

*Laboratory Assignment File*  
*for*

# **Big Data Analytics (CS0552)**

**Master of Technology**  
in  
**Computer Science & Engineering**

**Submitted by**

**Yugal**  
(Roll no. 25903053)

**Submitted to**

**Dr. Shveta Mahajan**  
(Assistant Professor, NIT Jalandhar)



**Department of Computer Science & Engineering**  
**Dr. B R Ambedkar National Institute of Technology Jalandhar**  
**Punjab, India-144008**  
**May, 2026**

## Contents

<b>Assignment 1: Install Apache Hadoop and Setup Single Node Cluster(27-01-2026)</b>	<b>1</b>
1.1 Step 1: Installing Java . . . . .	1
1.2 Step 2: Setup SSH . . . . .	2
1.3 Step 3: Download and Configure Hadoop . . . . .	3
1.4 Step 4: Environment Variable Configuration . . . . .	4
1.5 Step 5: Hadoop XML Configuration Files . . . . .	5
1.6 Step 6: Format NameNode & Start Hadoop Services . . . . .	10
1.7 Step 7: HDFS Operations . . . . .	11
1.8 Result . . . . .	14
1.9 Conclusion . . . . .	14
<b>Assignment 2: Load large datasets into HDFS and analyze block distribution. (03-02-2026)</b>	<b>15</b>
2.1 Step 1: Collect large Dataset (CSV/TXT) . . . . .	15
2.2 Step 2: Create HDFS directory. . . . .	15
2.3 Step 3: Upload large dataset into HDFS . . . . .	15
2.4 Step 4: Verify data storage . . . . .	16
2.5 Step 5: Analyze block distribution . . . . .	16
2.6 Step 6: Download processed dataset from HDFS to local system . . . . .	16
2.7 Step 7: Delete data from HDFS . . . . .	16
2.8 Step 8: Check file block information . . . . .	17
2.9 Result . . . . .	17
2.10 Conclusion . . . . .	17
<b>Assignment 3: Implement MapReduce programs for Data Processing. (10-02-2026)</b>	<b>18</b>
3.1 Step 1: Start Hadoop Services . . . . .	18
3.2 Step 2: Create Input Directory in HDFS . . . . .	18
3.3 Step 3: Write MapReduce Program . . . . .	19
3.4 Step 4: Compile Program and Create JAR . . . . .	21
3.5 Step 5: Execute MapReduce Job . . . . .	21
3.6 Step 6: Check Output Directory in HDFS . . . . .	21
3.7 Step 7: Display Result File . . . . .	22
3.8 Step 8: Modify Input and Re-run Job . . . . .	22
3.9 Result . . . . .	23
3.10 Conclusion . . . . .	23

## List of Figures

1	System Update Command . . . . .	1
2	Java Installation . . . . .	1
3	Java Version and Path Verification . . . . .	2
4	Install SSH Server . . . . .	2
5	Generate SSH Key Pair . . . . .	3
6	Create Hadoop User . . . . .	3
7	Download and Extract Hadoop . . . . .	4
8	Moving Hadoop . . . . .	4
9	.bashrc Modification . . . . .	5
10	hadoop-env.sh Modification . . . . .	6
11	core-site.xml Modification . . . . .	7
12	hdfs-site.xml Modification . . . . .	8
13	Creating Folder for NameNode and DataNode . . . . .	8
14	mapred-site.xml Modification . . . . .	9
15	yarn-site.xml Modification . . . . .	9
16	Format NameNode . . . . .	10
17	Start Hadoop Services . . . . .	10
18	Start YARN Services . . . . .	11
19	HDFS Operation 1 . . . . .	12
20	HDFS Operation 2 . . . . .	12
21	HDFS Result 1 . . . . .	13
22	HDFS Result 2 . . . . .	13
23	Start Hadoop Services and Verify with jps . . . . .	15
24	Created exceldata directory in hdfs and copied dataset . . . . .	15
25	Upload large dataset into HDFS . . . . .	15
26	Verify data storage in HDFS . . . . .	16
27	Analyze block distribution in HDFS . . . . .	16
28	Download processed dataset from HDFS to local system . . . . .	16
29	Delete data from HDFS . . . . .	16
30	Check file block information in HDFS . . . . .	17
31	Starting Hadoop services and verifying with jps . . . . .	18
32	Creating input directory and uploading dataset . . . . .	19
33	Mapper class implementation . . . . .	19
34	Reducer class implementation . . . . .	20
35	Driver class implementation . . . . .	20
36	Compilation and JAR creation . . . . .	21
37	Executing MapReduce job . . . . .	21

38	Output directory created in HDFS . . . . .	21
39	Displaying MapReduce output . . . . .	22
40	Updating input text file . . . . .	22
41	Re-running MapReduce with modified dataset . . . . .	22
42	New output after re-running MapReduce job . . . . .	23

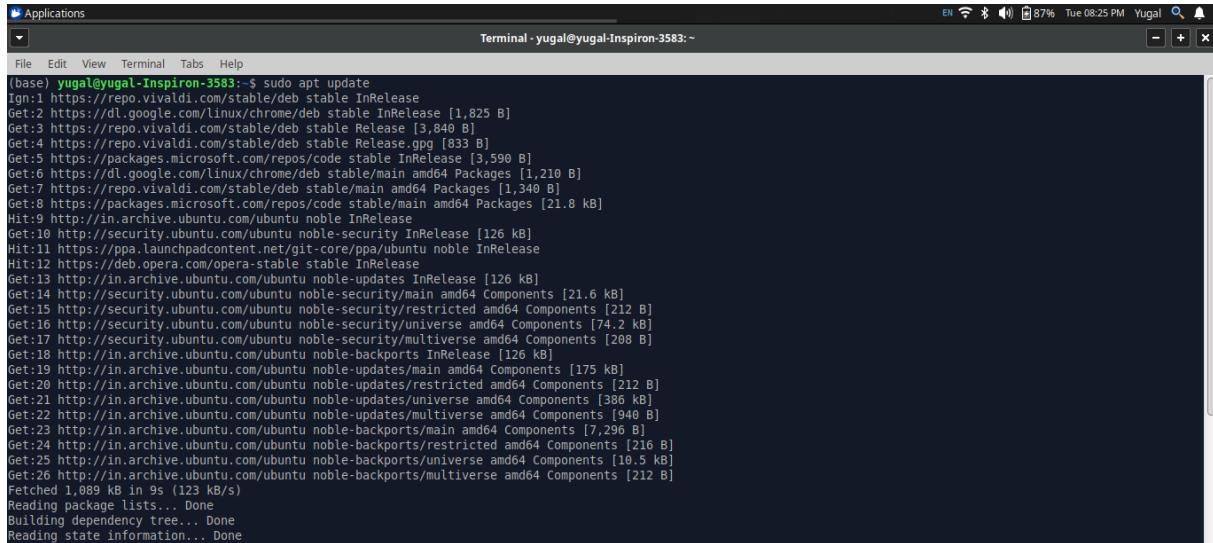
# Assignment 1: Install Apache Hadoop and Setup Single Node Cluster

**Objective:** To install and configure Apache Hadoop in pseudo-distributed mode and perform basic HDFS operations such as upload, delete, replication check, and permission handling.

