

## TABLE OF CONTENTS

<b>1. Start the name node and data nodes in cluster</b>	<b>1</b>
<b>2. Start the YARN resource and node managers</b>	<b>1</b>
<b>3. Check if all the daemons are active and running as Java processes</b>	<b>1</b>
<b>4. Access to the Hadoop NameNode using 9870 port.</b>	<b>2</b>
<b>5. Access individual Data Nodes using port 9864</b>	<b>2</b>
<b>6. The YARN Resource Manager is accessible on port 8088</b>	<b>3</b>
<b>Running Word count Example</b>	<b>3</b>
<b>1. Listing Files in HDFS</b>	<b>3</b>
<b>2. Inserting Data into HDFS</b>	<b>3</b>
<b>3. Running Word count Command</b>	<b>6</b>
<b>4. Show Results</b>	<b>6</b>

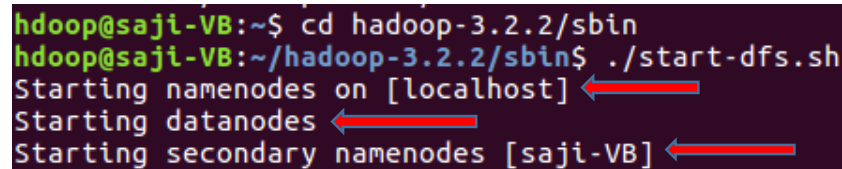
## Table of Figures

Figure 1 1. Start the name node and data nodes in cluster .....	1
Figure 2 Start the YARN resource and node managers .....	1
Figure 3 Check if all the daemons are active and running .....	1
Figure 4 Hadoop NameNode in browser .....	2
Figure 5 hadoop Data Node in browser .....	2
Figure 6 YARN Resource Manager in browser .....	3
Figure 7 create user directory .....	3
Figure 8 user directory .....	4
Figure 9 create user input directory .....	4
Figure 10 input directory .....	4
Figure 11 Download input text file .....	5
Figure 12 put command .....	5
Figure 13 file verification .....	5
Figure 14 created file inside /user/input directory .....	5
Figure 15 Home directory .....	6
Figure 16 2.perform the word count Map Reduce program .....	6
Figure 17 file name verification in result file .....	6
Figure 18 created output file .....	6
Figure 19 output success .....	7
Figure 20 word count .....	7

After successfully installed Hadoop on Ubuntu and deployed it in a pseudo-distributed mode. Then setting up the java and hadoop then start the distributed file system and follow the command listed below to start the namenode as well as the data nodes in cluster. Follow the prerequisites steps to run map reduce program.

### 1. Start the name node and data nodes in cluster

```
$ ./start-dfs.sh
```



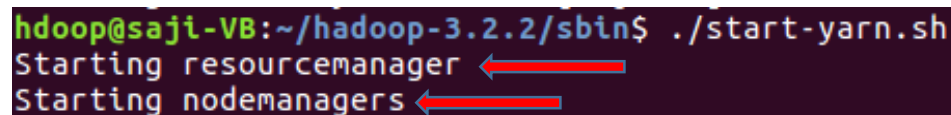
```
hadoop@saji-VB:~$ cd hadoop-3.2.2/sbin
hadoop@saji-VB:~/hadoop-3.2.2/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [saji-VB]
```

*Figure 1 1. Start the name node and data nodes in cluster*

Output shows that namenode, datanodes, and secondary namenode are running

### 2. Start the YARN resource and node managers

```
$ ./start-yarn.sh
```



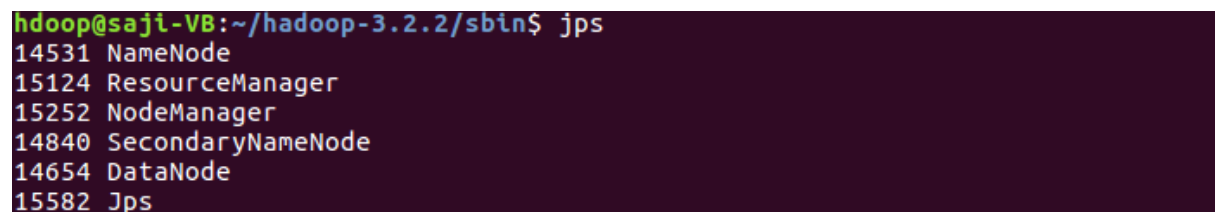
```
hadoop@saji-VB:~/hadoop-3.2.2/sbin$ ./start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

*Figure 2 Start the YARN resource and node managers*

Output informs that the processes are starting

### 3. Check if all the daemons are active and running as Java processes

