# PRINCIPLES OF BIG DATA MANAGEMENT

## PHASE #1

**TEAM SIZE:3**

**TEAM MEMBERS:** Sai Tejaswi Koppuravuri (sk6zb@mail.umkc.edu)
Pallavi Desai (pd2qd@mail.umkc.edu)
Anusha Palla (apgmc@mail.umkc.edu)

## Github Link of the Project:

https://github.com/SaitejaswiK/Principles-of-BigData-Management

## Objective:

- The principle point of this stage is to build up a framework to store, break down, and envision a social network's.
- Tasks:
  1. Collect social network's information (e.g. tweets) in any format preferred JSON.
  2. Store the content substance (e.g. tweet's content) from the information into a document in HDFS.
  3. Run a Word Count program in Apache Spark and Hadoop on the content document and store the yield and log records locally

## Applications/Software's Used:

Twitter Developer Account, Apache Spark, Python Hadoop.

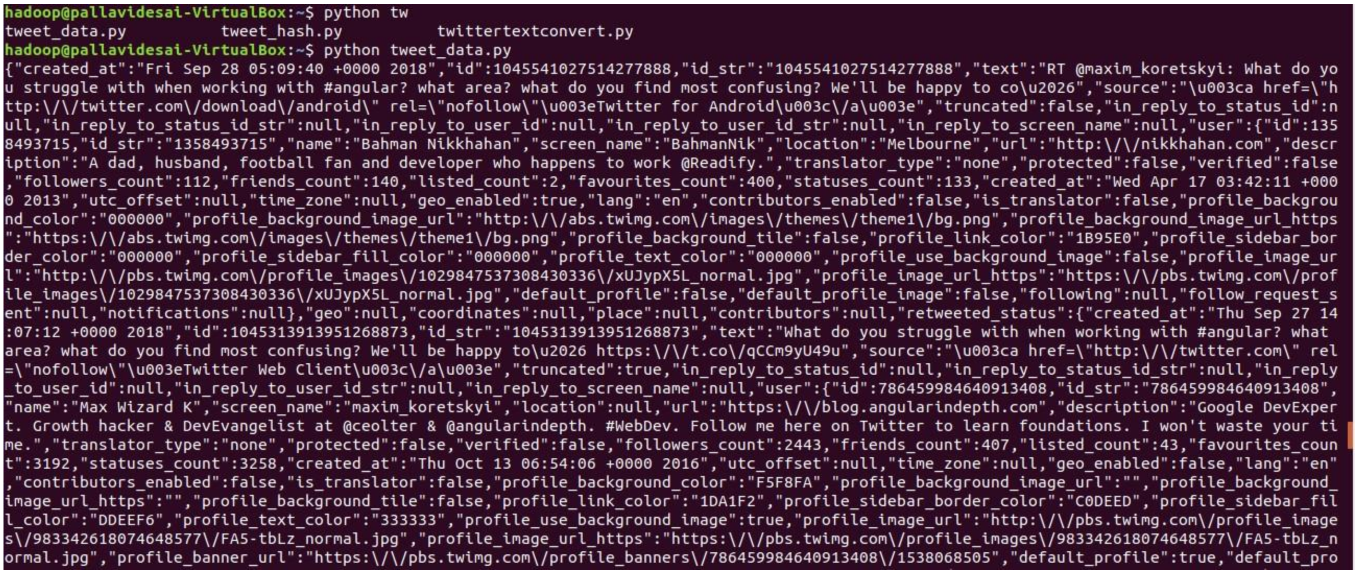
## Collecting tweets from Twitter:

- Firstly, we have made an developer account in Twitter utilizing beneath connect. https://apps.twitter.com/

- Below are the factors that contains the client certifications to get to Twitter API
    - ACCESS_TOKEN = " 779311765163171844-RCUoOhu2R53ugDk3O8xTX50rgi2zj4o"
    - ACCESS_SECRET = " y9Evdnwz1tfI43fIyun18OQOxgt6HQjWh6g3Gb99ExwOI"
    - CONSUMER_KEY = " xMJiyum9ZLKuGeZDPl1uL3qeU"
    - CONSUMER_SECRET = “6df8h8k2O7AwBJgYREWwTfwB1MFXVBuUm4PttByrGiRKDj6bI5"

- We have composed python program that is utilized to bring tweets in JSON design. (tweet_ data.py)

Link:[https://github.com/SaitejaswiK/Principles-of-BigDataManagement/blob/master/Source/Python%20Programs/tweet_data.py](https://github.com/SaitejaswiK/Principles-of-BigDataManagement/blob/master/Source/Python%20Programs/tweet_data.py)