## 1.1 Step 1: Installing Java

### Updating System

```
sudo apt update
```

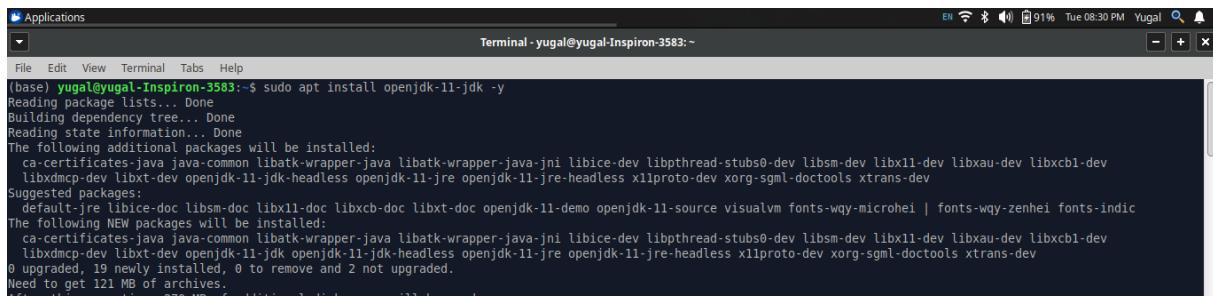


```
(base) yugal@yugal-Inspiron-3583:~$ sudo apt update
Ign:1 https://repo.vivaldi.com/stable/deb stable InRelease
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:3 https://repo.vivaldi.com/stable/deb stable Release [3,840 B]
Get:4 https://repo.vivaldi.com/stable/deb stable Release.gpg [833 B]
Get:5 https://packages.microsoft.com/repos/code/stable/main amd64 Packages [1,210 B]
Get:6 https://repo.vivaldi.com/stable/deb stable/main amd64 Packages [1,340 B]
Get:7 https://packages.microsoft.com/repos/code/stable/main amd64 Packages [21.8 kB]
Hit:8 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:9 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:10 https://ppa.launchpadcontent.net/git-core/ppa/ubuntu noble InRelease
Hit:11 https://deb.opera.com/opera-stable stable InRelease
Get:12 https://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:15 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [74.2 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:17 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [175 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [212 B]
Get:20 http://in.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [386 kB]
Get:21 http://in.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [940 B]
Get:22 http://in.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [7,296 B]
Get:23 http://in.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:24 http://in.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [10.5 kB]
Get:25 http://in.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Fetched 1,089 kB in 9s (123 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Figure 1: System Update Command

### Installing Java

```
sudo apt install openjdk-11-jdk -y
```

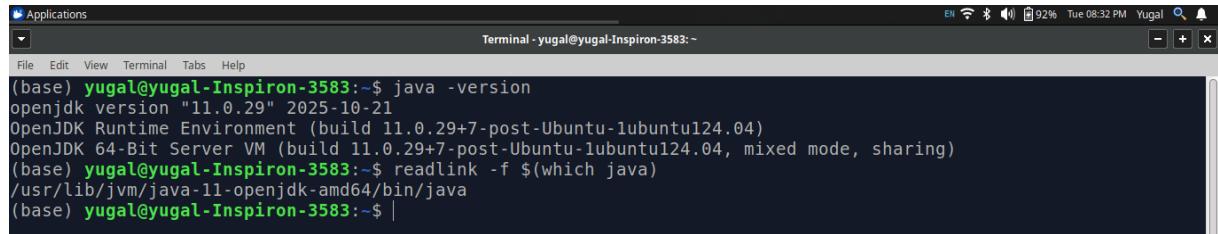


```
(base) yugal@yugal-Inspiron-3583:~$ sudo apt install openjdk-11-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java java-common libatk-wrapper-java libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev
  libxdmcp-dev libxt-dev openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo openjdk-11-source visualvm fonts-wqy-microhei | fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  ca-certificates-java java-common libatk-wrapper-java libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev
  libxdmcp-dev libxt-dev openjdk-11-jdk openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 19 newly installed, 0 to remove and 2 not upgraded.
Need to get 121 MB of archives.
```

Figure 2: Java Installation

## Java Version and Path Verification

```
java -version  
readlink -f $(which java)
```



The screenshot shows a terminal window titled "Terminal - yugal@yugal-Inspiron-3583:~". The window displays the output of two commands: "java -version" and "readlink -f \$(which java)". The output indicates that Java version 11.0.29 is installed, and the path to the Java executable is /usr/lib/jvm/java-11-openjdk-amd64/bin/java.

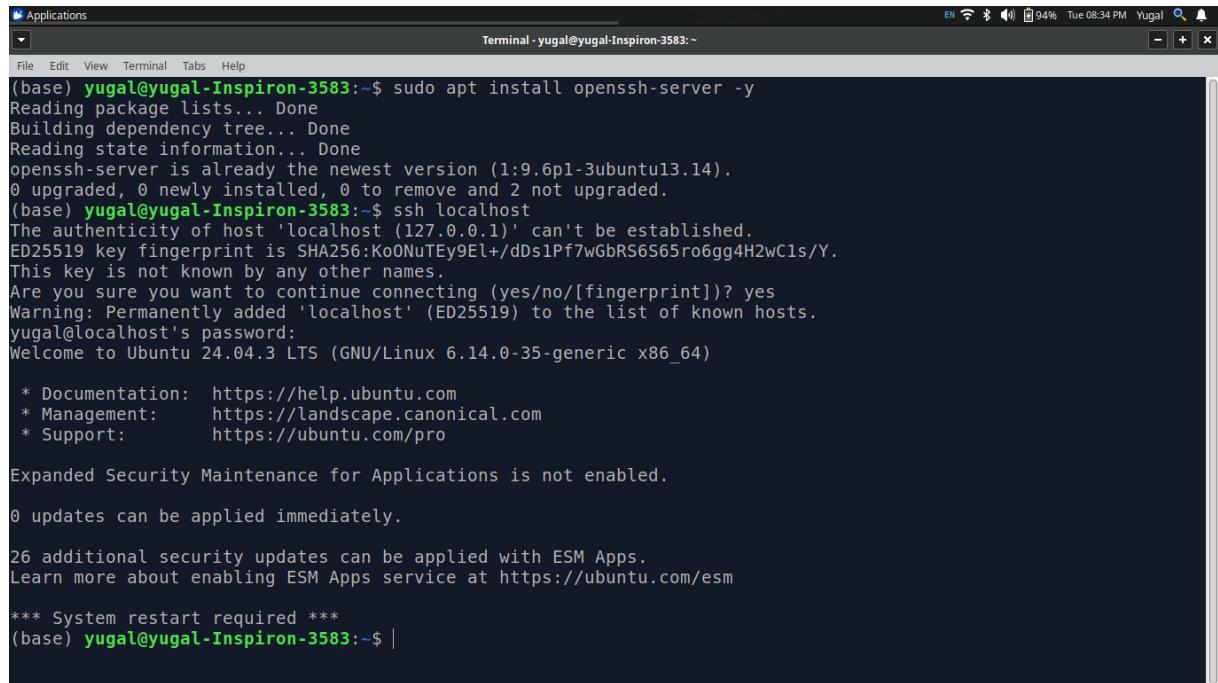
```
(base) yugal@yugal-Inspiron-3583:~$ java -version  
openjdk version "11.0.29" 2025-10-21  
OpenJDK Runtime Environment (build 11.0.29+7-post-Ubuntu-1ubuntu124.04)  
OpenJDK 64-Bit Server VM (build 11.0.29+7-post-Ubuntu-1ubuntu124.04, mixed mode, sharing)  
(base) yugal@yugal-Inspiron-3583:~$ readlink -f $(which java)  
/usr/lib/jvm/java-11-openjdk-amd64/bin/java  
(base) yugal@yugal-Inspiron-3583:~|
```

Figure 3: Java Version and Path Verification

## 1.2 Step 2: Setup SSH

### Install SSH Server

```
sudo apt install openssh-server -y  
ssh localhost
```



The screenshot shows a terminal window titled "Terminal - yugal@yugal-Inspiron-3583:~". The window displays the output of the command "sudo apt install openssh-server -y". It shows the package being installed, the fingerprint being checked, and the user being prompted to add the host to the list of known hosts. The user enters "yes" and provides their password. Finally, the user is welcomed to Ubuntu 24.04.3 LTS.

