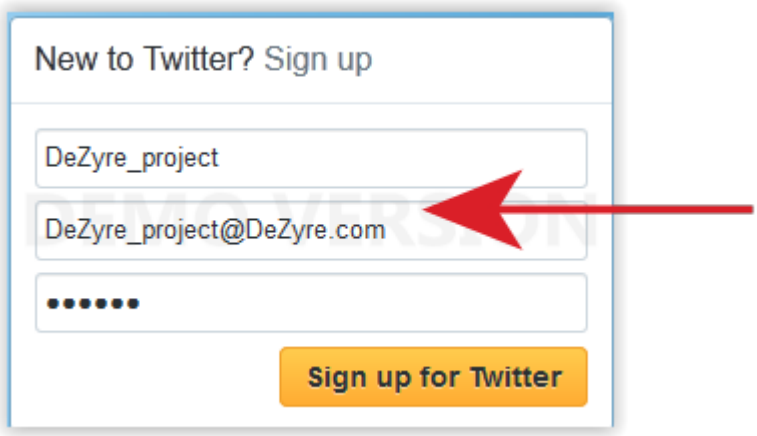
# Analyzing Twitter Data with Apache Hadoop

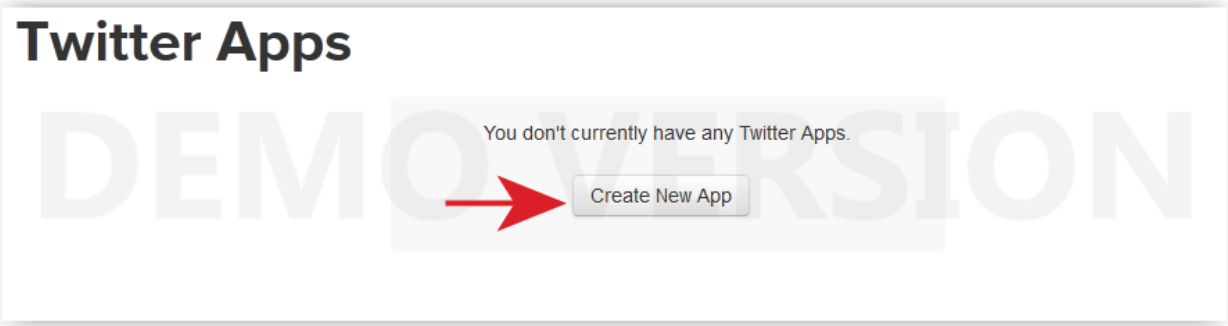
**Create twitter access token:**

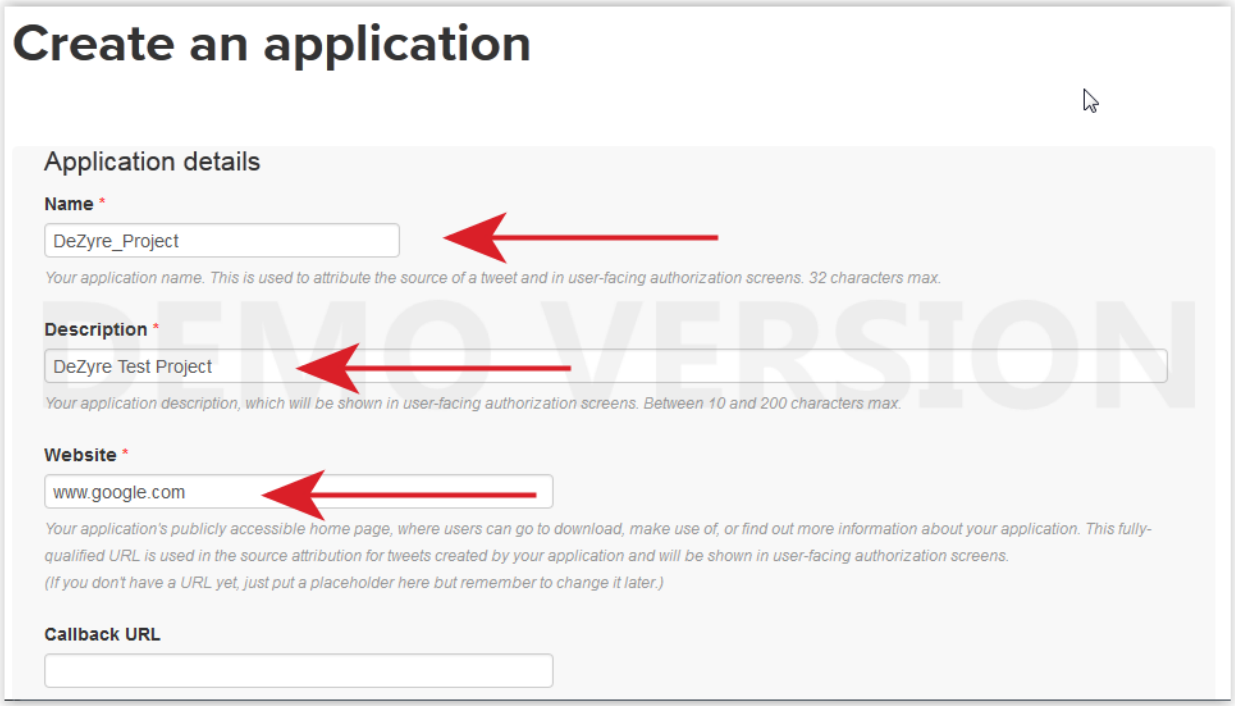
1. Create twitter account: Go to twitter.com and create a twitter account as shown below.