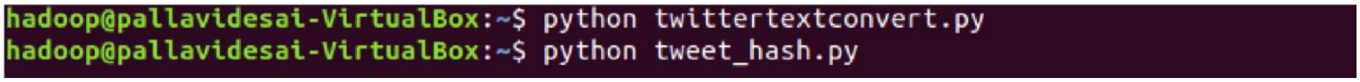


```
hadoop@pallavidesai-VirtualBox:~$ python tw
tweet_data.py          tweet_hash.py          twittertextconvert.py
hadoop@pallavidesai-VirtualBox:~$ python tweet_data.py
{"created_at":"Fri Sep 28 05:09:40 +0000 2018","id":1045541027514277888,"id_str":"1045541027514277888","text":"RT @maxim_koretskyi: What do yo
u struggle with when working with #angular? what area? what do you find most confusing? We'll be happy to co\u2026","source":"\u003ca href=\"h
ttp:\/\/twitter.com\/download\/android\" rel=\"nofollow\"\u003eTwitter for Android\u003c\/a\u003e","truncated":false,"in_reply_to_status_id":n
ull,"in_reply_to_status_id_str":null,"in_reply_to_user_id":null,"in_reply_to_user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":135
8493715,"id_str":"1358493715","name":"Bahman Nikkhahan","screen_name":"BahmanNik","location":"Melbourne","url":"http:\/\/nikkhahan.com","descr
iption":"A dad, husband, football fan and developer who happens to work @Readify.","translator_type":"none","protected":false,"verified":false
,"followers_count":112,"friends_count":140,"listed_count":2,"favourites_count":400,"statuses_count":133,"created_at":"Wed Apr 17 03:42:11 +000
0 2013","utc_offset":null,"time_zone":null,"geo_enabled":true,"lang":"en","contributors_enabled":false,"is_translator":false,"profile_backgrou
nd_color":"000000","profile_background_image_url":"http:\/\/abs.twimg.com\/images\/themes\/theme1\/bg.png","profile_background_image_url_https
":"https:\/\/abs.twimg.com\/images\/themes\/theme1\/bg.png","profile_background_tile":false,"profile_link_color":"1B95E0","profile_sidebar_bor
der_color":"000000","profile_sidebar_fill_color":"000000","profile_text_color":"000000","profile_use_background_image":false,"profile_image_ur
l":"http:\/\/pbs.twimg.com\/profile_images\/1029847537308430336\/xUJypX5L_normal.jpg","profile_image_url_https":"https:\/\/pbs.twimg.com\/prof
ile_images\/1029847537308430336\/xUJypX5L_normal.jpg","default_profile":false,"default_profile_image":false,"following":null,"follow_request_s
ent":null,"notifications":null},"geo":null,"coordinates":null,"place":null,"contributors":null,"retweeted_status":{"created_at":"Thu Sep 27 14
:07:12 +0000 2018","id":1045313913951268873,"id_str":"1045313913951268873","text":"What do you struggle with when working with #angular? what
area? what do you find most confusing? We'll be happy to\u2026 https:\/\/t.co\/qCCm9yU49u","source":"\u003ca href=\"http:\/\/twitter.com\" rel
=\"nofollow\"\u003eTwitter Web Client\u003c\/a\u003e","truncated":true,"in_reply_to_status_id":null,"in_reply_to_status_id_str":null,"in_reply
_to_user_id":null,"in_reply_to_user_id_str":null,"in_reply_to_screen_name":null,"user":{"id":786459984640913408,"id_str":"786459984640913408",
"name":"Max Wizard K","screen_name":"maxim_koretskyi","location":null,"url":"https:\/\/blog.angularindepth.com","description":"Google DevExper
t. Growth hacker & DevEvangelist at @ceolter & @angularindepth. #WebDev. Follow me here on Twitter to learn foundations. I won't waste your ti
me.","translator_type":"none","protected":false,"verified":false,"followers_count":2443,"friends_count":407,"listed_count":43,"favourites_coun
t":3192,"statuses_count":3258,"created_at":"Thu Oct 13 06:54:06 +0000 2016","utc_offset":null,"time_zone":null,"geo_enabled":false,"lang":"en"
,"contributors_enabled":false,"is_translator":false,"profile_background_color":"F5F8FA","profile_background_image_url":"","profile_background_
image_url_https":"","profile_background_tile":false,"profile_link_color":"1DA1F2","profile_sidebar_border_color":"C0DEED","profile_sidebar_fil
l_color":"DDEEF6","profile_text_color":"333333","profile_use_background_image":true,"profile_image_url":"http:\/\/pbs.twimg.com\/profile_image
s\/983342618074648577\/FA5-tbLz_normal.jpg","profile_image_url_https":"https:\/\/pbs.twimg.com\/profile_images\/983342618074648577\/FA5-tbLz_n
ormal.jpg","profile_banner_url":"https:\/\/pbs.twimg.com\/profile_banners\/786459984640913408\/1538068505","default_profile":true,"default_pro
```

