollCount = 10000

# Describing/Configuring the channel TwitterAgent.channels.MemChannel.type = memory

TwitterAgent.channels.MemChannel.capacity = 10000

TwitterAgent.channels.MemChannel.transactionCapacity = 100

## Binding the source and sink to the channel

TwitterAgent.sources.Twitter.channels = MemChannel

TwitterAgent.sinks.HDFS.channel = MemChannel

## To load enter

**>>bin/flume-ng agent -n TwitterAgent --conf ./conf/ -f conf/flume-twitter.conf -Dflume.root.logger=DEBUG,console**

**>> cd $FLUME\_HOME**

**>>bin/flume-ng agent --conf ./conf/ -f conf/twitter.conf**

**Dflume.root.logger=DEBUG,console -n TwitterAgent**

# Hive Installation

## Download and Extract

The downloaded Hive tar file needs to be extracted using the tar command with –xvf option as shown below -

**>>tar –xvf apache-hive-2.1.0-bin.tar.gz**

## Setting Home Dir

**>>sudo gedit ~/.baschrc**

# Set HIVE\_HOME

export HIVE\_HOME=”$HOME/hive/ apache-hive-2.1.0-bin”

PATH=$PATH:$HIVE\_HOME/bin

Export $PATH

**>>source .bashrc**

## Set HADOOP\_HOME in hive-config.sh

**>>sudo cd /usr/local/hive/conf**

**>>sudo gedit hive-config.sh**

export HADOOP\_HOME=/opt/hadoop

## Create a directory for the hive warehouse into hdfs.

**>>hadoop dfs –mkdir –p /user/hive/warehouse**

Modify the permissions

**>>hadoop dfs –chmod 765 /user/hive/warehouse**

**>>cd $HIVE\_HOME/bin/**

**>>schematool –initschema –dbtype derby**