```
$ jps
```



```
hadoop@saji-VB:~/hadoop-3.2.2/sbin$ jps
14531 NameNode
15124 ResourceManager
15252 NodeManager
14840 SecondaryNameNode
14654 DataNode
15582 Jps
```

*Figure 3 Check if all the daemons are active and running*

#### 4. Access to the Hadoop NameNode using 9870 port.

<http://localhost:9870>

The Name Node user interface provides a comprehensive overview of the entire cluster.

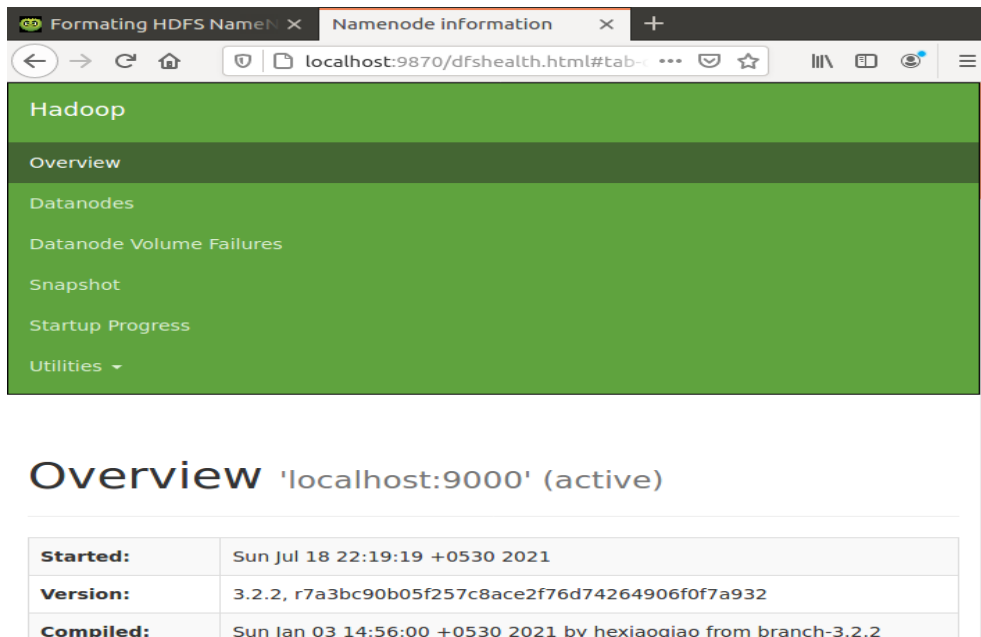


Figure 4 Hadoop NameNode in browser

#### 5. Access individual Data Nodes using port 9864

<http://localhost:9864>

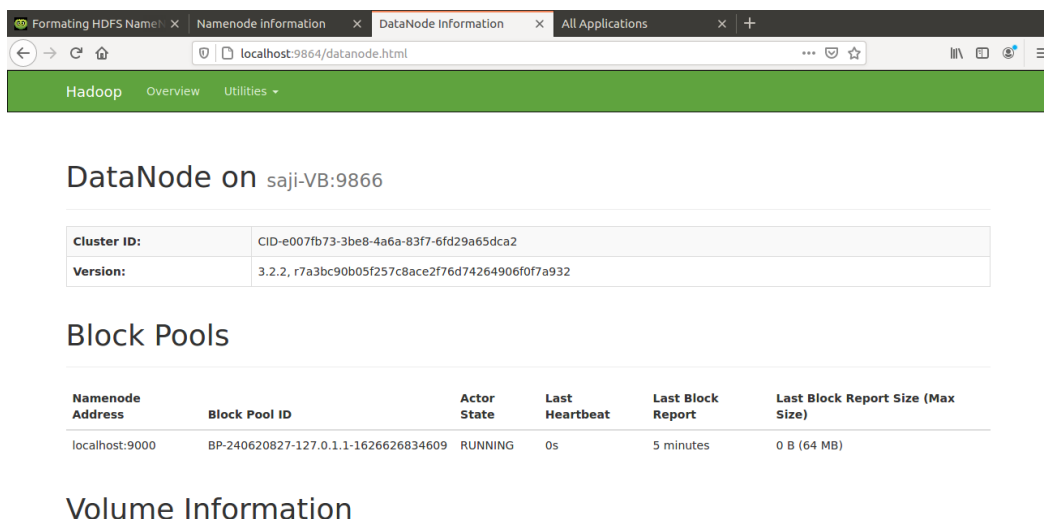


Figure 5 hadoop Data Node in browser

## 6. The YARN Resource Manager is accessible on port 8088

`http://localhost: 8088`

The Resource Manager is an invaluable tool that allows you to monitor all running processes in your Hadoop cluster.

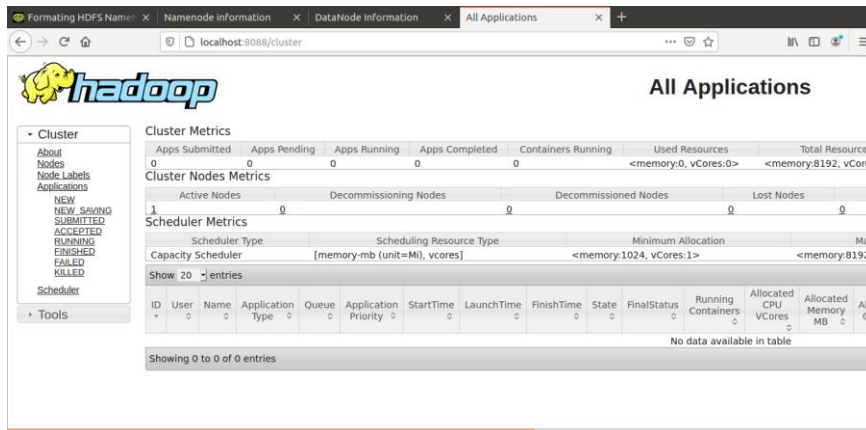


Figure 6 YARN Resource Manager in browser

## Running Word count Example

### 1. Listing Files in HDFS

After Starting HDFS. Listing Files in HDFS Using 'ls' command in the terminal, you can get a list of files in a directory and the status of a file.

