

### Experiment : 3.1

**Name:** Himanshu Singla

**UID:** 21BCS10379

**Branch:** BE-CSE

**Section/Group:** CC-FL-604-A

**Semester:** 6<sup>th</sup>

**Date of Performance:** 10-04-2024

**Subject Name:** Cloud Computing and Distributed System

**Subject Code:** 21CSP-378

**Aim:** Install Hadoop single node cluster and run simple applications like word count.

Hadoop framework is well comfortable in the Linux environment but for the users who are not familiar with Linux environment but want to use the hadoop framework can be make use of this article. This article is aim to Install hadoop single node cluster and run simple application like wordcount.

### **Procedure:**

1. Install Java
2. Configure and install hadoop
3. Test hadoop installation
4. Create wordcount program
5. Input file to mapreduce
6. Display the output

#### **I. JAVA Installation**

1. Go to official Java Downloading page  
<https://www.oracle.com/java/technologies/javase-jre8-downloads.html> 1.  
After downloading java, run the **jdk-8u241-windows-x64.exe** file
2. Follow the instructions and click next.
3. After finishing the installation it is need to set Java environment variable
4. Go to Start->Edit the System environment variable->Environment variable
5. Then Click new and enter variable name as “JAVA\_HOME”
6. In the value field Enter the java path such as  
“C:\Java\jdk1.8.0\_241”(Consider your installation folder)

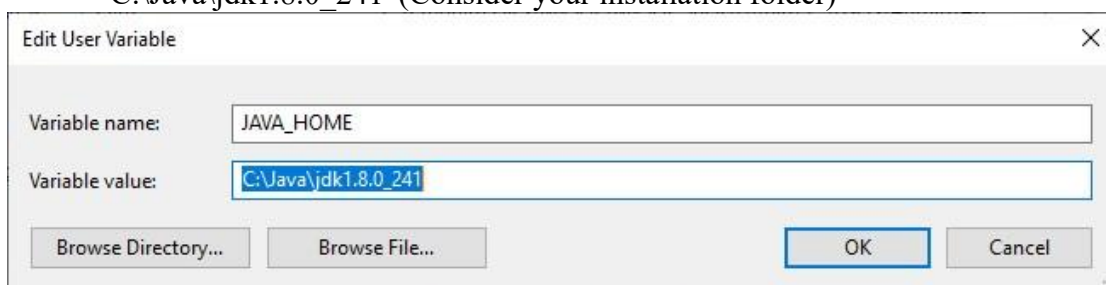


Fig-3.1

7. Go to path and click edit then type “%JAVA\_HOME%\bin”



Fig-3.2

8 . Then click Ok and Go to Command Prompt

9. Type “Java -version”. If it prints the installed version of java, now java successfully installed in your System.

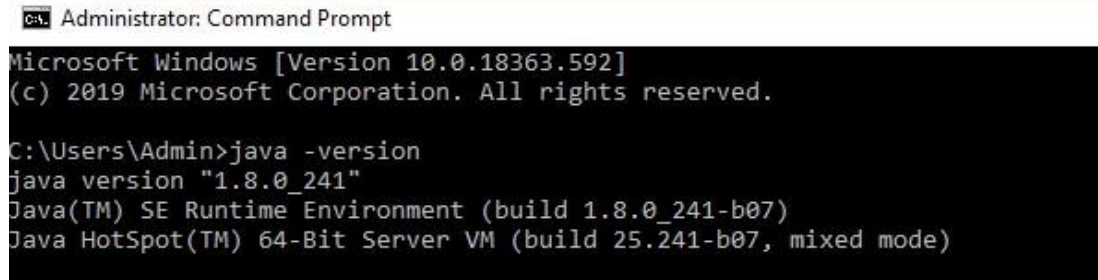


Fig-3.3

## II Configuring And Installing Hadoop

1. Download Hadoop 2.8.0 from

<http://archive.apache.org/dist/hadoop/core/hadoop-2.8.0/hadoop-2.8.0.tar.gz>) 2.

