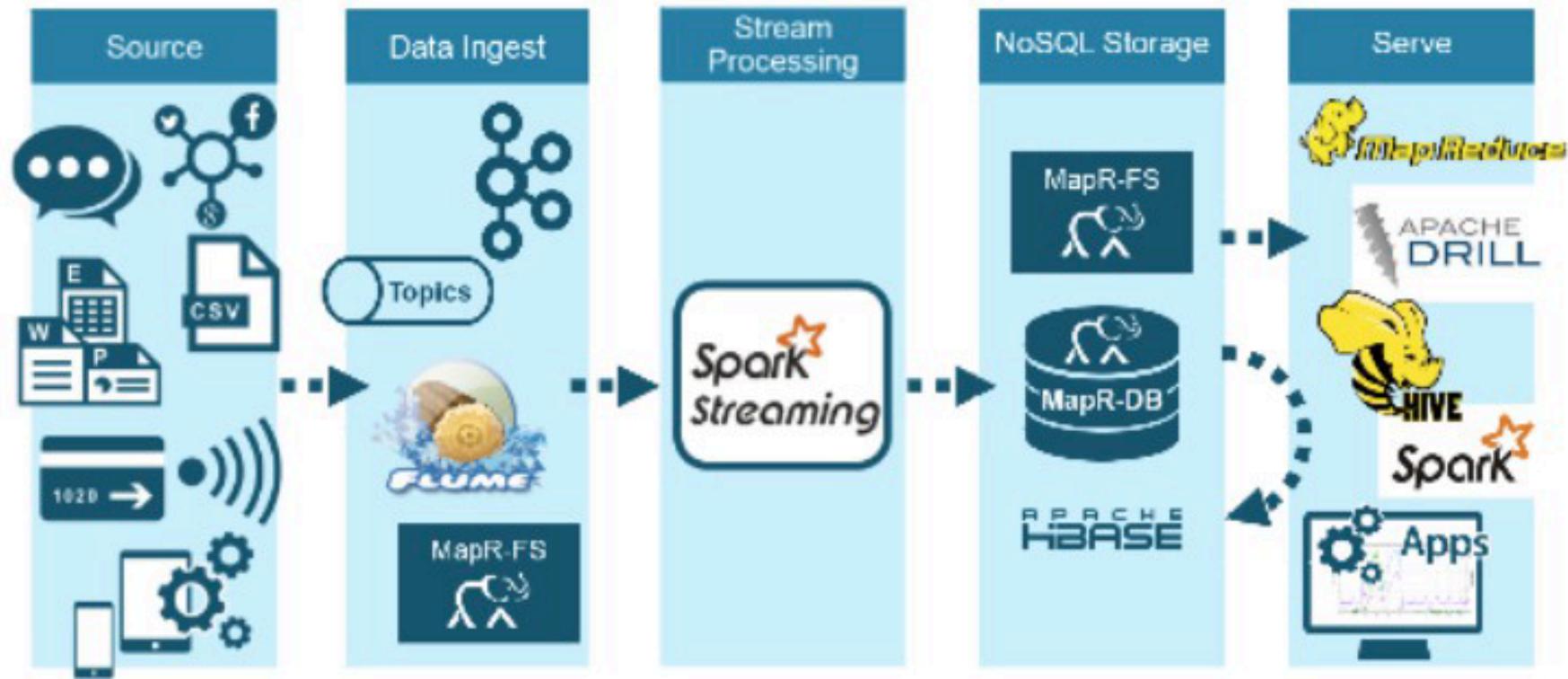


Spark Streaming

Stream Process Architecture



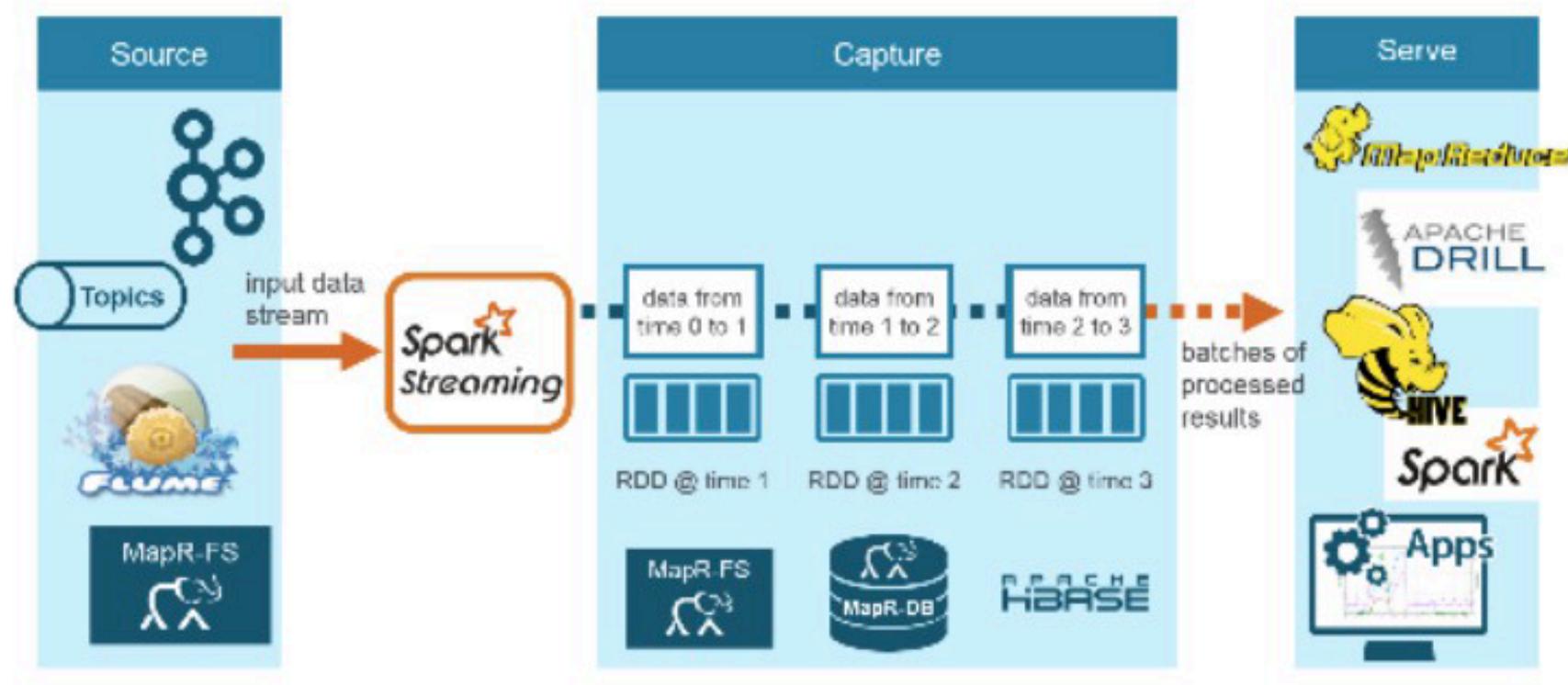