```
hadoop-3.2.2/bin/hadoop fs -ls <args>
```

### 2. Inserting Data into HDFS

The following steps used to insert the required file in the Hadoop File System. Follow these steps one by one according to the given order.

#### Step 01 - Create an input directory

##### 1. Create User directory

Before create an input directory inside user directory following command used to use to create user directory

```
$ hdfs dfs -mkdir /user
```

```
hadoop@sajil-VB:~$ hdfs dfs -mkdir /user
```

Figure 7 create user directory

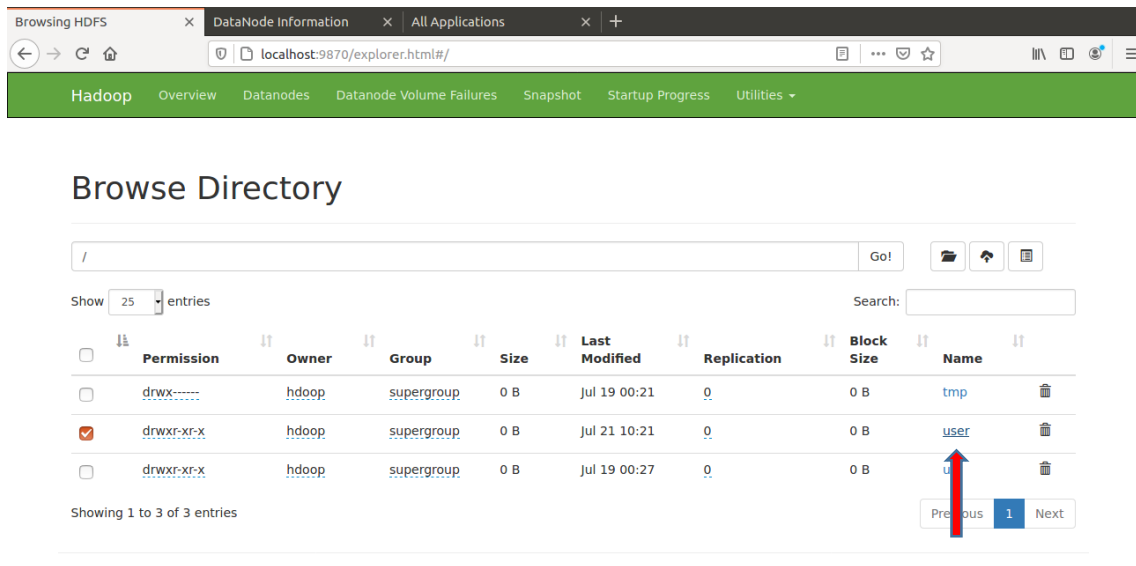


Figure 8 user directory

## 2. Create input directory

Following this command use to create input directory inside user directory