Extract the tar file ( in my case I used **7-zip** to extract the file and I stored the extracted file in the **D:\hadoop**)

3. After finishing the extraction it is need to set Hadoop environment variable

4. Go to Start->Edit the System environment variable->Environment variable

5. Then Click new and enter variable name as “HADOOP\_HOME”

6. In the value field Enter the java path such as “D:\hadoop”(Consider your installation folder)

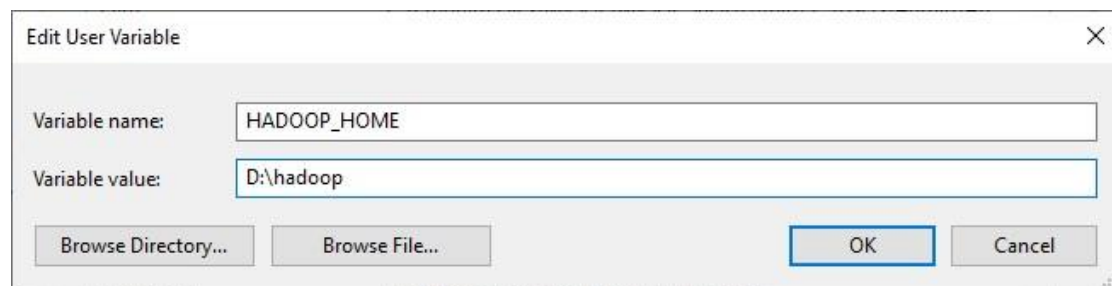


Fig-3.4

7. Go to path and click edit then type “%HADOOP\_HOME%\bin”



Fig-3.6

8. Now we have to configure the hadoop.

9. Go to D:/hadoop/etc/hadoop/.. folder, find the below mentioned files and paste the following.

### i. core-site.xml



```
<configuration>  <property>  <name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>  </property> </configuration>
```

ii. Rename " **mapred-site.xml.template** " to " **mapred-site.xml** " and edit this file D:\Hadoop/etc/hadoop/mapred-site.xml, paste below xml paragraph and save this file.

```
<configuration>  <property> &https://www.linkedin.com/redir/phishing-
page?url=lt%3Bname%26gt%3Bmapreduce%2eframework%2ename</name>
<value>yarn</value>  </property>
</configuration>
```

iii. Create folder "data" under "D:\Hadoop"

- Create folder "datanode" under "D:\Hadoop\data"
- Create folder "namenode" under "D:\Hadoop\data" data iv.

## hdfs-site.xml

```
<configuration>  <property>  <name>dfs.replication</name>
<value>1</value>  </property>  <property>
<name>dfs.namenode.name.dir</name>
<value>D:\hadoop\data\namenode</value>  </property>  <property>
<name>dfs.datanode.data.dir</name>
<value>D:\hadoop\data\datanode</value>  </property>
</configuration>
```

## v. yarn-site.xml

```
<configuration>  <property>  <name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>  </property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>  </property>
</configuration>
```

vi. Edit file D:\Hadoop/etc/hadoop/**hadoop-env.cmd** by closing the command line "JAVA\_HOME=%JAVA\_HOME%" instead of set "JAVA\_HOME= C:\Java\jdk1.8.0\_241" (if your java file in Program Files the instead of give **Progra~1** otherwise you will get JAVA\_HOME incorrectly set error)

vii. Download file Hadoop

Configuration.zip <https://github.com/Prithiviraj2503/hadoop-installation-windows>

viii. Delete file bin on D:\Hadoop\bin and replace it by the bin file of Downloaded configuration file (from Hadoop Configuration.zip).

ix. Open cmd and typing command "**hdfs namenode – format** ". You will see through command prompt which tasks are processing, after completion you will get a message like namenode format successfully and shutdown message

### III. Testing Hadoop Installation

#### 1. Open Cmd and type the following “Hadoop -version”

```
C:\Users\Admin>hadoop -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)
```