```
(base) yugal@yugal-Inspiron-3583:~$ sudo apt install openssh-server -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
openssh-server is already the newest version (1:9.6p1-3ubuntu13.14).  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
(base) yugal@yugal-Inspiron-3583:~$ ssh localhost  
The authenticity of host 'localhost (127.0.0.1)' can't be established.  
ED25519 key fingerprint is SHA256:KoONuTEy9El+/dDs1Pf7wGbRS6S65ro6gg4H2wC1s/Y.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'localhost' (ED25519) to the list of known hosts.  
yugal@localhost's password:  
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-35-generic x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/pro  
  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
  
26 additional security updates can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at https://ubuntu.com/esm  
  
*** System restart required ***  
(base) yugal@yugal-Inspiron-3583:~|
```

Figure 4: Install SSH Server

### Generate SSH Key Pair

```
ssh-keygen -t rsa -P ""  
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

```

ssh localhost
exit


```

Figure 5: Generate SSH Key Pair

### 1.3 Step 3: Download and Configure Hadoop

#### Create Hadoop User

```

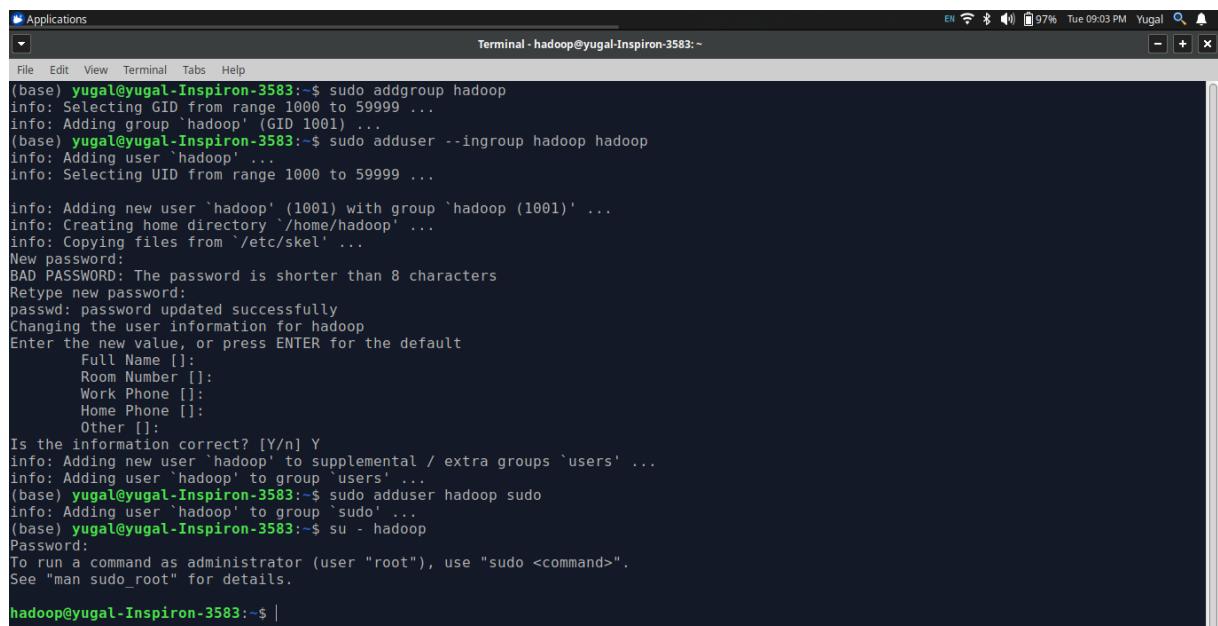
sudo addgroup hadoop
sudo adduser --ingroup hadoop hadoop
sudo adduser hadoop sudo
su - hadoop

```

Figure 6: Create Hadoop User

## Download and Extract Hadoop

```
exit  
sudo cp /home/yugal/Downloads/hadoop-3.3.0.tar.gz /home/hadoop/  
sudo chown hadoop:hadoop /home/hadoop/hadoop-3.3.0.tar.gz  
su - hadoop  
ls  
tar -xvzf hadoop-3.3.0.tar.gz
```

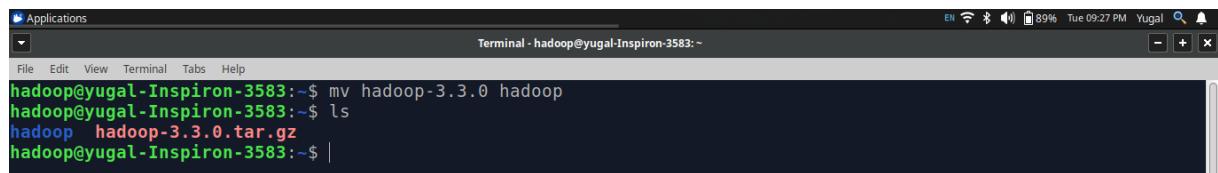


```
Applications Terminal - hadoop@yugal-Inspiron-3583:~  
File Edit View Terminal Tabs Help  
(base) yugal@yugal-Inspiron-3583:~$ sudo addgroup hadoop  
info: Selecting GID from range 1000 to 59999 ...  
info: Adding group `hadoop` (GID 1001) ...  
(base) yugal@yugal-Inspiron-3583:~$ sudo adduser --ingroup hadoop hadoop  
info: Adding user `hadoop` ...  
info: Selecting UID from range 1000 to 59999 ...  
info: Adding new user `hadoop` (1001) with group `hadoop (1001)` ...  
info: Creating home directory `/home/hadoop` ...  
info: Copying files from `/etc/skel` ...  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
passwd: password updated successfully  
Changing the user information for hadoop  
Enter the new value, or press ENTER for the default  
    Full Name []:  
    Room Number []:  
    Work Phone []:  
    Home Phone []:  
    Other []:  
Is the information correct? [Y/n] Y  
info: Adding new user `hadoop` to supplemental / extra groups `users` ...  
info: Adding user `hadoop` to group `users` ...  
(base) yugal@yugal-Inspiron-3583:~$ sudo adduser hadoop sudo  
info: Adding user `hadoop` to group `sudo` ...  
(base) yugal@yugal-Inspiron-3583:~$ su - hadoop  
Password:  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
hadoop@yugal-Inspiron-3583:~$ |
```

Figure 7: Download and Extract Hadoop

## Moving Hadoop

```
mv hadoop-3.3.0 hadoop  
ls
```



```
Applications Terminal - hadoop@yugal-Inspiron-3583:~  
File Edit View Terminal Tabs Help  
hadoop@yugal-Inspiron-3583:~$ mv hadoop-3.3.0 hadoop  
hadoop@yugal-Inspiron-3583:~$ ls  
hadoop hadoop-3.3.0.tar.gz  
hadoop@yugal-Inspiron-3583:~$ |
```

Figure 8: Moving Hadoop

## 1.4 Step 4: Environment Variable Configuration

### .bashrc Modification

```
nano ~/.bashrc
```

```
% Add the following lines at the end of the file
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME=/home/hadoop/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

The screenshot shows a desktop environment with a terminal window and a nano editor window. The terminal window shows the command `readlink -f $(which java)` and its output. The nano editor window shows the contents of the `.bashrc` file, which includes the provided export commands at the bottom.