FLAGSHIP FOR LIFE



Spark Streaming Architecture



FLAGSHIP FOR LIFE

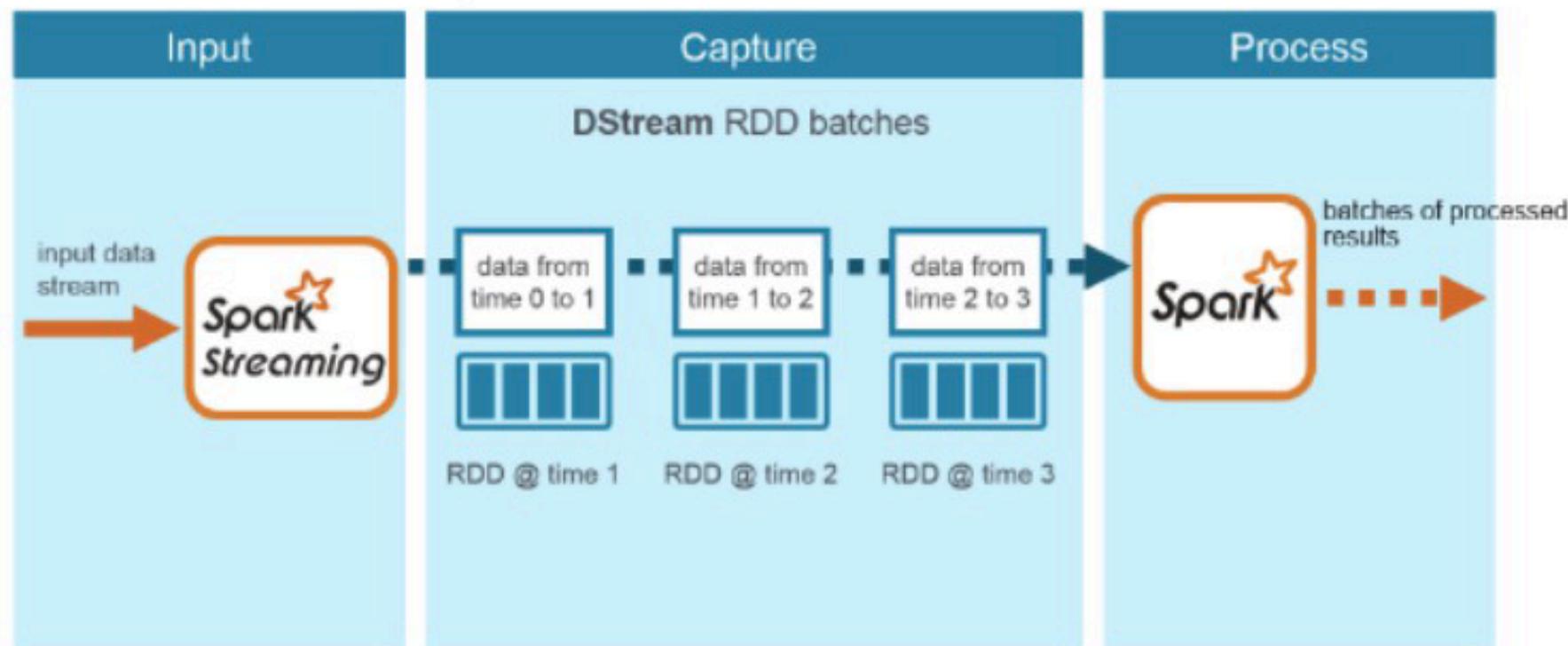