**Fig1: Tweets collection**

- The extricated record in JSON arrange contains all the tweet points of interest, for example, id, created at, text, profile_background_image_url and so forth.
- From JSON tweets record just the content substance is extricated utilizing Python program. The got content points of interest are put away in a record. (twittertextconvert.py)

Link:[https://github.com/SaitejaswiK/Principles-of-BigData-Management/blob/master/Source/Python%20Programs/twittertextconvert.py](https://github.com/SaitejaswiK/Principles-of-BigData-Management/blob/master/Source/Python%20Programs/twittertextconvert.py)



```
hadoop@pallavidesai-VirtualBox:~$ python twittertextconvert.py
hadoop@pallavidesai-VirtualBox:~$ python tweet_hash.py
```

**Fig 2: Creating a python file for Hashtags extraction**

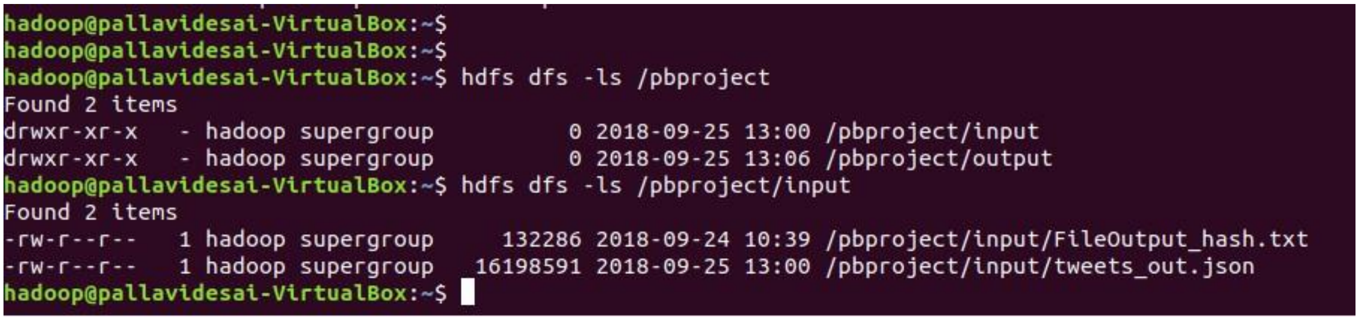
## Store the text content (e.g. tweet's text) from the data into a file in HDFS.

- The twitter tweets content substance record is moved from local to HFDS.
- First a folder is made in HDFS and the content document is moved from local to HDFS utilizing underneath order.

**Make directory in local**: hadoop fs - mkdir pbproject/input

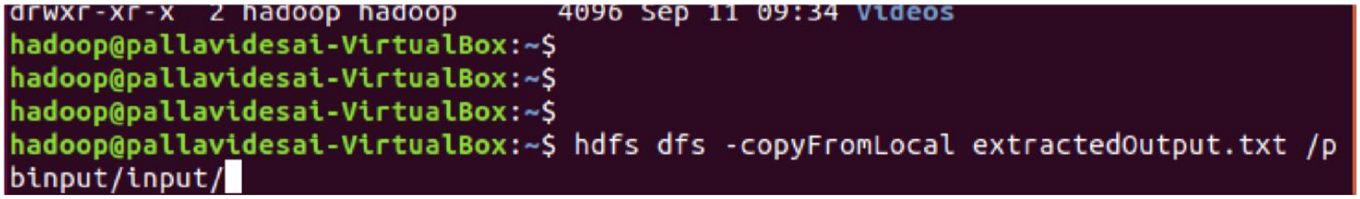
**Move content record from local to HDFS**: hadoop fs - copyFromLocal FileOutput.txt pbproject/input

**To list the records under a registry**: hadoop fs - ls pbproject/input



```
hadoop@pallavidesai-VirtualBox:~$
hadoop@pallavidesai-VirtualBox:~$
hadoop@pallavidesai-VirtualBox:~$ hdfs dfs -ls /pbproject
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2018-09-25 13:00 /pbproject/input
drwxr-xr-x   - hadoop supergroup          0 2018-09-25 13:06 /pbproject/output
hadoop@pallavidesai-VirtualBox:~$ hdfs dfs -ls /pbproject/input
Found 2 items
-rw-r--r--   1 hadoop supergroup     132286 2018-09-24 10:39 /pbproject/input/FileOutput_hash.txt
-rw-r--r--   1 hadoop supergroup   16198591 2018-09-25 13:00 /pbproject/input/tweets_out.json
hadoop@pallavidesai-VirtualBox:~$
```