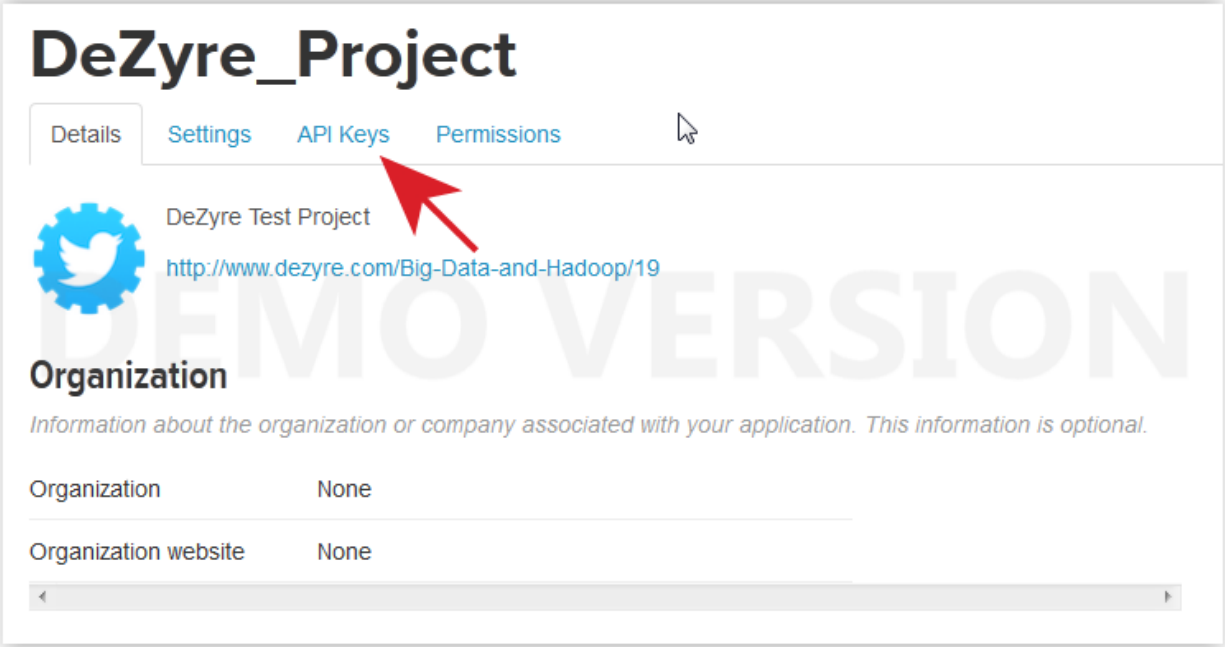

2. Verify email for your twitter account.

3. Go to apps.twitter.com and click on "Create App" as shown below in screenshots.

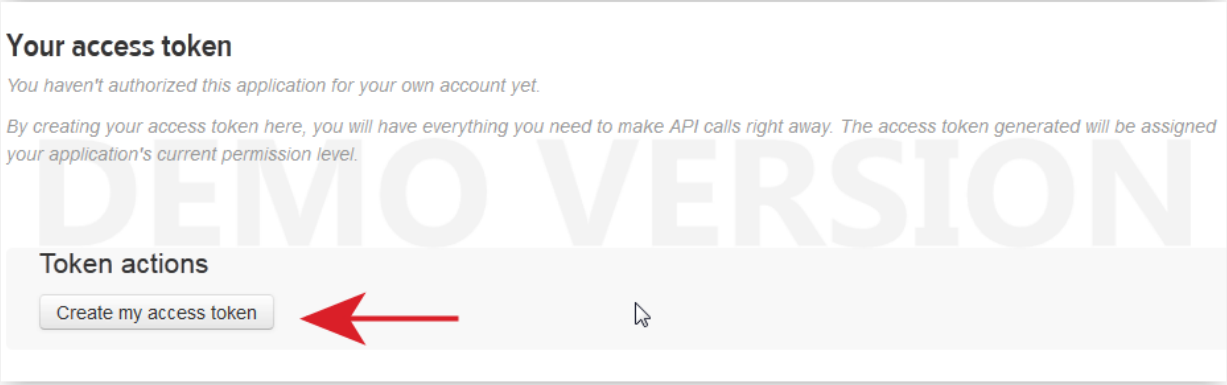




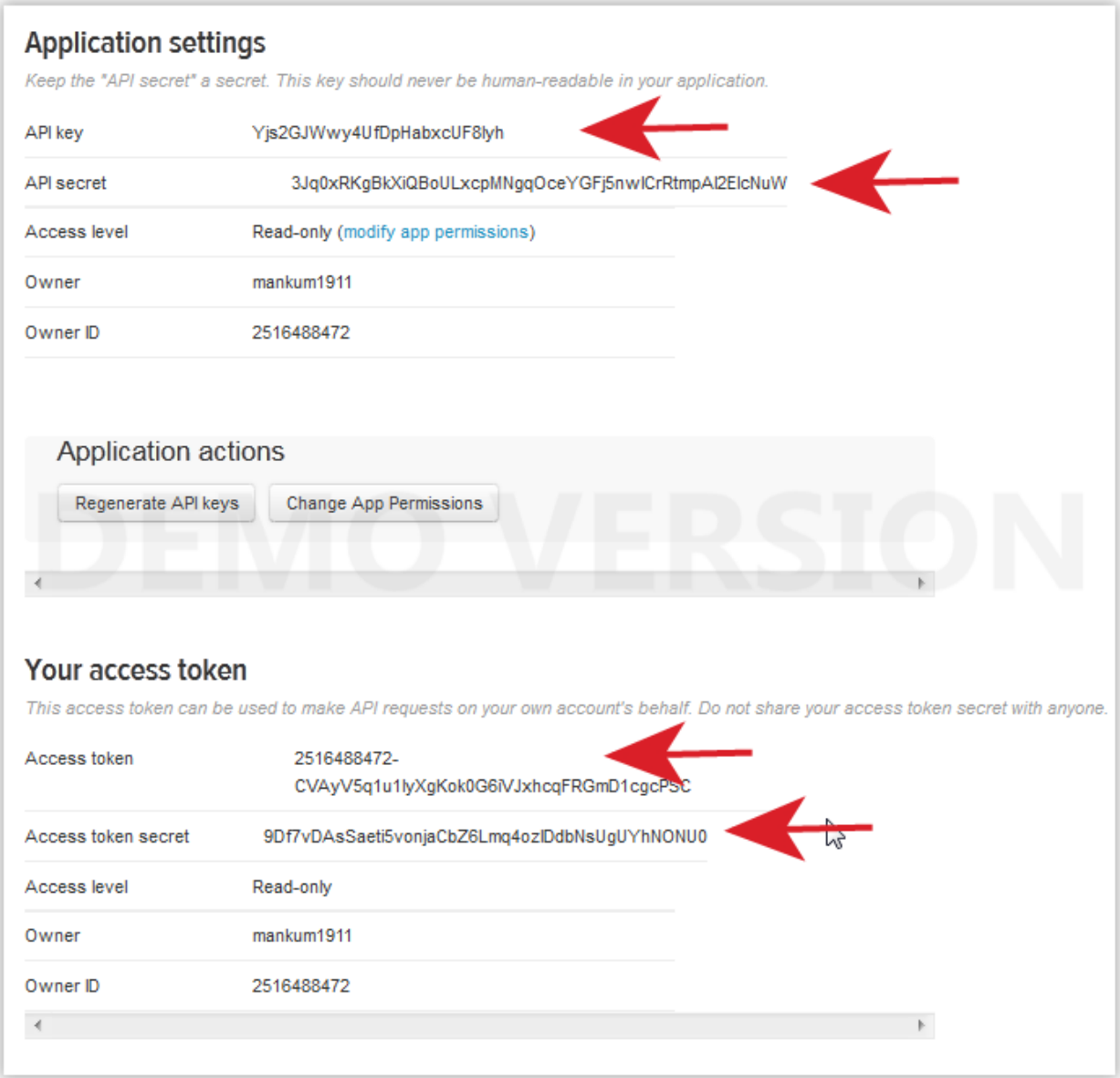

4. Once you have created an application, click on “API Keys” section.