Processing Spark DStreams



FLAGSHIP FOR LIFE

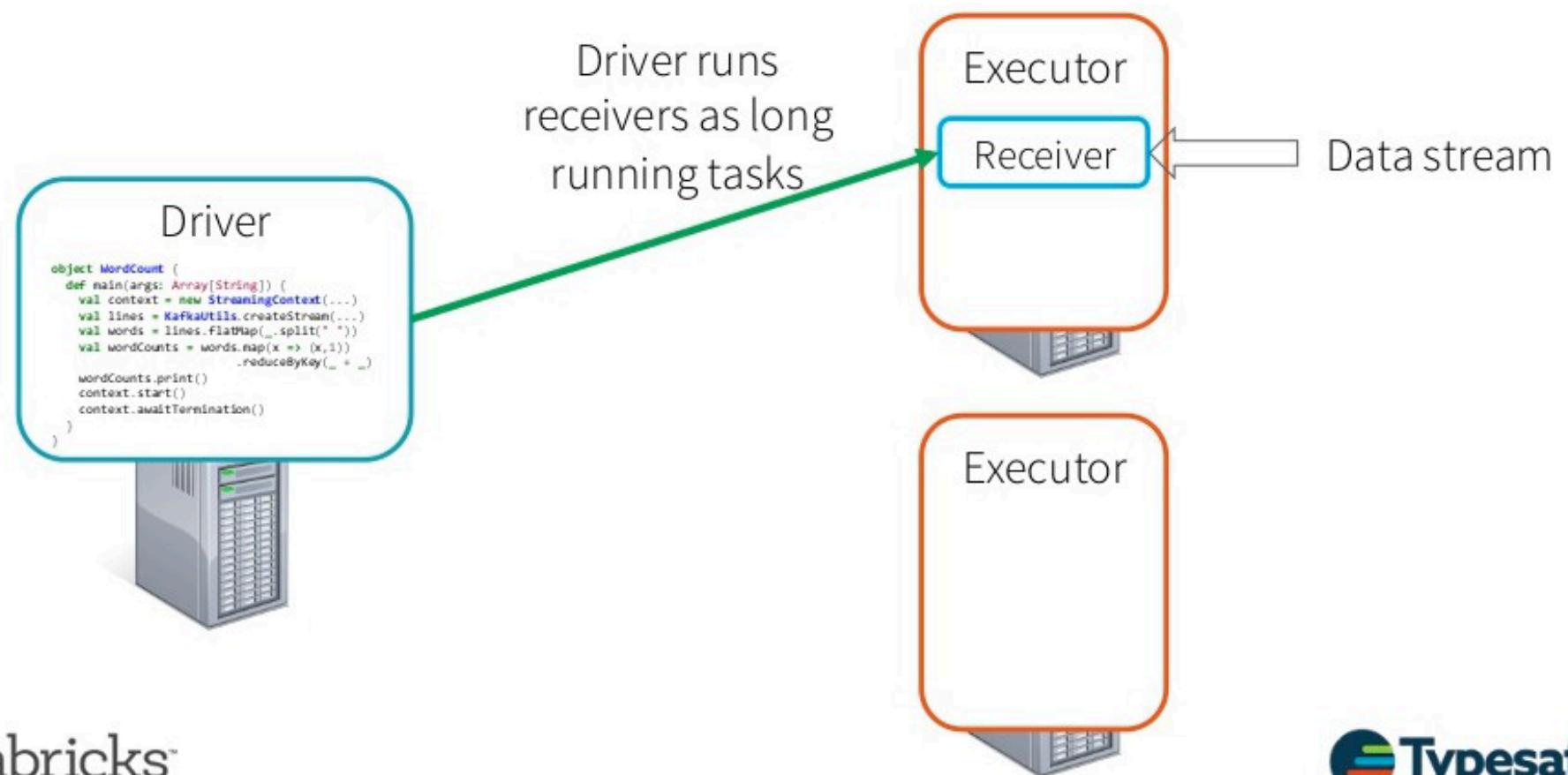
Processed results are pushed out in batches