```
hadoop@yugal-Inspiron-3583:~$ readlink -f $(which java)
/usr/lib/jvm/java-11-openjdk-amd64/bin/java
hadoop@yugal-Inspiron-3583:~$ nano ~/.bashrc|
```

```
File Edit View Terminal Tabs Help
GNU nano 7.2 /home/hadoop/.bashrc *

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME=/home/hadoop/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^H Replace ^U Paste ^J Justify ^/ Go To Line M-U Undo  
M-E Redo

Figure 9: .bashrc Modification

## 1.5 Step 5: Hadoop XML Configuration Files

### hadoop-env.sh Modification

```
source ~/.bashrc
echo $HADOOP_HOME
/home/hadoop/hadoop
cd $HADOOP_HOME/etc/hadoop
ls
nano hadoop-env.sh
% Add the following lines at the path
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

The screenshot shows two terminal windows side-by-side. The left terminal window is titled 'Terminal - hadoop@yugal-Inspiron-3583: ~/hadoop/etc/hadoop'. It displays a series of shell commands used to change directory to \$HADOOP\_HOME/etc/hadoop, source .bashrc, and echo \$HADOOP\_HOME. The right terminal window is also titled 'Terminal - hadoop@yugal-Inspiron-3583: ~/hadoop/etc/hadoop'. It shows the 'GNU nano 7.2' editor open on the 'hadoop-env.sh' file. The file contains configuration for Hadoop, including setting JAVA\_HOME to /usr/java/testing, defining default options for HDFS, and specifying the Java implementation (export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64). It also includes sections for Hadoop's location and configuration information.

```

hadoop@yugal-Inspiron-3583:~$ cd $HADOOP_HOME/etc/hadoop
-bash: cd: /etc/hadoop: No such file or directory
hadoop@yugal-Inspiron-3583:~$ source ~/.bashrc
hadoop@yugal-Inspiron-3583:~$ source ~/.bashrc
hadoop@yugal-Inspiron-3583:~$ echo $HADOOP_HOME
/home/hadoop/hadoop
hadoop@yugal-Inspiron-3583:~$ cd $HADOOP_HOME/etc/hadoop
hadoop@yugal-Inspiron-3583:~/hadoop/etc/hadoop$ nano hadoop-env.sh
hadoop@yugal-Inspiron-3583:~/hadoop/etc/hadoop$ |
```

```

GNU nano 7.2                               hadoop-env.sh *
#  JAVA_HOME=/usr/java/testing hdfs dfs -ls
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferable, modify this file accordingly.
#####
# Generic settings for HADOOP
#####
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

# Location of Hadoop. By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=

# Location of Hadoop's configuration information. i.e., where this
# file is living. If this is not defined, Hadoop will attempt to
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo  
 ^X Exit ^R Read File ^Y Replace ^U Paste ^J Justify ^L Go To Line M-E Redo