```
$ hdfs dfs -mkdir /user/input
```

```
hadoop@saji-VB:~$ hdfs dfs -mkdir /user/input
```

Figure 9 create user input directory

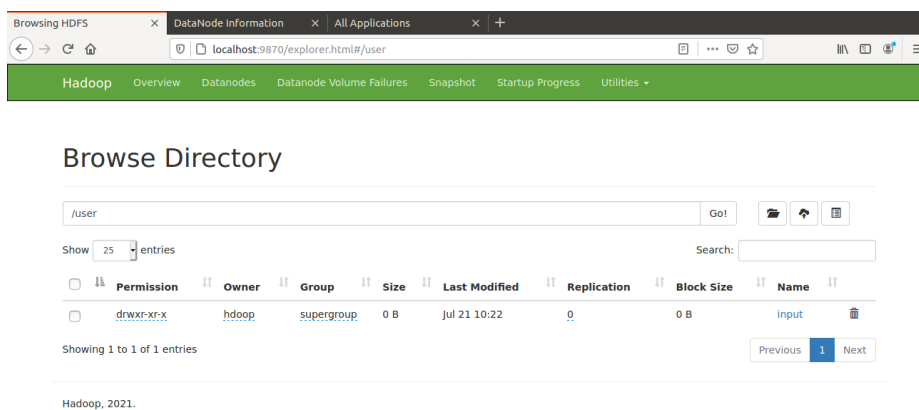


Figure 10 input directory

## 3. Download input text file (4300-0.txt)

```
$ wget http://www.gutenberg.org/files/4300/4300-0.txt
```

Using this command to download input text file and ls command to verify the downloading

```
hadoop@sajl-VB:~$ wget http://www.gutenberg.org/files/4300/4300-0.txt
--2021-07-19 00:02:25-- http://www.gutenberg.org/files/4300/4300-0.txt
Resolving www.gutenberg.org (www.gutenberg.org)... 152.19.134.47, 2610:28:309
0:3000:0:bad:cafe:47
Connecting to www.gutenberg.org (www.gutenberg.org)|152.19.134.47|:80... conn
ected.
HTTP request sent, awaiting response... 302 Found
Location: https://www.gutenberg.org/files/4300/4300-0.txt [following]
--2021-07-19 00:02:28-- https://www.gutenberg.org/files/4300/4300-0.txt
Connecting to www.gutenberg.org (www.gutenberg.org)|152.19.134.47|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 1586336 (1.5M) [text/plain]
Saving to: '4300-0.txt'

4300-0.txt          100%[=====>] 1.51M  510KB/s  in 3.0s

2021-07-19 00:02:34 (510 KB/s) - '4300-0.txt' saved [1586336/1586336]

hadoop@sajl-VB:~$ ls
4300-0.txt  examples.desktop  hadoop-3.2.2.tar.gz  tmpdata
dfsdata     hadoop-3.2.2      hadoop-3.2.2.tar.gz.1
hadoop@sajl-VB:~$
```

Figure 11 Download input text file

**Step 02 - transfer and save data files from local systems to the HDFS using the put command.**

```
hadoop@sajl-VB:~$ hdfs dfs -put /home/hadoop/4300-0.txt /user/input
```

Figure 12 put command

**Verify the file**

```
$ hdfs dfs -ls /user/input
```

Then use “ls” command to verify the file. There is a input file (4300-0.txt) inside user/input directory.