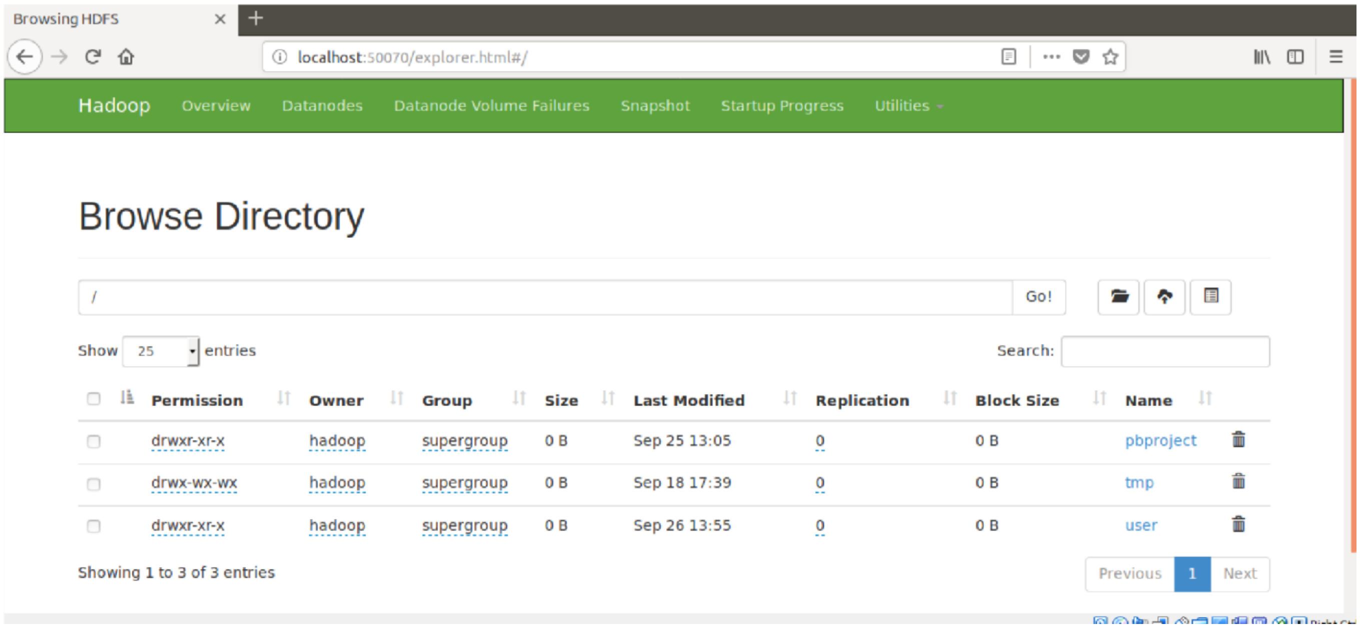
**Fig 3: HDFS Commands**



```
drwxr-xr-x  2 hadoop hadoop      4096 Sep 11 09:34 Videos
hadoop@pallavidesai-VirtualBox:~$
hadoop@pallavidesai-VirtualBox:~$
hadoop@pallavidesai-VirtualBox:~$
hadoop@pallavidesai-VirtualBox:~$ hdfs dfs -copyFromLocal extractedOutput.txt /p
binput/input/
```

**Fig 4: Creating Extracted Text in Hadoop**

- The directory created and the files moved to HDFS can be viewed as shown below.



Browse Directory

/

Show 25 entries

Search:

| □ | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|---|
| □ | drwxr-xr-x | hadoop | supergroup | 0 B | Sep 25 13:05 | 0 | 0 B | pbproject |
| □ | drwx-wx-wx | hadoop | supergroup | 0 B | Sep 18 17:39 | 0 | 0 B | tmp |
| □ | drwxr-xr-x | hadoop | supergroup | 0 B | Sep 26 13:55 | 0 | 0 B | user |

Showing 1 to 3 of 3 entries

**Fig 5: Directory in HDFS**

Link of the tweets output: https://github.com/SaitejaswiK/Principles-of-BigDataManagement/blob/master/Source/Twitter%20Tweets/FileOutput.txt