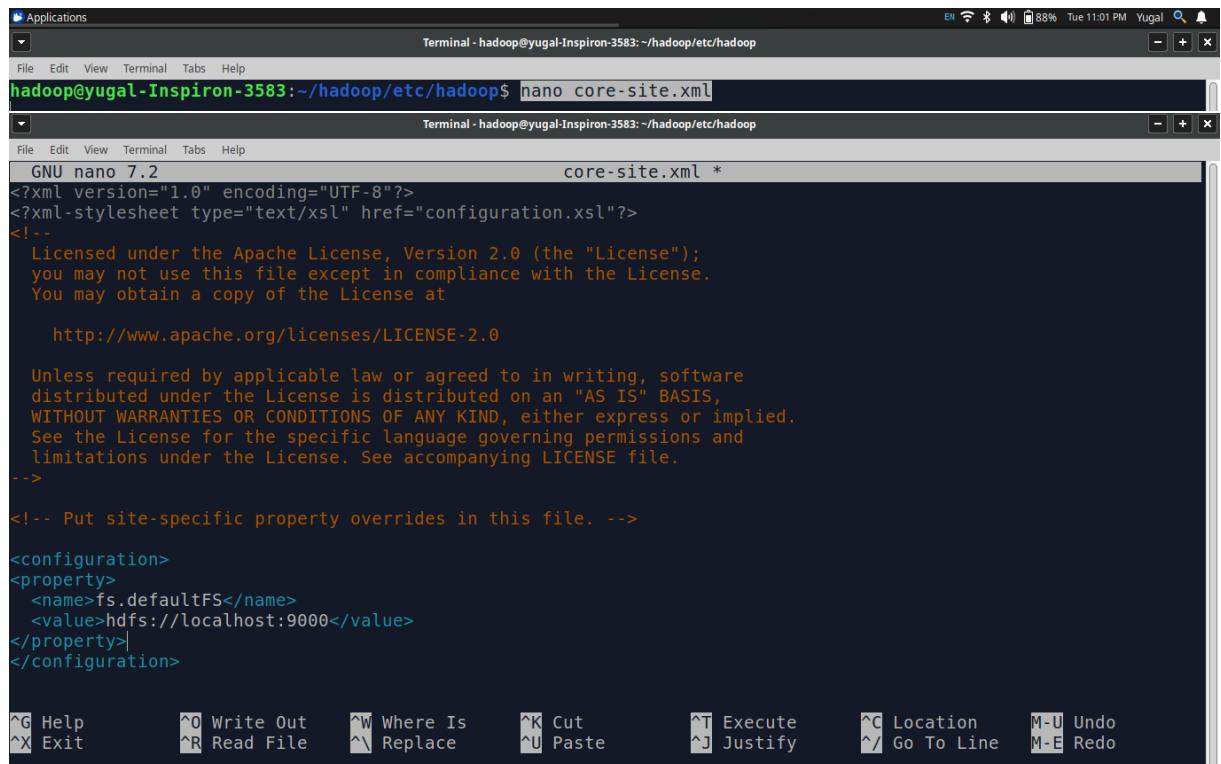
Figure 10: hadoop-env.sh Modification

## core-site.xml Modification

```

nano core-site.xml
% Add the following property inside <configuration> tag
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
```

```
</property>
```



```
hadoop@yugal-Inspiron-3583:~/hadoop/etc/hadoop$ nano core-site.xml
GNU nano 7.2
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

Figure 11: core-site.xml Modification

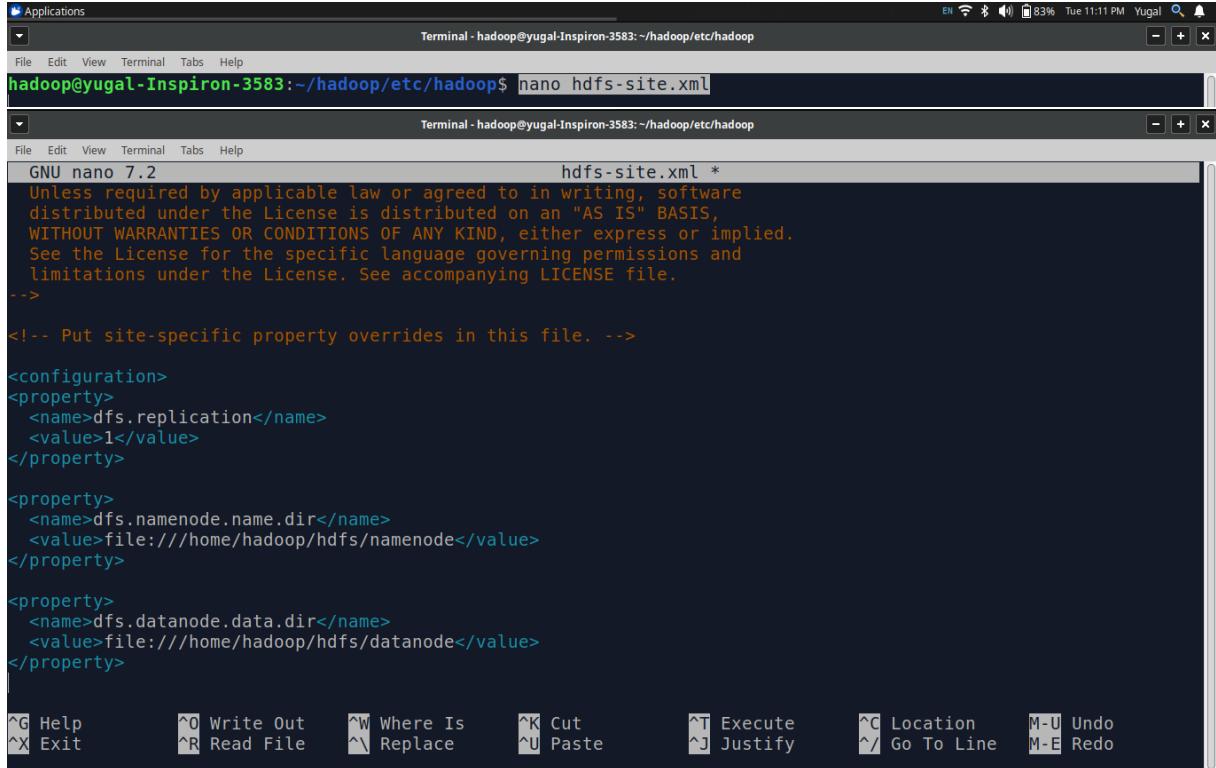
### hdfs-site.xml Modification

```
nano hdfs-site.xml
% Add the following property inside <configuration> tag
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>

<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///home/hadoop/hdfs/namenode</value>
</property>

<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:///home/hadoop/hdfs/datanode</value>
```

```
</property>
```



```
hadoop@yugal-Inspiron-3583:~/hadoop/etc/hadoop$ nano hdfs-site.xml
GNU nano 7.2
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>

<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///home/hadoop/hdfs/namenode</value>
</property>

<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:///home/hadoop/hdfs/datanode</value>
</property>
```

Figure 12: hdfs-site.xml Modification

### Creating Folder for NameNode and DataNode

```
mkdir -p ~/hdfs/namenode
mkdir -p ~/hdfs/datanode
```



```
hadoop@yugal-Inspiron-3583:~/hadoop/etc/hadoop$ mkdir -p ~/hdfs/namenode
hadoop@yugal-Inspiron-3583:~/hadoop/etc/hadoop$ mkdir -p ~/hdfs/datanode
hadoop@yugal-Inspiron-3583:~/hadoop/etc/hadoop$ |
```

Figure 13: Creating Folder for NameNode and DataNode

### mapred-site.xml Modification

```
nano mapred-site.xml
% Add the following property inside <configuration> tag
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
```

```

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>

```

Figure 14: mapred-site.xml Modification

### yarn-site.xml Modification

```

nano yarn-site.xml
% Add the following property inside <configuration> tag
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>

```

```

GNU nano 7.2                               yarn-site.xml *
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
</configuration>


```

Figure 15: yarn-site.xml Modification

## 1.6 Step 6: Format NameNode & Start Hadoop Services

## Format NameNode

```
hdfs namenode -format
```

Figure 16: Format NameNode

## Start Hadoop Services

start-dfs.sh

jps

```
hadoop@yugal-Inspiron-3583:~$ hadoop@yugal-Inspiron-3583:~$ start-dfs.sh Starting namenodes on [localhost] Starting datanodes Starting secondary namenodes [yugal-Inspiron-3583] hadoop@yugal-Inspiron-3583:~$ | hadoop@yugal-Inspiron-3583:~$ jps 81351 Jps 81062 SecondaryNameNode 80713 NameNode 80847 DataNode
```

Figure 17: Start Hadoop Services

## Start YARN Services

start-yarn.sh

jps

```

hadoop@yugal-Inspiron-3583:~$ jps
81351 Jps
81062 SecondaryNameNode
80713 NameNode
80847 DataNode
hadoop@yugal-Inspiron-3583:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@yugal-Inspiron-3583:~$ jps
81792 NodeManager
82355 Jps
81670 ResourceManager
81062 SecondaryNameNode
80713 NameNode
80847 DataNode
hadoop@yugal-Inspiron-3583:~$ |
hadoop@yugal-Inspiron-3583:~$ jps
81351 Jps
81062 SecondaryNameNode
80713 NameNode
80847 DataNode
hadoop@yugal-Inspiron-3583:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@yugal-Inspiron-3583:~$ jps
81792 NodeManager
82355 Jps
81670 ResourceManager
81062 SecondaryNameNode
80713 NameNode
80847 DataNode
hadoop@yugal-Inspiron-3583:~$ |




The terminal shows the execution of the 'start-yarn.sh' command, which starts the ResourceManager and NodeManagers. The browser window shows the 'DFS Health Overview' page at 'localhost:9870/dfshealth.html#tab-overview', with tabs for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities.



|                |                                                            |
|----------------|------------------------------------------------------------|
| Started:       | Tue Jan 27 23:49:34 +0530 2026                             |
| Version:       | 3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af           |
| Compiled:      | Tue Jul 07 00:14:00 +0530 2020 by brahma from branch-3.3.0 |
| Cluster ID:    | CID-7d9aa263-5957-4992-9da1-32c648d8341c                   |
| Block Pool ID: | BP-1685072063-127.0.1.1-1769537156943                      |


```

Figure 18: Start YARN Services

## 1.7 Step 7: HDFS Operations

```
% Create a test file
echo "Hello Big Data" > file.txt
% Make directory in HDFS
hdfs dfs -mkdir /data
```

```
% Upload file to HDFS
hdfs dfs -put file.txt /data
% List files
hdfs dfs -ls /data
% Check blocks & replication
hdfs fsck /data/file.txt -files -blocks
% Change permission
hdfs dfs -chmod 777 /data/file.txt
% Delete file
hdfs dfs -rm /data/file.txt
hdfs dfs -ls /data
```

```
File Edit View Terminal Tabs Help
Terminal - hadoop@yugal-Inspiron-3583: ~
hadoop@yugal-Inspiron-3583:~$ echo "Hello Big Data" > file.txt
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -mkdir /data
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -put file.txt /data
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -ls /data
Found 1 items
-rw-r--r-- 1 hadoop supergroup 15 2026-01-28 00:50 /data/file.txt
hadoop@yugal-Inspiron-3583:~$ hdfs fsck /data/file.txt -files -blocks
Connecting to namenode via http://localhost:9870/fsck?ugi=hadoop&files=1&blocks=1&path=%2Fdata%2Ffile.txt
FSCK started by hadoop (auth:SIMPLE) from /127.0.0.1 for path /data/file.txt at Wed Jan 28 00:51:19 IST 2026
/data/file.txt 15 bytes, replicated: replication=1, 1 block(s): OK
0. BP-1685072063-127.0.1.1-1769537156943.blk_1073741825_1001 len=15 Live_repl=1