5. Click on “Create My Access Token” button in the “Your access token” section.

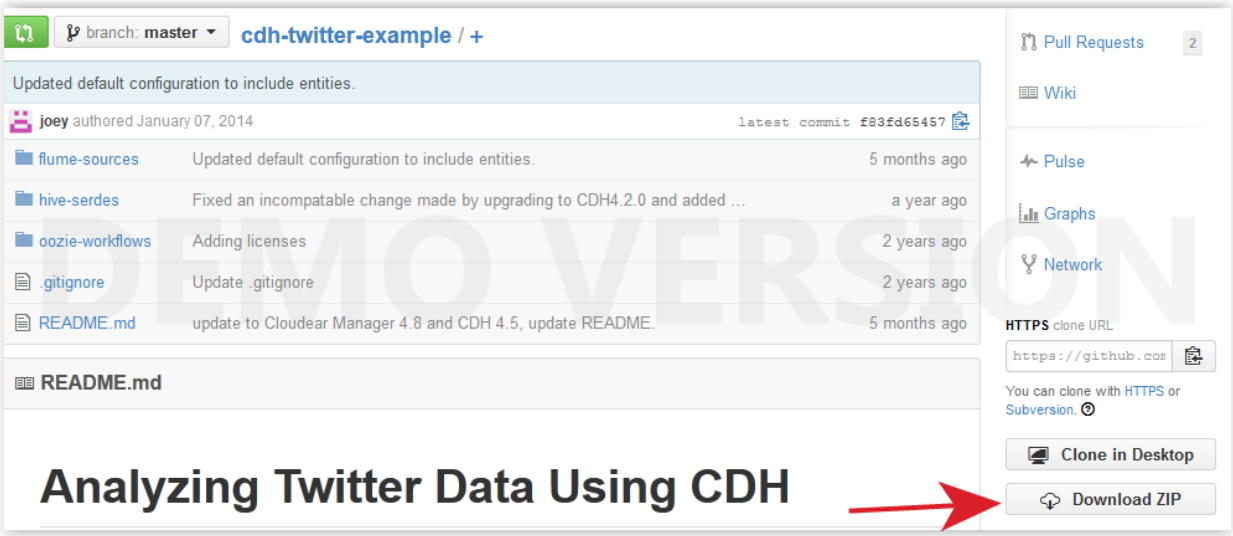


6. Your access token will be generated in some time which you would need to use in flume.conf file for flume configuration in next few steps.