## Extracting Hashtags from the obtained output:

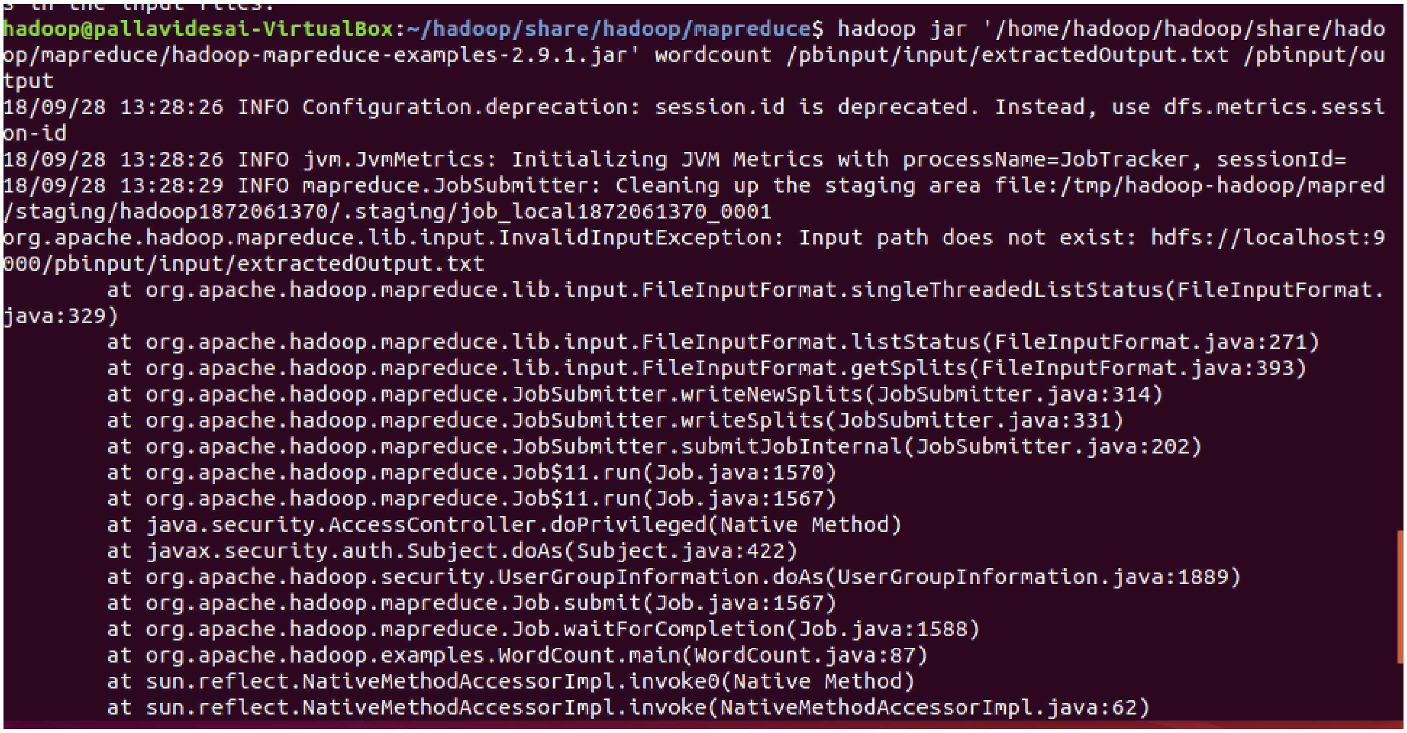
- A python program has been written for the extraction of URL's and hashtags from the obtained output. (tweets_hash.py)

  Link of the code: https://github.com/SaitejaswiK/Principles-of-BigData-Management/blob/master/Source/Python%20Programs/tweet_hash.py

Link of the extracted Hashtags output: https://github.com/SaitejaswiK/Principles-of-BigDataManagement/blob/master/Source/Twitter%20Tweets/FileOutput_hash.txt