Status: HEALTHY
Number of data-nodes: 1
Number of racks: 1
Total dirs: 0
Total symlinks: 0
```

Figure 19: HDFS Operation 1

```
Erasure Coded Block Groups:
Total size: 0 B
Total files: 0
Total block groups (validated): 0
Minimally erasure-coded block groups: 0
Over-erasure-coded block groups: 0
Under-erasure-coded block groups: 0
Unsatisfactory placement block groups: 0
Average block group size: 0.0
Missing block groups: 0
Corrupt block groups: 0
Missing internal blocks: 0
Blocks queued for replication: 0
FSCK ended at Wed Jan 28 00:51:19 IST 2026 in 37 milliseconds

The filesystem under path '/data/file.txt' is HEALTHY
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -chmod 777 /data/file.txt
chmod: '/data/file.txt': No such file or directory
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -rm /data/file.txt|
```

Figure 20: HDFS Operation 2

Browse Directory

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rwxrwxrwx	hadoop	supergroup	15 B	Jan 28 00:50	1	128 MB	file.txt

Showing 1 to 1 of 1 entries

Figure 21: HDFS Result 1

SUBMITTED Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used
0	0	0	0	0	0 B

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes
1	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus
No data available in table											

Showing 0 to 0 of 0 entries

Figure 22: HDFS Result 2

## **1.8 Result**

Hadoop was successfully installed and configured in pseudo-distributed mode, and various HDFS operations were performed successfully.

## **1.9 Conclusion**

This experiment provided hands-on experience in setting up a Big Data environment using Hadoop and understanding the working of HDFS through practical command execution.

## Assignment 2: Load large datasets into HDFS and analyze block distribution.

**Objective:** Understand how Apache Hadoop HDFS stores large files by splitting them into fixed-size blocks and placing them on DataNodes.

### 2.1 Step 1: Collect large Dataset (CSV/TXT)

```
su - hadoop  
start-dfs.sh  
start-yarn.sh  
jps
```

```
(base) yugal@yugal-Inspiron-3583:~$ su - hadoop  
Password:  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
hadoop@yugal-Inspiron-3583:~$ start-dfs.sh  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [yugal-Inspiron-3583]  
hadoop@yugal-Inspiron-3583:~$ start-yarn.sh  
Starting resourcemanager  
Starting nodemanagers  
Starting historyserver  
  
hadoop@yugal-Inspiron-3583:~$ jps  
4771 NodeManager  
3924 NameNode  
4136 DataNode  
5128 Jps  
4315 SecondaryNameNode  
4651 ResourceManager
```

Figure 23: Start Hadoop Services and Verify with jps

### 2.2 Step 2: Create HDFS directory.

```
% Created 352 MB csv file using:excel and uploaded to hdfs:  
% Created exceldata directory in hdfs:  
hdfs dfs -mkdir /exceldata  
ls -lh mydataset.csv  
hdfs dfs -put mydataset.csv /exceldata  
hdfs dfs -ls /exceldata  
  
hadoop@yugal-Inspiron-3583:~$ cp "/home/yugal/Desktop/Semester 2/Labs/Big Data Analytics Lab/File/Lab2/mydataset.csv" ~/  
hadoop@yugal-Inspiron-3583:~$ ls -lh mydataset.csv  
-rw-rw-r-x 1 hadoop hadoop 124M Feb 3 15:11 mydataset.csv  
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -put mydataset.csv /exceldata  
hadoop@yugal-Inspiron-3583:~$  
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -ls /exceldata  
hadoop@yugal-Inspiron-3583:~$ Found 1 items  
-rw-r--r-- 1 hadoop supergroup 129488515 2026-02-03 15:11 /exceldata/mydataset.csv
```

Figure 24: Created exceldata directory in hdfs and copied dataset

### 2.3 Step 3: Upload large dataset into HDFS

```
hdfs dfs -put mydataset.csv /exceldata
```

Figure 25: Upload large dataset into HDFS

## 2.4 Step 4: Verify data storage

```
hdfs dfs -ls /exceledata
```

```
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -ls /exceledata
Found 1 items
-rw-r--r-- 1 hadoop supergroup 129488515 2026-02-03 15:11 /exceledata/mydataset.csv
```

Figure 26: Verify data storage in HDFS

## 2.5 Step 5: Analyze block distribution

```
hdfs fsck /exceledata/mydataset.csv -files -blocks -locations
```

```
hadoop@yugal-Inspiron-3583:~$ hdfs fsck /exceledata/mydataset.csv -files -blocks -locations
Connecting to namenode via http://localhost:9870/fsck?ugi=hadoop&files=1&blocks=1&locations=1&path=%2Fexceledata%2Fmydataset.csv
FSCK started by hadoop (auth:SIMPLE) from /127.0.0.1 for path /exceledata/mydataset.csv at Tue Feb 03 15:18:34 IST 2026

/exceledata/mydataset.csv 369487717 bytes, replicated: replication=1, 3 block(s): OK
0. BP-1685072063-127.0.1.1-1769537156943:blk_1073741830_1006 len=134217728 Live_repl=1 [DatanodeInfoWithStorage[127.0.0.1:9866,DS-47acaae1-e863-462a-9aec-290782a597df,DISK]]
1. BP-1685072063-127.0.1.1-1769537156943:blk_1073741831_1007 len=134217728 Live_repl=1 [DatanodeInfoWithStorage[127.0.0.1:9866,DS-47acaae1-e863-462a-9aec-290782a597df,DISK]]
2. BP-1685072063-127.0.1.1-1769537156943:blk_1073741832_1008 len=101052261 Live_repl=1 [DatanodeInfoWithStorage[127.0.0.1:9866,DS-47acaae1-e863-462a-9aec-290782a597df,DISK]]

Status: HEALTHY
Number of data-nodes: 1
Number of racks: 1
Total dirs: 0
Total symlinks: 0

Replicated Blocks:
Total size: 369487717 B
Total files: 1
Total blocks (validated): 3 (avg. block size 123162572 B)
Minimally replicated blocks: 3 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 0 (0.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 1
Average block replication: 1.0
Missing blocks: 0
Corrupt blocks: 0
Missing replicas: 0 (0.0 %)
Blocks queued for replication: 0

Erasure Coded Block Groups:
Total size: 0 B
Total files: 0
Total block groups (validated): 0
Minimally erasure-coded block groups: 0
Over-erasure-coded block groups: 0
```

Figure 27: Analyze block distribution in HDFS

## 2.6 Step 6: Download processed dataset from HDFS to local system

```
hdfs dfs -get /exceledata/mydataset.csv mydataset_from_hdfs.csv
```

```
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -get /exceledata/mydataset.csv mydataset_from_hdfs.csv
```

Figure 28: Download processed dataset from HDFS to local system

## 2.7 Step 7: Delete data from HDFS