Fig-3.7

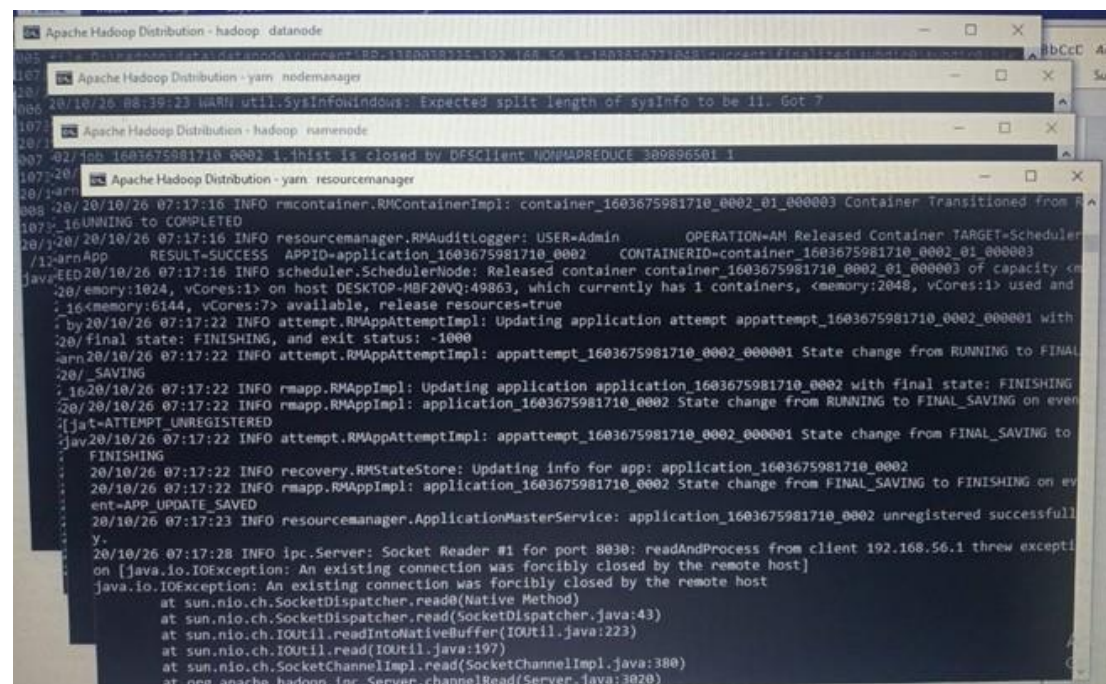
#### 2. To start the hadoop locate to “D:\hadoop\sbin” via command prompt and press **start-all.cmd**

```
Administrator: Command Prompt

C:\Users\Admin>D:
D:\>cd hadoop\sbin
D:\hadoop\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons
```

Fig-3.8

Now, you can see the namenode, datanode and yarn engines getting start,



The screenshot shows several overlapping command prompt windows displaying Hadoop logs. The logs include messages from the namenode, datanode, and yarn components, indicating the successful start and operation of the Hadoop distributed system. Key messages include 'INFO rmcontainer.RMContainerImpl: container\_1603675981710\_0002\_01\_000003 Container Transitioned from RUNNING to COMPLETED' and 'INFO resourcemanager.RMAuditLogger: USER=Admin OPERATION=AM Released Container TARGET=Scheduler'.

Fig-3.9

#### 3. Now type “jps”. JPS (Java Virtual Machine Process Status Tool) is a command is used to check all the Hadoop daemons like NameNode, DataNode, ResourceManager, NodeManager etc.



```
O:\hadoop\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

O:\hadoop\sbin>jps
5296 NameNode
2372 Jps
9192 ResourceManager
10140 NodeManager
9420 DataNode
```