**Setting up flume agent:**

1. Go to this link and download code from here (https://github.com/cloudera/cdh-twitter-example) **OR** else you can use the jar which is shared under twitter-project.



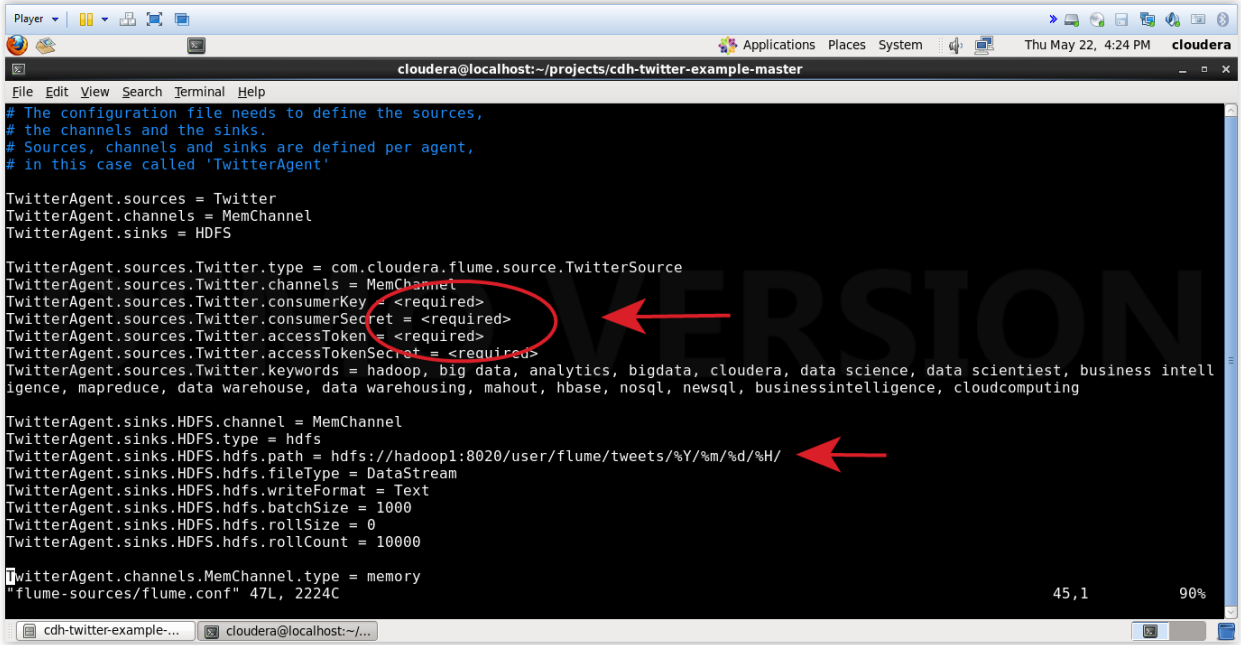

2. Extract the jar and run following command for building flume source and copy to flume library. ***Its not needed if you are using flume-sources-jar shared in twitter-project folder.***

```
$ cd flume-sources/

$ mvn clean install –DskipTests

$ sudo cp /flume-sources-1.0-SNAPSHOT.jar /usr/lib/flume-ng/lib/

$ sudo chmod +r  /usr/lib/flume-ng/lib/flume-sources-1.0-SNAPSHOT.jar
```