```
hdfs dfs -rm -r /exceledata
```

```
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -get /exceledata/mydataset.csv mydataset_from_hdfs.csv
hdfs dfs -rm -r /exceledata
get: 'mydataset_from_hdfs.csv': File exists
Deleted /exceledata
hadoop@yugal-Inspiron-3583:~$ |
```

Figure 29: Delete data from HDFS

## 2.8 Step 8: Check file block information

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	hadoop	supergroup	352.37 MB	Feb 03 15:18	1	128 MB	mydataset.csv

Figure 30: Check file block information in HDFS

## 2.9 Result

The large dataset (mydataset.csv) was successfully uploaded to HDFS under the /exceldata directory. The file was split into blocks and distributed across DataNodes, as verified by the block distribution analysis. The dataset was also downloaded back to the local system, confirming that data retrieval from HDFS works correctly. Finally, the dataset was deleted from HDFS, demonstrating proper data management and cleanup.

## 2.10 Conclusion

This assignment provided hands-on experience with Apache Hadoop HDFS, demonstrating how to upload large datasets, analyze block distribution, and manage data within the HDFS environment.

## Assignment 3: Implement MapReduce programs for Data Processing.

**Objective:** Apply distributed computing concepts by implementing a MapReduce program in Hadoop to process data efficiently across distributed nodes.

### 3.1 Step 1: Start Hadoop Services

```
su - hadoop  
start-dfs.sh  
start-yarn.sh  
jps
```

```
(base) yugal@yugal-Inspiron-3583:~$ su - hadoop  
Password:  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
hadoop@yugal-Inspiron-3583:~$ start-dfs.sh  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [yugal-Inspiron-3583]  
hadoop@yugal-Inspiron-3583:~$ start-yarn.sh  
Starting resourcemanager  
Starting nodemanagers  
hadoop@yugal-Inspiron-3583:~$ jps  
4593 DataNode  
5174 NodeManager  
4454 NameNode  
5048 ResourceManager  
4763 SecondaryNameNode  
5518 Jps
```

Figure 31: Starting Hadoop services and verifying with jps

### 3.2 Step 2: Create Input Directory in HDFS

```
nano input.txt  
ls  
cat input.txt  
hdfs dfs -mkdir /input  
hdfs dfs -put input.txt /input  
hdfs dfs -ls /input
```

```

hadoop@yugal-Inspiron-3583:~$ nano input.txt
hadoop@yugal-Inspiron-3583:~$ ls
bigfile.txt  hadoop          hdfs      mydataset.csv      snap
file.txt     hadoop-3.3.0.tar.gz  input.txt  mydataset_from_hdfs.csv
hadoop@yugal-Inspiron-3583:~$ cat input.txt
hadoop mapreduce hadoop
big data hadoop mapreduce
mapreduce big data
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -mkdir /input
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -put input.txt /input
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -ls /input
Found 1 items
-rw-r--r--  1 hadoop supergroup       69 2026-02-10 13:54 /input/input.txt

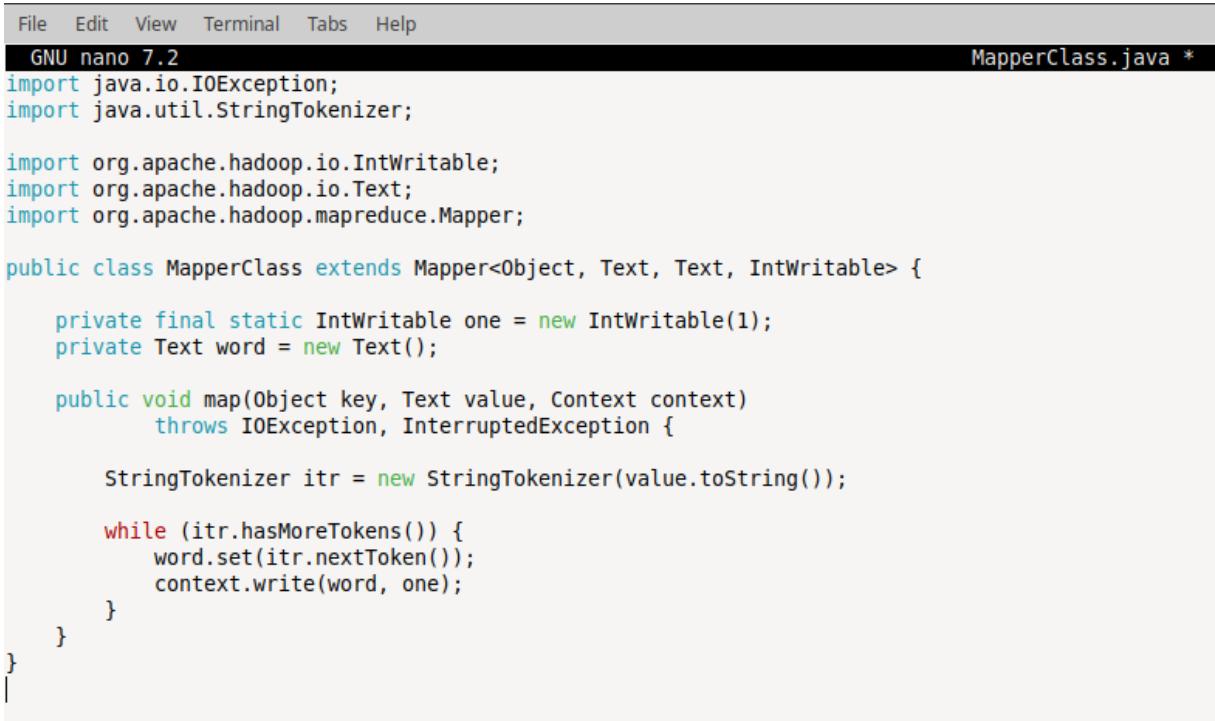
```

Figure 32: Creating input directory and uploading dataset

### 3.3 Step 3: Write MapReduce Program

**Mapper Class:** Processes input data and generates intermediate key-value pairs.

```
nano Mapper.java
```



```

File Edit View Terminal Tabs Help
GNU nano 7.2                                     MapperClass.java *
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MapperClass extends Mapper<Object, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context)
        throws IOException, InterruptedException {

        StringTokenizer itr = new StringTokenizer(value.toString());

        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
        }
    }
}

```

Figure 33: Mapper class implementation

**Reducer Class:** Aggregates values for each key produced by the mapper.

```
nano Reducer.java
```

```

GNU nano 7.2                                         ReducerClass.java *

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class ReducerClass extends Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {

        int sum = 0;

        for (IntWritable val : values) {
            sum += val.get();
        }

        context.write(key, new IntWritable(sum));
    }
}

```

Figure 34: Reducer class implementation

**Driver Class:** Configures job parameters such as input/output paths, mapper, reducer, and execution settings.