Fig-3.10

4. Open: <http://localhost:8088> in any browser

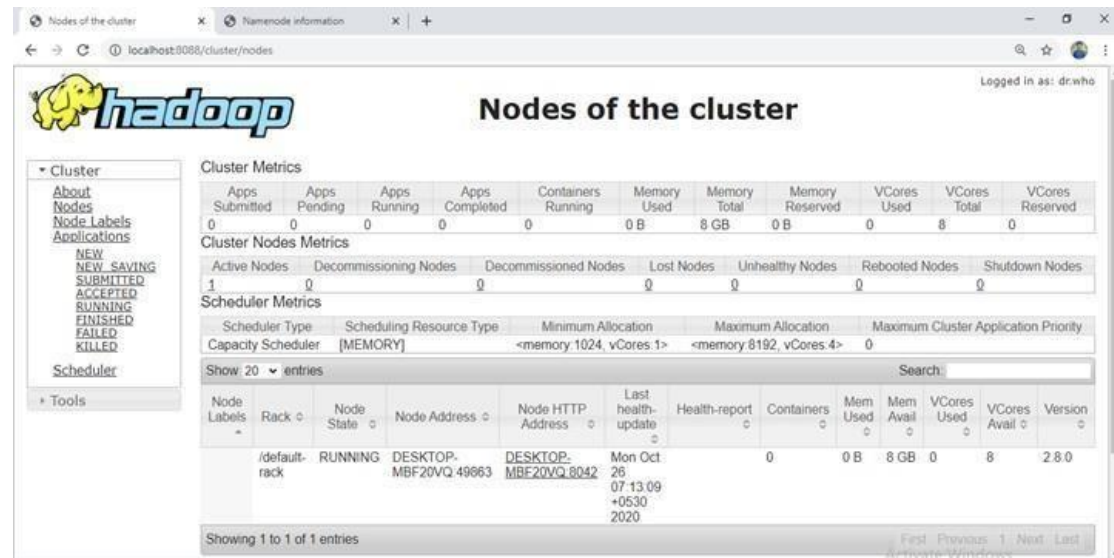


Fig-3.11

5. Open: <http://localhost:50070> in any browser

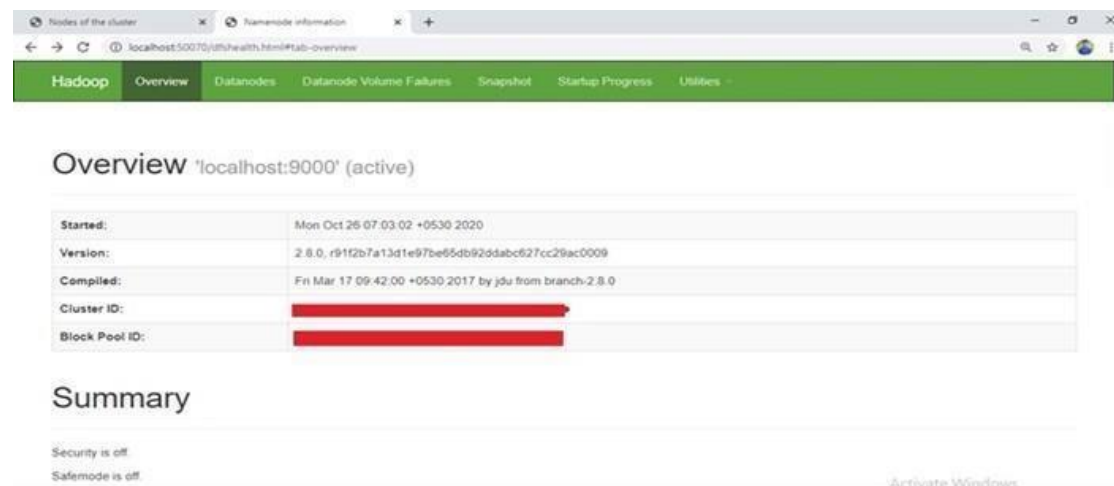


Fig-3.12

Now hadoop succesfully installed in your System.