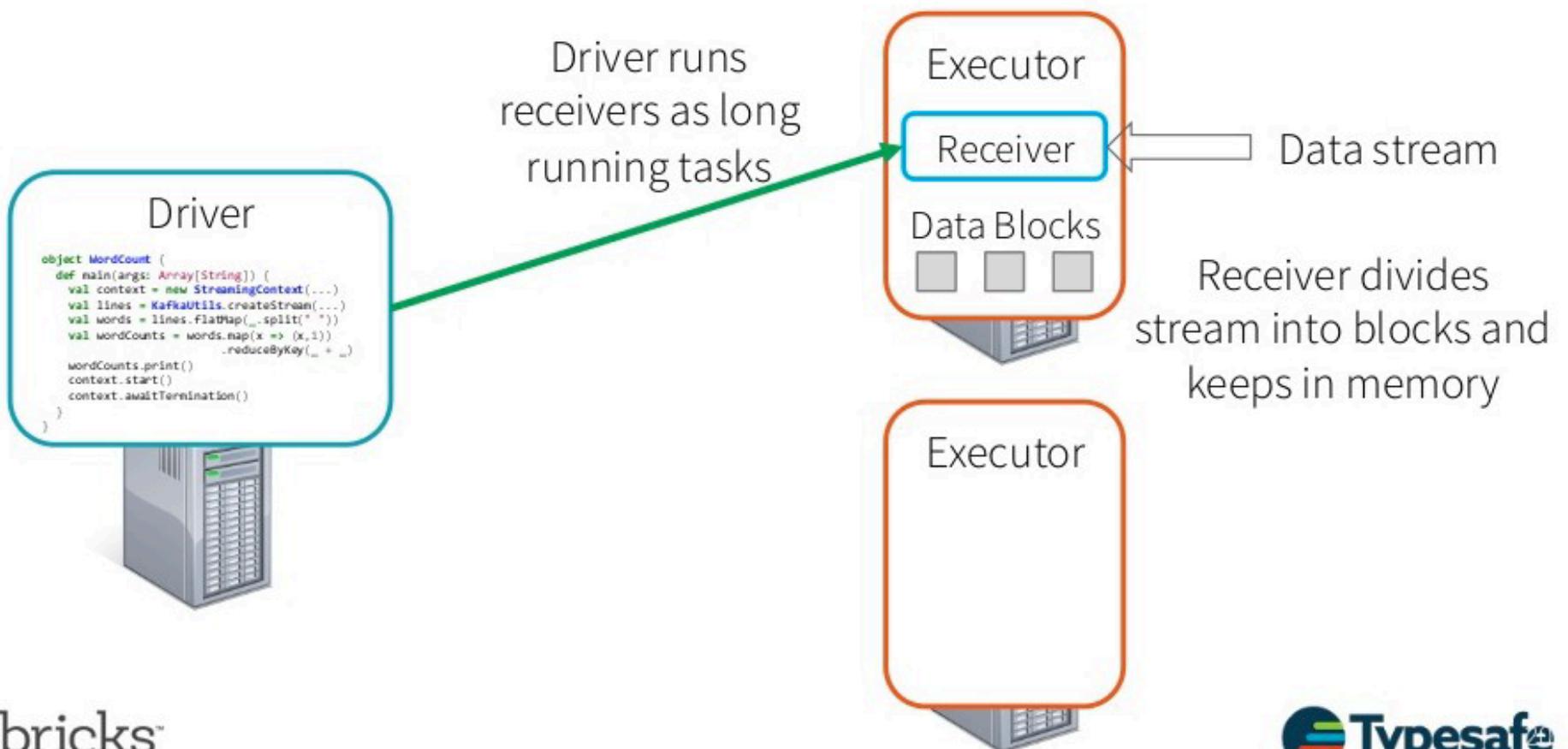
Streaming Architecture



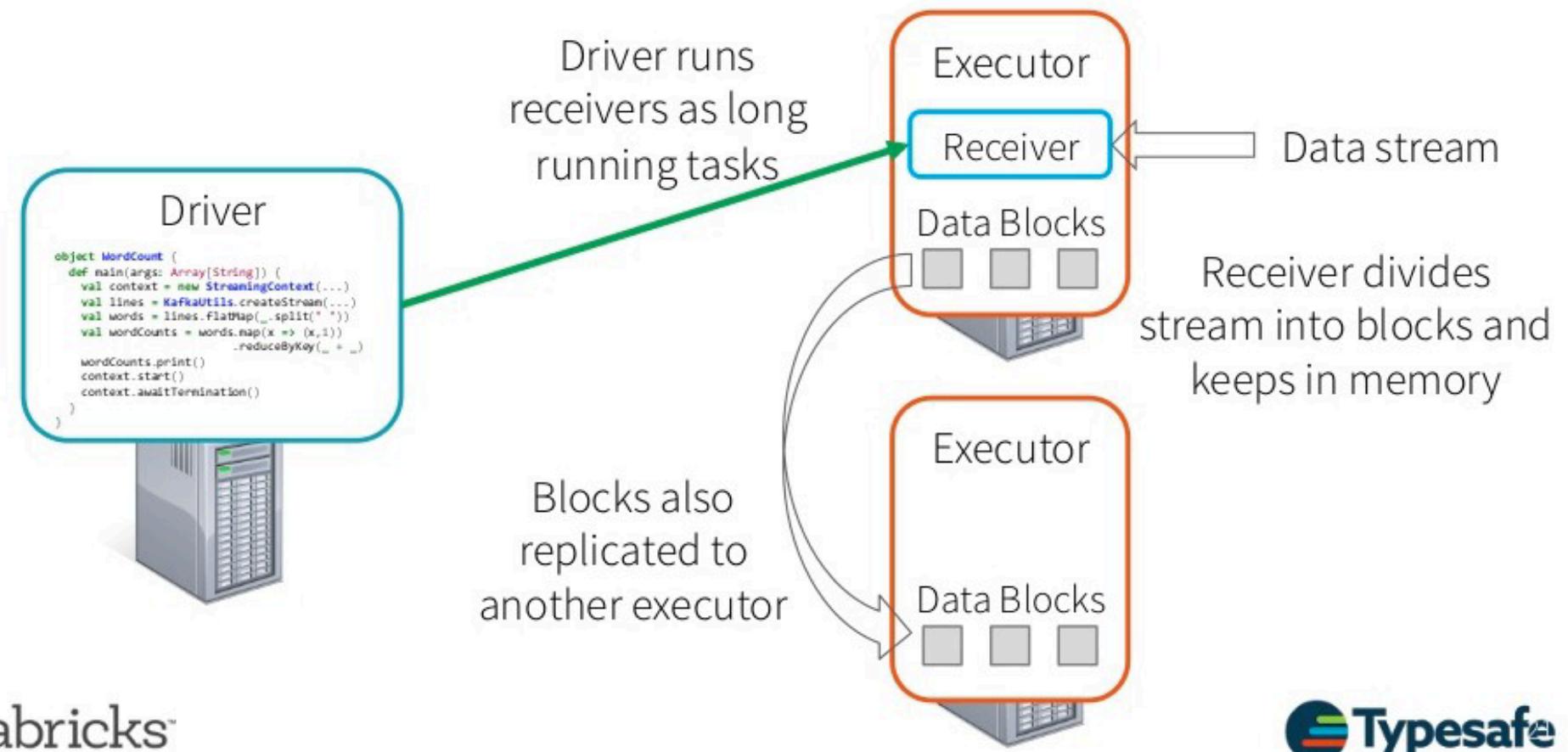
Spark Streaming Application: Receive data