```
nano Driver.java
```

```

GNU nano 7.2                                         Driver.java *

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class Driver{

    public static void main(String[] args) throws Exception{

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Word Count");
        job.setJarByClass(Driver.class);
        job.setMapperClass(MapperClass.class);
        job.setReducerClass(ReducerClass.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}

```

Figure 35: Driver class implementation

### 3.4 Step 4: Compile Program and Create JAR

```
javac -classpath 'hadoop classpath' -d . Mapper.java Reducer.java Driver.java  
jar -cvf wordcount.jar *.class
```

```
hadoop@yugal-Inspiron-3583:~$ javac -classpath `hadoop classpath` -d . MapperClass.java ReducerClass.java Driver.java  
hadoop@yugal-Inspiron-3583:~$ jar -cvf wordcount.jar *.class  
added manifest  
adding: Driver.class(in = 1342) (out= 741)(deflated 44%)  
adding: MapperClass.class(in = 1680) (out= 728)(deflated 56%)  
adding: ReducerClass.class(in = 1591) (out= 661)(deflated 58%)
```

Figure 36: Compilation and JAR creation

### 3.5 Step 5: Execute MapReduce Job

```
hadoop jar wordcount.jar Driver /input /output
```

```
hadoop@yugal-Inspiron-3583:~$ hadoop jar wordcount.jar Driver /input /output  
2026-02-10 14:49:26,254 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032  
2026-02-10 14:49:26,823 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.  
2026-02-10 14:49:26,869 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1770715104976_0002  
2026-02-10 14:49:27,964 INFO input.FileInputFormat: Total input files to process : 1  
2026-02-10 14:49:28,492 INFO mapreduce.JobSubmitter: number of splits:1  
2026-02-10 14:49:29,226 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1770715104976_0002  
2026-02-10 14:49:29,226 INFO mapreduce.JobSubmitter: Executing with tokens: []  
2026-02-10 14:49:29,497 INFO conf.Configuration: resource-types.xml not found  
2026-02-10 14:49:29,498 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.  
2026-02-10 14:49:29,924 INFO impl.YarnClientImpl: Submitted application application_1770715104976_0002  
2026-02-10 14:49:30,143 INFO mapreduce.Job: The url to track the job: http://yugal-Inspiron-3583:8088/proxy/application_1770715104976_0002/  
2026-02-10 14:49:30,145 INFO mapreduce.Job: Running job: job_1770715104976_0002  
2026-02-10 14:49:42,586 INFO mapreduce.Job: Job job_1770715104976_0002 running in uber mode : false  
2026-02-10 14:49:42,588 INFO mapreduce.Job: map 0% reduce 0%  
2026-02-10 14:49:49,823 INFO mapreduce.Job: map 100% reduce 0%  
2026-02-10 14:49:57,932 INFO mapreduce.Job: map 100% reduce 100%  
2026-02-10 14:49:58,977 INFO mapreduce.Job: Job job_1770715104976_0002 completed successfully  
2026-02-10 14:49:59,128 INFO mapreduce.Job: Counters: 54  
File System Counters  
FILE: Number of bytes read=135  
FILE: Number of bytes written=527713  
FILE: Number of read operations=0  
FILE: Number of large read operations=0  
FILE: Number of write operations=0  
HDFS: Number of bytes read=171  
HDFS: Number of bytes written=34  
HDFS: Number of read operations=8  
HDFS: Number of large read operations=0  
HDFS: Number of write operations=2  
HDFS: Number of bytes read erasure-coded=0  
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)=4923  
Total time spent by all reduces in occupied slots (ms)=4834  
Total time spent by all map tasks (ms)=4923  
Total time spent by all reduce tasks (ms)=4834
```

Figure 37: Executing MapReduce job

### 3.6 Step 6: Check Output Directory in HDFS

```
hdfs dfs -ls /output
```

```
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -ls /output  
Found 2 items  
-rw-r--r-- 1 hadoop supergroup 0 2026-02-10 14:49 /output/_SUCCESS  
-rw-r--r-- 1 hadoop supergroup 34 2026-02-10 14:49 /output/part-r-00000
```

Figure 38: Output directory created in HDFS

### 3.7 Step 7: Display Result File

```
hdfs dfs -cat /mapreduce_output/part-r-00000
```

```
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -cat /output/part-r-00000
big      2
data     2
hadoop   3
mapreduce 3
```

Figure 39: Displaying MapReduce output

### 3.8 Step 8: Modify Input and Re-run Job

```
nano input.txt
hdfs dfs -put input.txt /input
hdfs dfs -rm -r /output
hadoop jar wordcount.jar Driver /input /output
hdfs dfs -cat /output/part-r-00000
```

```
GNU nano 7.2
hadoop mapreduce hadoop
big data hadoop mapreduce
mapreduce big data
hadoop big big data
|
```

Figure 40: Updating input text file

```
hadoop@yugal-Inspiron-3583:~$ nano input.txt
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -put -f input.txt /input
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -rm -r /output
Deleted /output
hadoop@yugal-Inspiron-3583:~$ hadoop jar wordcount.jar Driver /input /output
2026-02-10 14:52:21,999 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2026-02-10 14:52:22,752 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2026-02-10 14:52:22,783 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hadoop/.staging/job_1770715104976_0003
2026-02-10 14:52:23,275 INFO input.FileInputFormat: Total input files to process : 1
2026-02-10 14:52:23,434 INFO mapreduce.JobSubmitter: number of splits:1
2026-02-10 14:52:24,032 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1770715104976_0003
2026-02-10 14:52:24,033 INFO mapreduce.JobSubmitter: Executing with tokens: []
2026-02-10 14:52:24,475 INFO conf.Configuration: resource-types.xml not found
2026-02-10 14:52:24,479 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2026-02-10 14:52:24,639 INFO impl.YarnClientImpl: Submitted application application_1770715104976_0003
2026-02-10 14:52:24,698 INFO mapreduce.Job: The url to track the job: http://yugal-Inspiron-3583:8088/proxy/application_1770715104976_0003/
2026-02-10 14:52:24,698 INFO mapreduce.Job: Running job: job_1770715104976_0003
2026-02-10 14:52:34,086 INFO mapreduce.Job: Job job_1770715104976_0003 running in uber mode : false
2026-02-10 14:52:34,088 INFO mapreduce.Job: map 0% reduce 0%
2026-02-10 14:52:41,245 INFO mapreduce.Job: map 100% reduce 0%
2026-02-10 14:52:49,359 INFO mapreduce.Job: map 100% reduce 100%
2026-02-10 14:52:50,431 INFO mapreduce.Job: Job job_1770715104976_0003 completed successfully
2026-02-10 14:52:50,608 INFO mapreduce.Job: Counters: 54
    File System Counters
        FILE: Number of bytes read=179
        FILE: Number of bytes written=527801
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=191
```

Figure 41: Re-running MapReduce with modified dataset

```

Map output records=14
Map output bytes=145
Map output materialized bytes=179
Input split bytes=102
Combine input records=0
Combine output records=0
Reduce input groups=4
Reduce shuffle bytes=179
Reduce input records=14
Reduce output records=4
Spilled Records=28
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=100
CPU time spent (ms)=1950
Physical memory (bytes) snapshot=473153536
Virtual memory (bytes) snapshot=5457133568
Total committed heap usage (bytes)=524288000
Peak Map Physical memory (bytes)=283176960
Peak Map Virtual memory (bytes)=2720944128
Peak Reduce Physical memory (bytes)=189976576
Peak Reduce Virtual memory (bytes)=2736189440
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=89
File Output Format Counters
Bytes Written=34
hadoop@yugal-Inspiron-3583:~$ hdfs dfs -cat /output/part-r-00000
big    4
data   3
hadoop 4
mapreduce 3
hadoop@yugal-Inspiron-3583:~$ |

```

Figure 42: New output after re-running MapReduce job

### 3.9 Result

The MapReduce job was successfully executed on Hadoop. Input data stored in HDFS was processed using Mapper and Reducer classes, and the output was generated in the HDFS output directory. Re-running the job with modified input demonstrated changes in output based on data size and content.

### 3.10 Conclusion

This assignment demonstrated the practical implementation of the MapReduce programming model in Hadoop. It showed how distributed computing enables scalable data processing by dividing tasks into mapping and reducing phases, improving efficiency for large datasets.