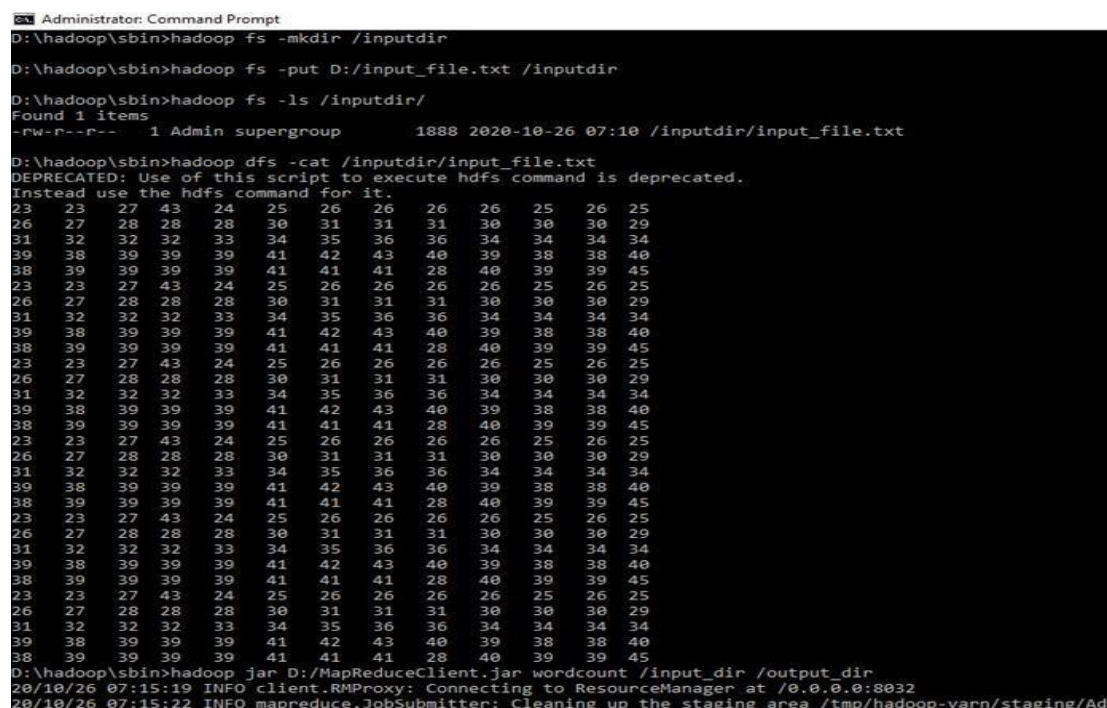
## IV. Simple WordCount Program

- 1) After successful hadoop installation we need to create an directory in the hadoop file system
- 2) Start the hadoop via command prompt \$ **start-all.cmd**

- 3) By using **\$jps** command Ensure hadoop nodes are running
- 4) To create a directory, use: **\$ hadoop fs -mkdir /inputdir**
- 5) To input a file within a directory, use: **\$ hadoop fs -put D:/input\_file.txt/inputdir**
- 6) To ensure whether your file successfully imported, use: **\$ hadoop fs -ls /inputdir/**

- 7) To view the content of the file, use: **\$ hadoop dfs -cat /inputdir/input\_file.txt**

Link for input file : <https://github.com/Prithiviraj2503/hadoop-installation-windows>



```
Administrator: Command Prompt
D:\hadoop\sbin>hadoop fs -mkdir /inputdir

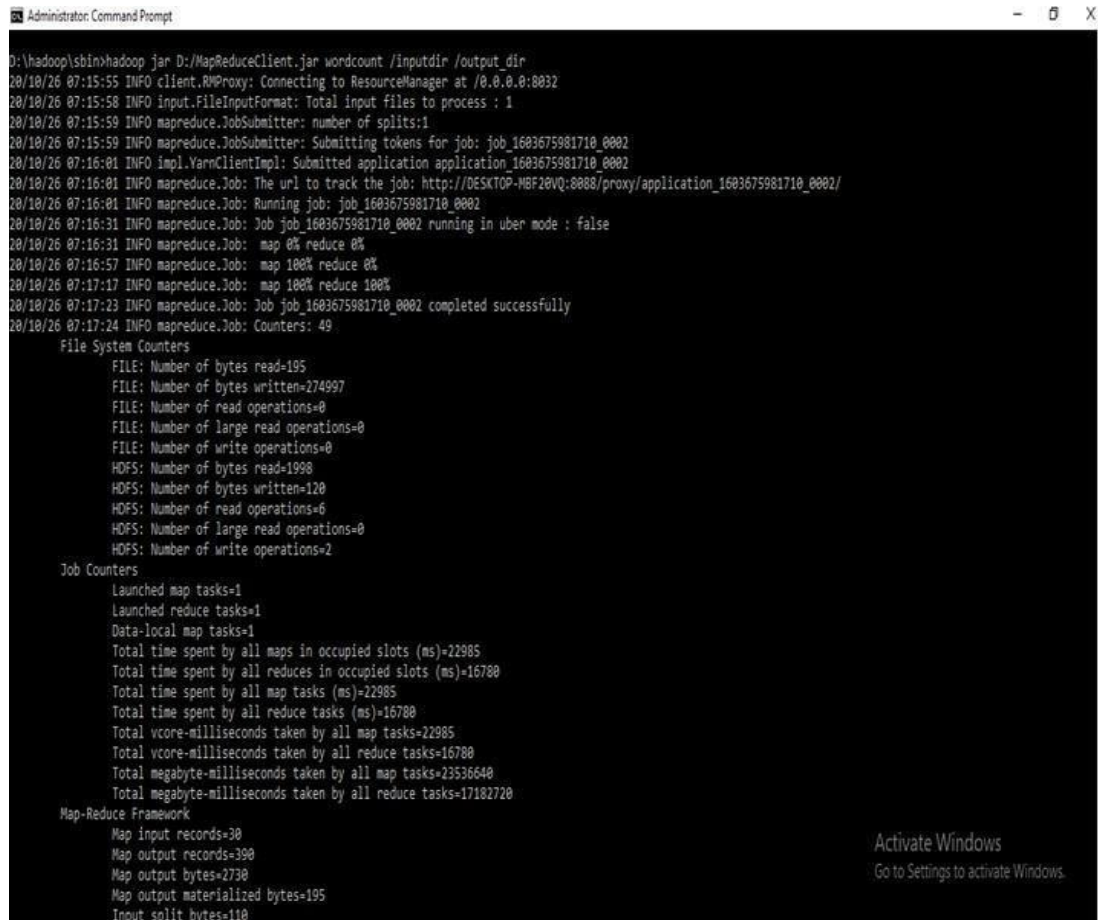
D:\hadoop\sbin>hadoop fs -put D:/input_file.txt /inputdir

D:\hadoop\sbin>hadoop fs -ls /inputdir/
Found 1 items
-rw-r--r-- 1 Admin supergroup          1888 2020-10-26 07:10 /inputdir/input_file.txt

D:\hadoop\sbin>hadoop dfs -cat /inputdir/input_file.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
23 23 27 43 24 25 26 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
23 23 27 43 24 25 26 26 26 26 25 26 25
26 27 28 28 28 30 31 31 31 30 30 30 29
31 32 32 32 33 34 35 36 36 34 34 34 34
39 38 39 39 39 41 42 43 40 39 38 38 40
38 39 39 39 39 41 41 41 28 40 39 39 45
D:\hadoop\sbin>hadoop jar D:/MapReduceClient.jar wordcount /input_dir /output_dir
20/10/26 07:15:19 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/10/26 07:15:22 INFO mapreduce.JobSubmitter: Cleaning up the staging area /tmp/hadoop-yarn/staging/Adm
```