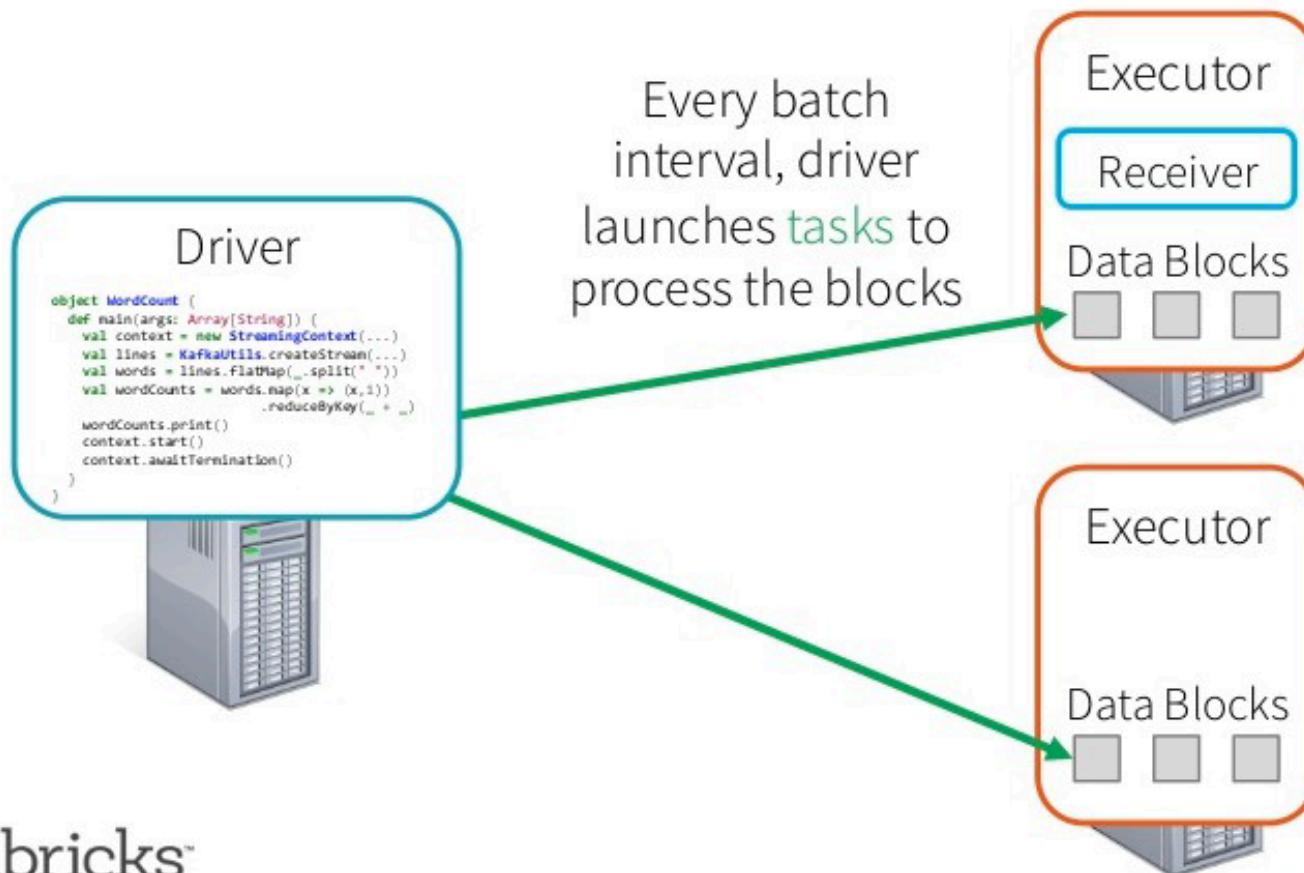
Spark Streaming Application: Receive data