```
hadoop@sajl-VB:~$ hdfs dfs -ls /user/input
Found 1 items
-rw-r--r-- 1 hadoop supergroup 1586336 2021-07-21 10:29 /user/input/4300-0.txt
hadoop@sajl-VB:~$
```

Figure 13 file verification

The screenshot shows the Hadoop web interface for browsing HDFS. The browser address bar shows 'localhost:9870/explorer.html#/user/input'. The page title is 'Browse Directory'. Below the title, there is a search bar and a 'Go!' button. The main content area displays a table of files in the directory. The table has columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. A single file is listed: 4300-0.txt, owned by hadoop, with a size of 1.51 MB and a replication factor of 1. The file is highlighted in blue. At the bottom of the page, it says 'Hadoop, 2021.'

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hadoop	supergroup	1.51 MB	Jul 21 10:29	1	128 MB	4300-0.txt

Figure 14 created file inside /user/input directory

### 3. Running Word count Command

Running Word count Command Use the following commands to perform the word count Map Reduce example. The commands below will read all files from the input folder and run them through the Map Reduce jar file. The output directory will be created after the task has been successfully completed.

1. navigate to home directory (ex: \$ cd hadoop-3.2.2/)

```
hadoop@saji-VB:~$ cd hadoop-3.2.2/
```

Figure 15 Home directory

2. perform the word count Map Reduce programme

```
$ hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.2.jar wordcount /user/input /user/output
```

```
hadoop@saji-VB:~/hadoop-3.2.2$ hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.2.jar wordcount /user/input /user/output
2021-07-21 10:40:08,147 INFO client.RMPProxy: Connecting to ResourceManager at /127.0.0.1:8032
2021-07-21 10:40:09,747 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1626842636130_0001
```

Figure 16 2.perform the word count Map Reduce program

### 4. Show Results

Using the following command, verify the names of the result files produced under the hadoop3.2.2@/user/output file system.