Fig-3.13

- 8) Now apply mapreduce program to the input file. We have a **mapReduceClient.jar** which contain java mapper and reducer programs. After applying the jar file you can see the task performed in the mapreduce phase. All the results of completed tasks will be printed in the command prompt. Link for mapReduceClient.jar : <https://github.com/Prithiviraj2503/hadoop-installation-windows>



```
Administrator: Command Prompt

D:\hadoop\sbin>hadoop jar D:\MapReduceClient.jar wordcount /inputdir /output_dir
20/10/26 07:15:55 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/10/26 07:15:58 INFO input.FileInputFormat: Total input files to process : 1
20/10/26 07:15:59 INFO mapreduce.JobSubmitter: number of splits:1
20/10/26 07:15:59 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1603675981710_0002
20/10/26 07:16:01 INFO impl.YarnClientImpl: Submitted application application_1603675981710_0002
20/10/26 07:16:01 INFO mapreduce.Job: The url to track the job: http://DESKTOP-WBF20VQ:8088/proxy/application_1603675981710_0002/
20/10/26 07:16:01 INFO mapreduce.Job: Running job: job_1603675981710_0002
20/10/26 07:16:31 INFO mapreduce.Job: Job job_1603675981710_0002 running in uber mode : false
20/10/26 07:16:31 INFO mapreduce.Job: map 0% reduce 0%
20/10/26 07:16:57 INFO mapreduce.Job: map 100% reduce 0%
20/10/26 07:17:17 INFO mapreduce.Job: map 100% reduce 100%
20/10/26 07:17:23 INFO mapreduce.Job: Job job_1603675981710_0002 completed successfully
20/10/26 07:17:24 INFO mapreduce.Job: Counters: 49

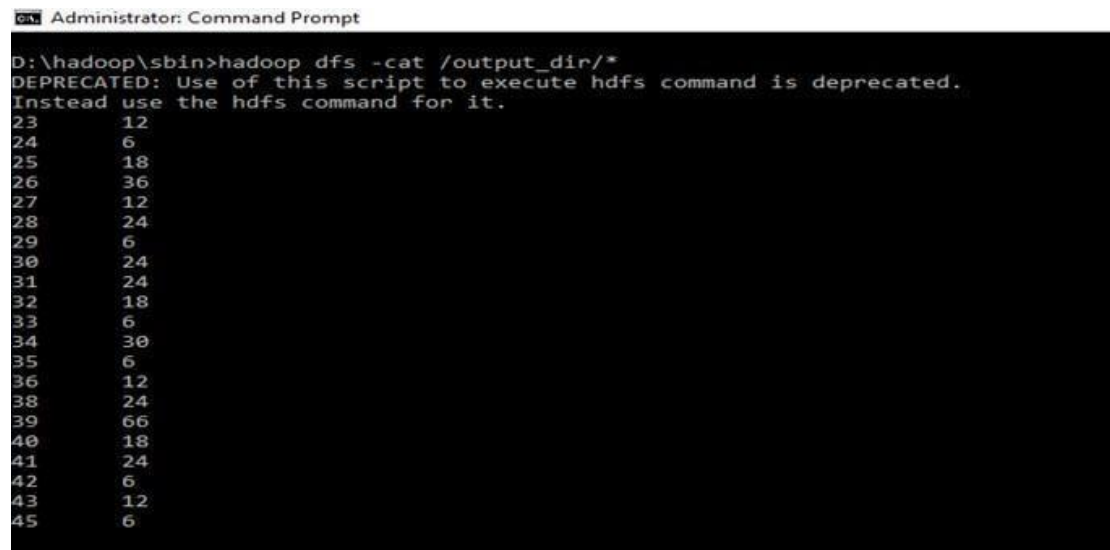
File System Counters
  FILE: Number of bytes read=195
  FILE: Number of bytes written=274997
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=1998
  HDFS: Number of bytes written=120
  HDFS: Number of read operations=6
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=2

Job Counters
  Launched map tasks=1
  Launched reduce tasks=1
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=22985
  Total time spent by all reduces in occupied slots (ms)=16780
  Total time spent by all map tasks (ms)=22985
  Total time spent by all reduce tasks (ms)=16780
  Total vcore-milliseconds taken by all map tasks=22985
  Total vcore-milliseconds taken by all reduce tasks=16780
  Total megabyte-milliseconds taken by all map tasks=23536640
  Total megabyte-milliseconds taken by all reduce tasks=17182720

Map-Reduce Framework
  Map input records=30
  Map output records=390
  Map output bytes=2730
  Map output materialized bytes=195
  Input split bytes=110
```