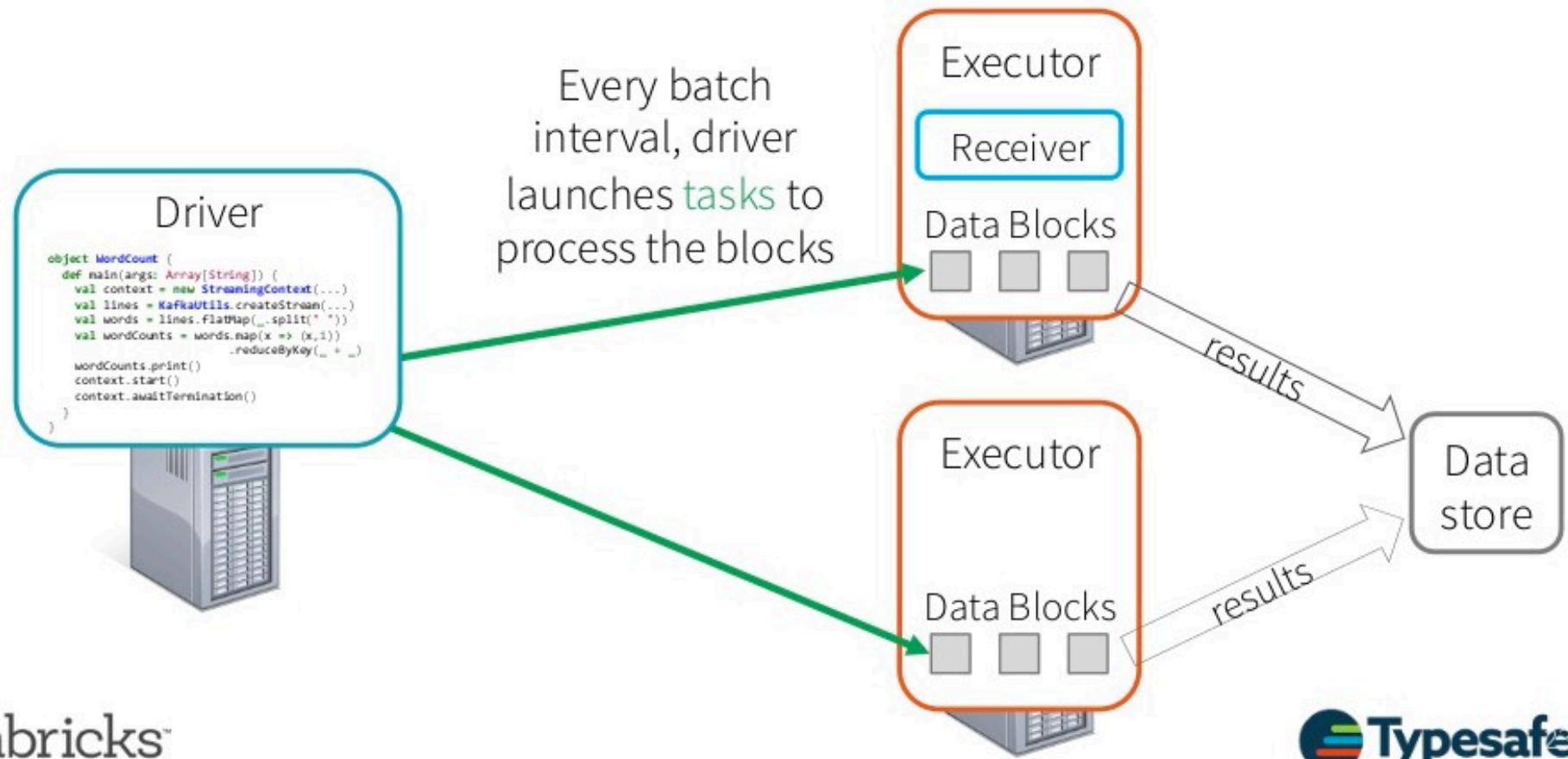
Spark Streaming Application: Receive data



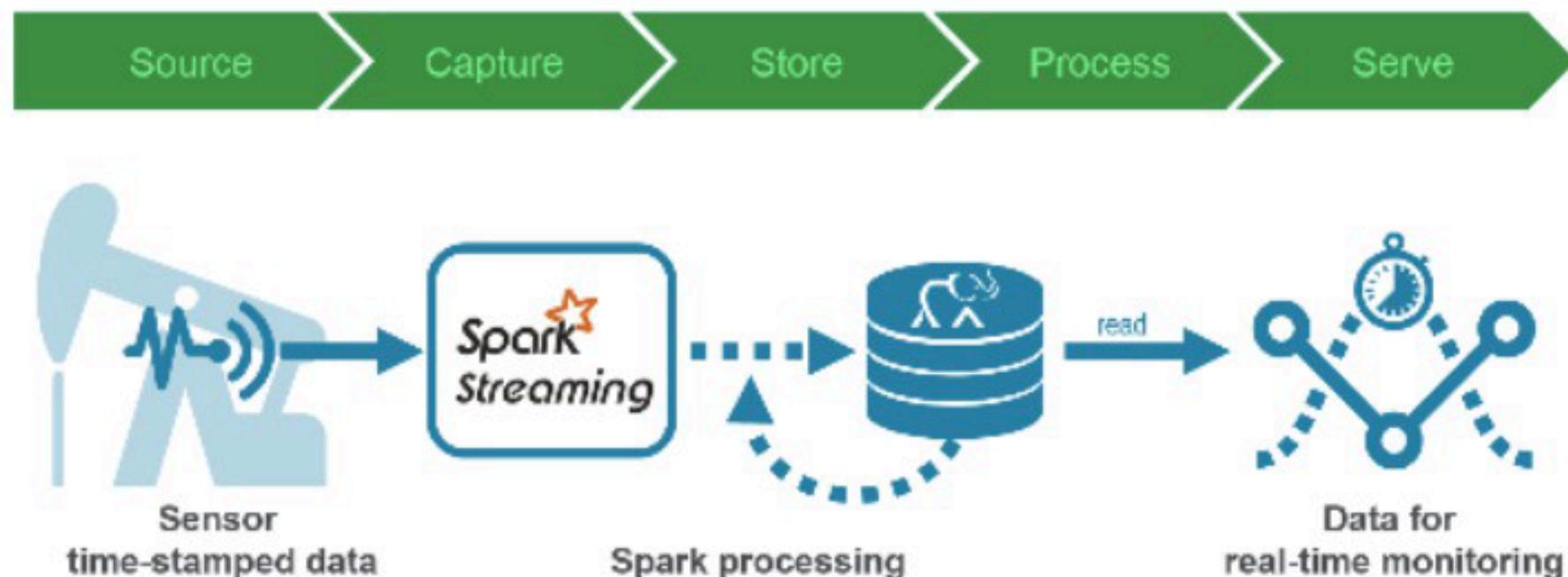
Spark Streaming Application: Process data



Spark Streaming Application: Process data



Use Case: Time Series Data



DStream Functional operations

flatMap(flatMapFunc)

filter(filterFunc)

map(mapFunc)

mapPartitions(mapPartFunc, preservePartitioning)

foreachRDD(foreachFunc)

DStream Output operations

print()

saveAsHadoopFiles(...)

saveAsTextFiles(...)

saveAsObjectFiles(...)

saveAsNewAPIHadoopFiles(...)

foreachRDD(..)