## Run a Word Count program in Apache Hadoop on the text file and store the output and log files locally.

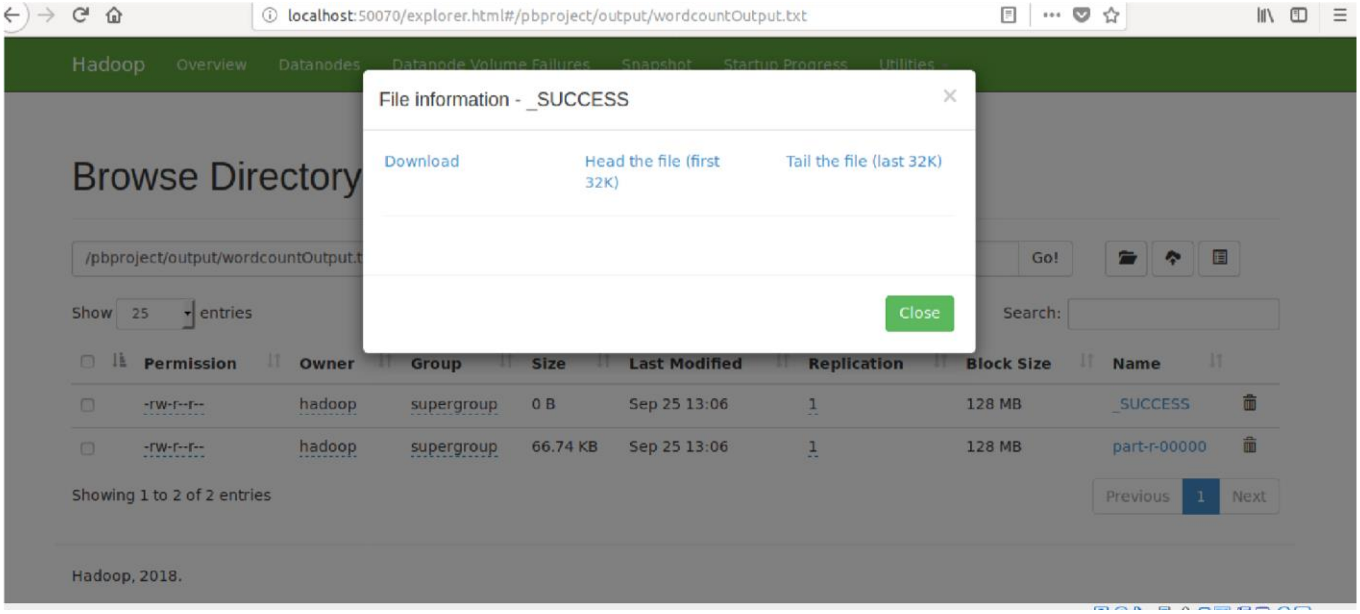
- First of all, to run word count program on set of data we require.
- Using Hadoop, run the word count example for the obtained tweets file.



```
s in the input files:
hadoop@pallavidesai-VirtualBox:~/hadoop/share/hadoop/mapreduce$ hadoop jar '/home/hadoop/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.1.jar' wordcount /pbinput/input/extractedOutput.txt /pbinput/output
18/09/28 13:28:26 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
18/09/28 13:28:26 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
18/09/28 13:28:29 INFO mapreduce.JobSubmitter: Cleaning up the staging area file:/tmp/hadoop-hadoop/mapred/staging/hadoop1872061370/.staging/job_local1872061370_0001
org.apache.hadoop.mapreduce.lib.input.InvalidInputException: Input path does not exist: hdfs://localhost:9000/pbinput/input/extractedOutput.txt
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.singleThreadedListStatus(FileInputFormat.java:329)
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.listStatus(FileInputFormat.java:271)
        at org.apache.hadoop.mapreduce.lib.input.FileInputFormat.getSplits(FileInputFormat.java:393)
        at org.apache.hadoop.mapreduce.JobSubmitter.writeNewSplits(JobSubmitter.java:314)
        at org.apache.hadoop.mapreduce.JobSubmitter.writeSplits(JobSubmitter.java:331)
        at org.apache.hadoop.mapreduce.JobSubmitter.submitJobInternal(JobSubmitter.java:202)
        at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1570)
        at org.apache.hadoop.mapreduce.Job$11.run(Job.java:1567)
        at java.security.AccessController.doPrivileged(Native Method)
        at javax.security.auth.Subject.doAs(Subject.java:422)
        at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1889)
        at org.apache.hadoop.mapreduce.Job.submit(Job.java:1567)
        at org.apache.hadoop.mapreduce.Job.waitForCompletion(Job.java:1588)
        at org.apache.hadoop.examples.WordCount.main(WordCount.java:87)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
```

**Fig 6: Running Wordcount in Hadoop**

## Output in Hadoop Browser



File information - _SUCCESS

Download | Head the file (first 32K) | Tail the file (last 32K)

Close

Browse Directory

/pbproject/output/wordcountOutput.t

Go!

Show 25 entries

Search:

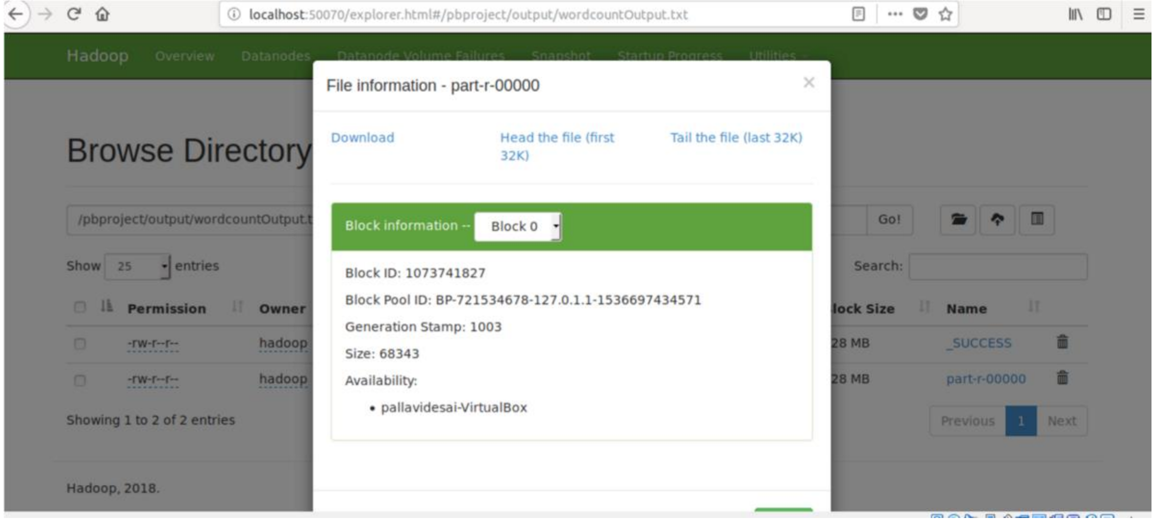
| | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|---|
| □ | -rw-r--r-- | hadoop | supergroup | 0 B | Sep 25 13:06 | 1 | 128 MB | _SUCCESS |
| □ | -rw-r--r-- | hadoop | supergroup | 66.74 KB | Sep 25 13:06 | 1 | 128 MB | part-r-00000 |