```
hadoop@saji-VB:~/hadoop-3.2.2$ hdfs dfs -ls /user/output
Found 2 items
-rw-r--r-- 1 hadoop supergroup 0 2021-07-21 10:41 /user/output/_SUCCESS
-rw-r--r-- 1 hadoop supergroup 530493 2021-07-21 10:41 /user/output/part-r-000000
```

Figure 17 file name verification in result file

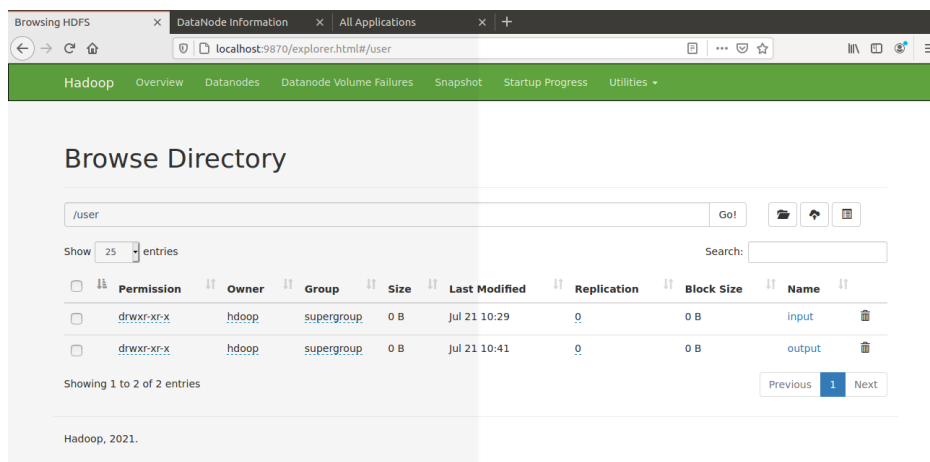


Figure 18 created output file

Show the content of the result file, which contains the word count result. Each word's count will be visible.

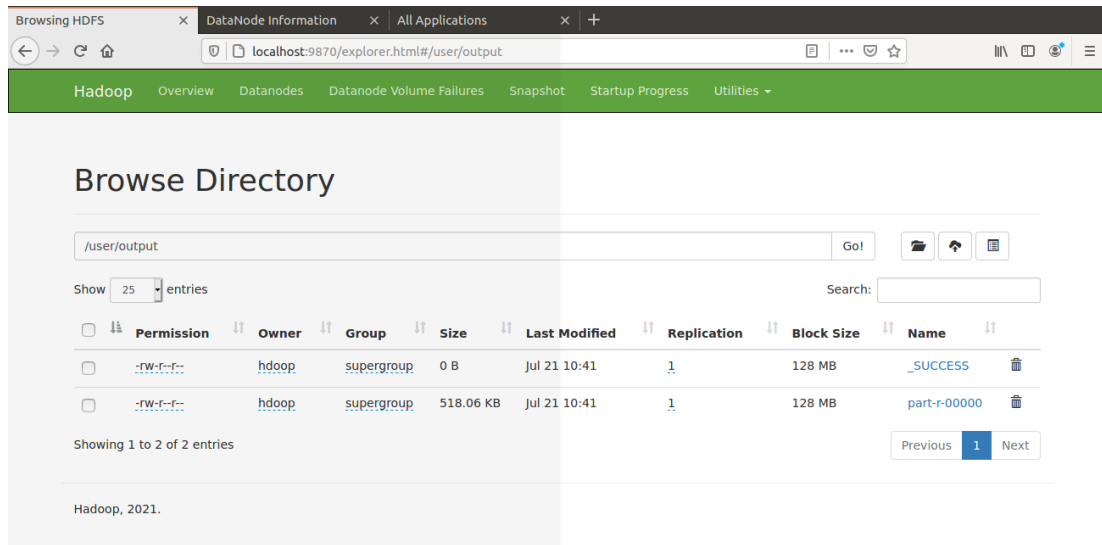


Figure 19 output success

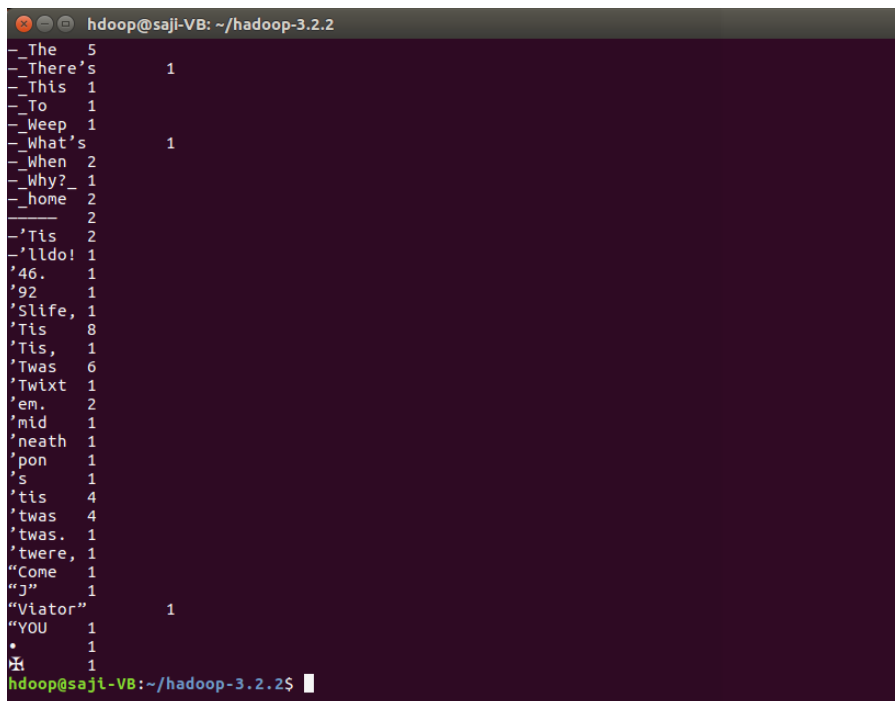


Figure 20 word count