Spark Streaming

Get Example Code

```
$ cd spark
```

```
$ wget https://github.com/bobbylovelomovie/trainbigdata/raw/master/Spark/StreamingWordCount.py
```

```
1  from pyspark import SparkContext
2  from pyspark.streaming import StreamingContext
3
4  # Create a local StreamingContext with two working thread and batch interval of 1 second
5  sc = SparkContext("local[2]", "NetworkWordCount")
6  ssc = StreamingContext(sc, 10)
7
8  lines = ssc.socketTextStream("localhost", 9999)
9  words = lines.flatMap(lambda line: line.split(" "))
10
11 # Count each word in each batch
12 pairs = words.map(lambda word: (word, 1))
13 wordCounts = pairs.reduceByKey(lambda x, y: x + y)
14
15 # Print the first ten elements of each RDD generated in this DStream to the console
16 wordCounts.pprint()
17 #wordCounts.saveAsTextFiles("hdfs:///user/cloudera/output/sparkstream/sparkstream")
18 ssc.start()          # Start the computation
19 ssc.awaitTermination() # Wait for the computation to terminate
```

Spark Streaming

Run Python Spark

```
$ spark-submit StreamingWordCount.py
```

Running the netcat server on another window

```
$ nc -lk 9999
```

```
[cloudera@quickstart ~]$ nc -lk 9999
Hello Bigdata Training
```

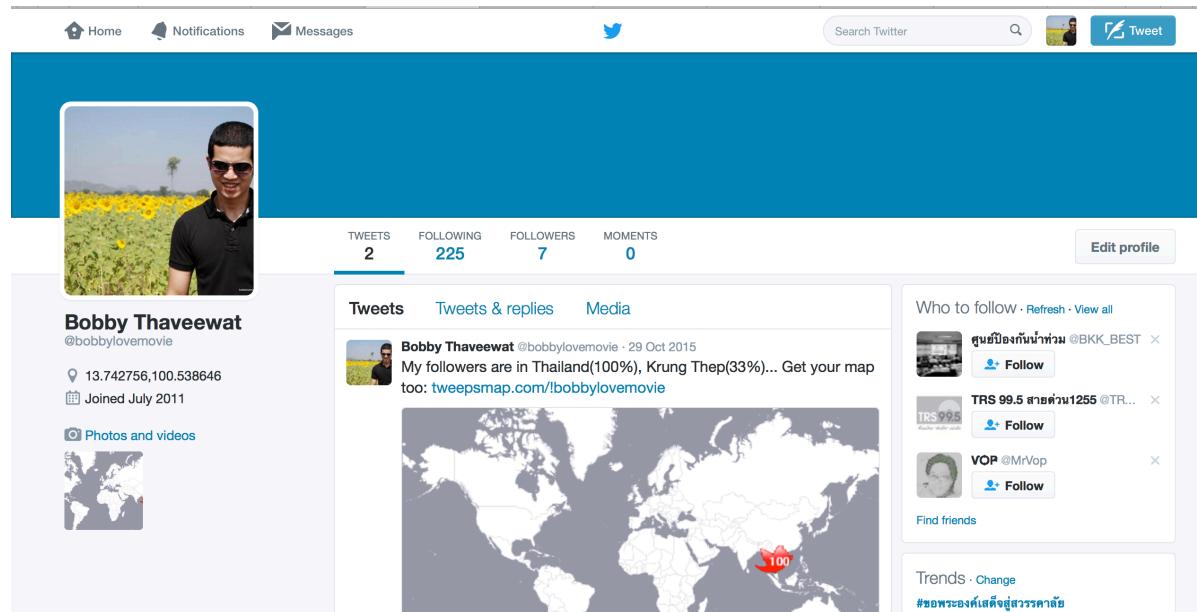
Result SparkStreaming

```
-----
Time: 2016-10-15 08:22:30
-----
```

```
(u'', 1)
(u'Training', 1)
(u'Hello', 1)
(u'Bigdata', 1)
```

Streaming Twitter data

Create a new Twitter App Login to your Twitter @ twitter.com



Create a new Twitter App <https://apps.twitter.com>



Twitter Apps

 [Bobby_Hadoop_App](#)
bobby hadoop Demo App

[Create New App](#)

Create a new Twitter App (cont.)



Enter all the details in the application:

Application Details

Name *

Bobby_SparkStreaming_Demo_App

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Streaming Twitter data Demo App

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

http://www.it.acc.chula.ac.th

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.

(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Agreement

Yes, I have read and agree to the [Twitter Developer Agreement](#).

Create a new Twitter App (cont.)