Showing 1 to 2 of 2 entries

Previous 1 Next

Hadoop, 2018.

**Fig 7: Files in HDFS**



File information - part-r-00000

Download | Head the file (first 32K) | Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073741827

Block Pool ID: BP-721534678-127.0.1.1-1536697434571

Generation Stamp: 1003

Size: 68343

Availability:

- pallavidesai-VirtualBox

Browse Directory

/pbproject/output/wordcountOutput.t

Go!

Show 25 entries

Search:

| | Permission | Owner | lock Size | Name |
|---|---|---|---|---|
| □ | -rw-r--r-- | hadoop | 28 MB | _SUCCESS |
| □ | -rw-r--r-- | hadoop | 28 MB | part-r-00000 |

Showing 1 to 2 of 2 entries

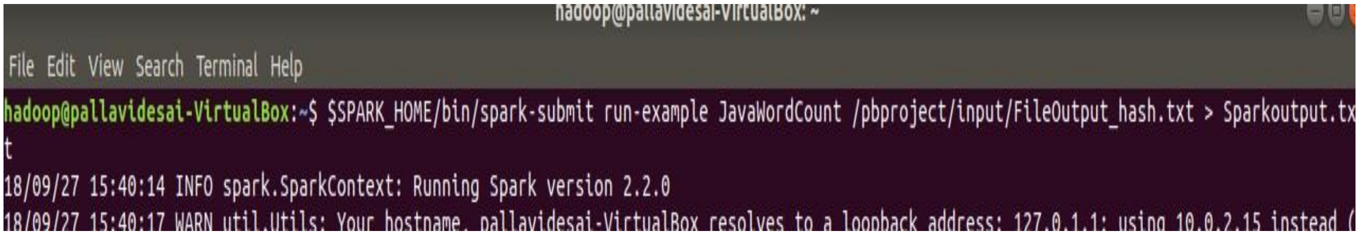
Previous 1 Next

Hadoop, 2018.

**Fig 8: Directory Information**

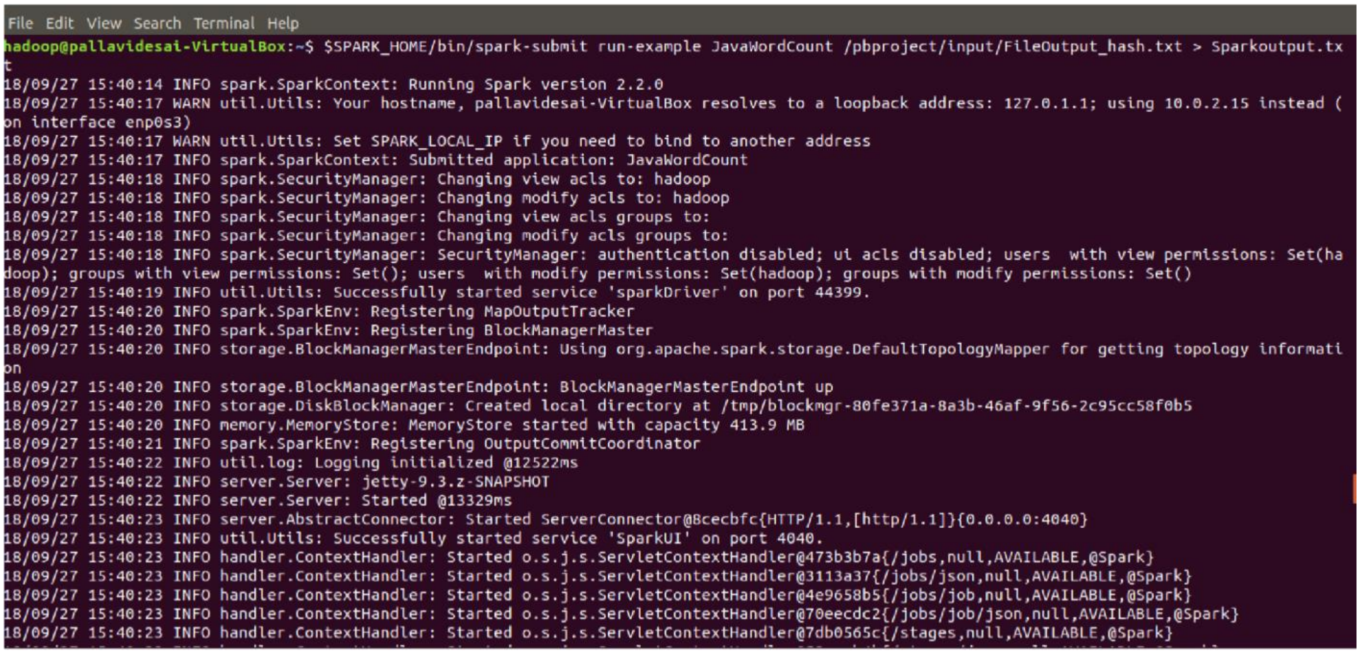
## Run a Word Count program in Apache Spark on the text file and store the output and log files locally.