3. Create a HDFS directory "/user/cloudera/twitter/" where you will get the twitter inputs.

```
$ hadoop dfs -mkdir /user/cloudera/twitter/
```

4. Edit flume conf “flume-sources/flume.conf” to change twitter keys and tokens and hdfs path. Use twitter access tokens from twitter account. Change HDFS path to “/user/cloudera/twitter/%Y/%m/%d/%H/”




5. Now start flume-agent using below command.

```
$ flume-ng agent -n TwitterAgent -c conf -f flume.conf
```

6. In some time file will start coming in HDFS. Check using below command.

```
$ hadoop dfs -ls /user/cloudera/twitter
```

**Hive Setup**

1. Run following command for building SerDe library **OR** else you can use the jar which is shared under twitter-project.

```
$ cd hive-serdes/

$ mvn clean install –DskipTests
```

2. Start hive using hive command.

3. Now add jar using add jar command as shown below.

```
Hive> add jar hive-serdes-1.0-SNAPSHOT.jar;
```

4. Create a external table in hive as shown below.

```
Hive> CREATE EXTERNAL TABLE tweets_partioned (
  id BIGINT,
  created_at STRING,
  source STRING,
  favorited BOOLEAN,
  retweeted_status STRUCT<
    text:STRING,
    user:STRUCT<screen_name:STRING,name:STRING>,
    retweet_count:INT>,
  entities STRUCT<
    urls:ARRAY<STRUCT<expanded_url:STRING>>,
    user_mentions:ARRAY<STRUCT<screen_name:STRING,name:STRING>>,
    hashtags:ARRAY<STRUCT<text:STRING>>>,
  text STRING,
  user STRUCT<
    screen_name:STRING,
    name:STRING,
    friends_count:INT,
    followers_count:INT,
    statuses_count:INT,
    verified:BOOLEAN,
    utc_offset:INT,
    time_zone:STRING>,
  in_reply_to_screen_name STRING
)
PARTITIONED BY (year INT, month INT, dt INT, dthour INT)
ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe'
LOCATION '/user/cloudera/tweets';
```

**Oozie Workflow Setup:**

**Oozzie workflow here is used to create partition based on directories.**

1. External JARs are provided to Oozie through a lib directory in the workflow directory. The workflow will need a copy hive-serdes JAR. Run following command :

```
$ mkdir -p oozie-workflows/lib

$ cp hive-serdes-1.0-SNAPSHOT.jar twitter-project/oozie-workflows/lib/

$  cp /var/lib/oozie/mysql-connector-java.jar oozie-workflows/lib
```

2. Copy hive-site.xml to the oozie-workflows directory

```
$ cp /etc/hive/conf/hive-site.xml oozie-workflows/

$ sudo chown oozie:oozie oozie-workflows/hive-site.xml

$ hadoop dfs -copyFromLocal  oozie-workflows /user/cloudera/

$ oozie job -oozie http://localhost:11000/oozie -config oozie-workflows/job.properties -run
```

**If you are not able to do oozie setup**

Using Alter table commands similar to below one add partition to the table.

***alter table tweets_partioned add partition (year=2014, month=05, dt=24, dthour=05);***
***alter table tweets_partioned add partition (year=2014, month=05, dt=24, dthour=08);***

***Check whether has been successfully added and it queryable or not.***

***hive> select * from tweets_partioned where year=2014 and month=05 and dt=24 and dthour=05 limit 10;***

**To find out influential celebrity or people**

Command:

```
select t.retweeted_screen_name, sum(retweets) as total_retweets, count(*) as tweet_count
from
(
select retweeted_status.user.screen_name as retweeted_screen_name,
retweeted_status.text, max(retweeted_status.retweet_count) as retweets from
tweets_partioned group by retweeted_status.user.screen_name,retweeted_status.text
)
t
group by t.retweeted_screen_name
order by total_retweets DESC, tweet_count ASC
limit 10;
```