Your application will be created:

Bobby_SparkStreaming_Demo_App

[Test OAuth](#)

Details

Settings

Keys and Access Tokens

Permissions



Streaming Twitter data Demo App

<http://www.it.acc.chula.ac.th>

Organization

Information about the organization or company associated with your application. This information is optional.

Organization None

Organization website None

Application Settings

Your application's Consumer Key and Secret are used to [authenticate](#) requests to the Twitter Platform.

Access level Read and write ([modify app permissions](#))

Consumer Key (API Key) eSevkYKyO94uGtFxxHoGwXDpX ([manage keys and access tokens](#))

Callback URL None

Callback URL Locked No

Sign in with Twitter Yes

App-only authentication <https://api.twitter.com/oauth2/token>

Request token URL https://api.twitter.com/oauth/request_token

Create a new Twitter App (cont.)



Click on Keys and Access Tokens:

Bobby_SparkStreaming_Demo_App

Details Settings **Keys and Access Tokens** Permissions

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	eSevkYKyO94uGtFxxHoGwXDpX
Consumer Secret (API Secret)	OvjqlF8VmNScaeMGZjz9e1lfq0TxehmwkJ454wHCCUG0AvED
Access Level	Read and write (modify app permissions)
Owner	bobbylovemovie
Owner ID	344100790

Create a new Twitter App (cont.)



Click on Keys and Access Tokens:

Application Actions

[Regenerate Consumer Key and Secret](#) [Change App Permissions](#)

Your Access Token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.

Token Actions



Create my access token

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	344100790-rOcjGTw9wdmD5FVwacOaR7ZriGPRILBZiha43Tyg
Access Token Secret	2t41euCCQiwmdkihxkBWAS2rOTfTJpTAHUy27fOAZILWE
Access Level	Read and write
Owner	bobbylovemovie
Owner ID	344100790

Download the third-party libraries

```
$ wget https://github.com/bobbylovemovie/trainbigdata/raw/master/Spark/  
twitter4j-core-4.0.2.jar  
  
$ wget https://github.com/bobbylovemovie/trainbigdata/raw/master/Spark/  
twitter4j-stream-4.0.2.jar  
  
$ wget https://github.com/bobbylovemovie/trainbigdata/raw/master/Spark/  
spark-streaming-twitter_2.10-1.2.0.jar
```

Run Spark-shell

```
$ spark-shell --jars spark-streaming-twitter_2.10-1.2.0.jar,twitter4j-  
stream-4.0.2.jar,twitter4j-core-4.0.2.jar
```

Running Spark commands

```
$ scala> :paste
// Entering paste mode (ctrl-D to finish)
import org.apache.spark.streaming.twitter._
import twitter4j.auth._
import twitter4j.conf._
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.spark._
import org.apache.spark.streaming._
import org.apache.spark.streaming.StreamingContext._
val ssc = new StreamingContext(sc, Seconds(10))
val cb = new ConfigurationBuilder
```

Running Spark commands(cont.)

```
cb.setDebugEnabled(true).setOAuthConsumerKey("eSevkYKyO94uGtFxxHoG  
wXDpX").setOAuthConsumerSecret("OvjqlF8VmNScaeMGZjz9e1flfq0Txehmw  
kJMJ454wHCCUG0AvED").setOAuthAccessToken("344100790-  
rOcjGTw9wdmD5FVwacOaR7ZriGPRILBZiha43Tyg").setOAuthAccessTokenSe  
cret("2t41euCCQiwmdkihxkBWAS2rOTfTJpTAHUy27fOAZILWE")  
val auth = new OAuthAuthorization(cb.build)  
val tweets = TwitterUtils.createStream(ssc,Some(auth))  
val status = tweets.map(status => status.getText)  
status.print  
ssc.checkpoint("hdfs://user/cloudera/data/tweets")  
ssc.start  
ssc.awaitTermination
```

HUE Query Editors Data Browsers Workflows Search Security

File Browser

Search for file name Actions Move to trash Upload New

Home / user / cloudera / data / tweets History Trash

<input type="checkbox"/>	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	..		cloudera	cloudera	drwxr-xr-x	October 15, 2016 09:07 AM
<input type="checkbox"/>	.		cloudera	cloudera	drwxr-xr-x	October 15, 2016 09:09 AM
<input type="checkbox"/>	5f370383-6967-4d7f-8e04-20526f5fcce1		cloudera	cloudera	drwxr-xr-x	October 15, 2016 09:07 AM
<input type="checkbox"/>	checkpoint-1476547700000	5.0 KB	cloudera	cloudera	-rw-r--r--	October 15, 2016 09:08 AM
<input type="checkbox"/>	checkpoint-1476547710000	5.0 KB	cloudera	cloudera	-rw-r--r--	October 15, 2016 09:08 AM
<input type="checkbox"/>	checkpoint-1476547720000	5.0 KB	cloudera	cloudera	-rw-r--r--	October 15, 2016 09:08 AM
<input type="checkbox"/>	checkpoint-1476547730000	5.0 KB	cloudera	cloudera	-rw-r--r--	October 15, 2016 09:08 AM
<input type="checkbox"/>	checkpoint-1476547740000	5.0 KB	cloudera	cloudera	-rw-r--r--	October 15, 2016 09:09 AM
<input type="checkbox"/>	checkpoint-1476547750000	5.0 KB	cloudera	cloudera	-rw-r--r--	October 15, 2016 09:09 AM
<input type="checkbox"/>	checkpoint-1476547760000	5.0 KB	cloudera	cloudera	-rw-r--r--	October 15, 2016 09:09 AM