Fig-3.14

- 9) After completed the mapreduce tasks the output will be stored in the **output\_dir** directory To see the output, use: **\$ hadoop dfs -cat /output\_dir/**



```
Administrator: Command Prompt

D:\hadoop\sbin>hadoop dfs -cat /output_dir/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
23      12
24      6
25      18
26      36
27      12
28      24
29      6
30      24
31      24
32      18
33      6
34      30
35      6
36      12
38      24
39      66
40      18
41      24
42      6
43      12
45      6
```

Fig-3.15

- 10) To stop the hadoop type **\$stop-all.cmd**

Now the hadoop single node cluster was installed successfully and the simple word count program were executed successfully in your windows system.

```
D:\hadoop\sbin>stop-all.cmd
This script is Deprecated. Instead use stop-dfs.cmd and stop-yarn.cmd
SUCCESS: Sent termination signal to the process with PID 9340.
SUCCESS: Sent termination signal to the process with PID 10652.
stopping yarn daemons
SUCCESS: Sent termination signal to the process with PID 8576.
SUCCESS: Sent termination signal to the process with PID 11128.

INFO: No tasks running with the specified criteria.

D:\hadoop\sbin>
```

Fig-3.16

### Analysis:

- This provides a clear, step-by-step guide for installing and configuring Hadoop on a Windows system, along with running a basic WordCount program.
- It covers essential tasks such as setting up Java, configuring Hadoop, testing the installation, and executing the WordCount program.
- The instructions are detailed, including screenshots for clarity.
- However, it could benefit from explanations of Hadoop concepts, troubleshooting tips, and considerations for security.
- Overall, it's a useful resource for beginners aiming to set up Hadoop on Windows.

### Conclusion:

- In this experiment, we installed and ran Hadoop on a Windows environment, complete with executing a simple WordCount program.
- By following the detailed instructions provided, users can successfully set up their Hadoop single-node cluster and perform basic MapReduce tasks.
- While the guide covers essential steps and includes helpful visuals, there's room for improvement in terms of explaining Hadoop concepts, offering troubleshooting guidance, and addressing security considerations.
- Nonetheless, it serves as a valuable resource for beginners seeking to explore Hadoop in a Windows setting.

### Result:

- Installed and ran Hadoop on Windows, including executing a WordCount program, and explained in depth the concepts and addressing potential issues.