- Then,  after running the word count example on Hadoop, now it's time to run the same word count example using Apache Spark.
- The output obtained from the word count running on Apache Hadoop is almost similar to the output obtained from Apache Spark except the minor differences.


hadoop@pallavidesai-VirtualBox: ~
File Edit View Search Terminal Help

```
hadoop@pallavidesai-VirtualBox:~$ $SPARK_HOME/bin/spark-submit run-example JavaWordCount /pbproject/input/FileOutput_hash.txt > Sparkoutput.tx
t
18/09/27 15:40:14 INFO spark.SparkContext: Running Spark version 2.2.0
18/09/27 15:40:17 WARN util.Utils: Your hostname, pallavidesai-VirtualBox resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (
```

**Fig 9: Spark Commands**



```
hadoop@pallavidesai-VirtualBox:~$ $SPARK_HOME/bin/spark-submit run-example JavaWordCount /pbproject/input/FileOutput_hash.txt > Sparkoutput.tx
t
18/09/27 15:40:14 INFO spark.SparkContext: Running Spark version 2.2.0
18/09/27 15:40:17 WARN util.Utils: Your hostname, pallavidesai-VirtualBox resolves to a loopback address: 127.0.1.1; using 10.0.2.15 instead (
on interface enp0s3)
18/09/27 15:40:17 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
18/09/27 15:40:17 INFO spark.SparkContext: Submitted application: JavaWordCount
18/09/27 15:40:18 INFO spark.SecurityManager: Changing view acls to: hadoop
18/09/27 15:40:18 INFO spark.SecurityManager: Changing modify acls to: hadoop
18/09/27 15:40:18 INFO spark.SecurityManager: Changing view acls groups to:
18/09/27 15:40:18 INFO spark.SecurityManager: Changing modify acls groups to:
18/09/27 15:40:18 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(ha
doop); groups with view permissions: Set(); users  with modify permissions: Set(hadoop); groups with modify permissions: Set()
18/09/27 15:40:19 INFO util.Utils: Successfully started service 'sparkDriver' on port 44399.
18/09/27 15:40:20 INFO spark.SparkEnv: Registering MapOutputTracker
18/09/27 15:40:20 INFO spark.SparkEnv: Registering BlockManagerMaster
18/09/27 15:40:20 INFO storage.BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology informati
on
18/09/27 15:40:20 INFO storage.BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
18/09/27 15:40:20 INFO storage.DiskBlockManager: Created local directory at /tmp/blockmgr-80fe371a-8a3b-46af-9f56-2c95cc58f0b5
18/09/27 15:40:20 INFO memory.MemoryStore: MemoryStore started with capacity 413.9 MB
18/09/27 15:40:21 INFO spark.SparkEnv: Registering OutputCommitCoordinator
18/09/27 15:40:22 INFO util.log: Logging initialized @12522ms
18/09/27 15:40:22 INFO server.Server: jetty-9.3.z-SNAPSHOT
18/09/27 15:40:22 INFO server.Server: Started @13329ms
18/09/27 15:40:23 INFO server.AbstractConnector: Started ServerConnector@8cecbfc{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
18/09/27 15:40:23 INFO util.Utils: Successfully started service 'SparkUI' on port 4040.
18/09/27 15:40:23 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@473b3b7a{/jobs,null,AVAILABLE,@Spark}
18/09/27 15:40:23 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@3113a37{/jobs/json,null,AVAILABLE,@Spark}
18/09/27 15:40:23 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@4e9658b5{/jobs/job,null,AVAILABLE,@Spark}
18/09/27 15:40:23 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@70eecdc2{/jobs/job/json,null,AVAILABLE,@Spark}
18/09/27 15:40:23 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@7db0565c{/stages,null,AVAILABLE,@Spark}
```

**Fig 10: Sample Word Count Output**

Word count output Link:

https://github.com/SaitejaswiK/Principles-of-BigData-Management/blob/master/Source/WordCount%20Output/part-r-00000