

BIG DATA CLOUD LAB MANUAL

NAME : PRAGADEESHWARAN K J

ROLL.NO: 20BIS031

COURSE : BIGDATA TECHNOLOGIES

CODE : U18ISI5202T

LAB EXERCISE-1

AIM:

Perform setting up and Installing Hadoop in its three operating modes: Standalone, Pseudo distributed, Fully distributed

INSTALLATION STEPS:

Step 1: Install OpenJDK on Ubuntu-\$ sudo apt install

openjdk-8-jdk -y

Verify the installed version of Java- \$ java -version

Step 2: Create Hadoop User and Configure Password-less SSH

Add a new user hadoop - \$ sudo adduser hadoop

Add the hadoop user to the sudo group-\$ sudo usermod -aG sudo hadoop

Switch to the created user - \$ sudo su - hadoop

Install the OpenSSH server and client- \$ apt install openssh-server openssh-client -y

\$ sudo su - hadoop

Generate public and private key pairs- \$ ssh-keygen -t rsa

Add the generated public key from id_rsa.pub to authorized_keys

```
$ sudo cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Change the permissions of the authorized_keys file

```
$ sudo chmod 640 ~/.ssh/authorized_keys
```

Verify if the password-less SSH is functional

```
$ ssh localhost
```

Step 3: Install Apache Hadoop

Log in with hadoop user- \$ sudo su - hadoop

Download the latest stable version of Hadoop-

```
$ wget https://downloads.apache.org/hadoop/common/hadoop-3.3.1/hadoop-3.3.1.tar.gz
```

Extract the downloaded file- \$ tar -xvf hadoop-3.3.1.tar.gz

Move the extracted directory to the /usr/local/ directory.

```
$ sudo mv hadoop-3.3.1 /usr/local/hadoop
```

Create directory to store system logs-

```
$ sudo mkdir /usr/local/hadoop/logs
```

Change the ownership of the hadoop directory

```
$ sudo chown -R hadoop:hadoop /usr/local/hadoop
```

Step 4: Configure Hadoop

Edit file ~/.bashrc to configure the Hadoop environment variables

```
$ sudo nano ~/.bashrc
```

Add the following lines to the file. Save and close the file.

```
export HADOOP_HOME=/usr/local/hadoop
```

```
export HADOOP_INSTALL=$HADOOP_HOME
```

```
export HADOOP_MAPRED_HOME=$HADOOP_HOME
```

```
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export YARN_HOME=$HADOOP_HOME  
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native  
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin  
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"  
Activate the environment variables.  
$ source ~/.bashrc
```

Step 5: Configure Java Environment Variables

Find the OpenJDK directory- \$ readlink -f /usr/bin/javac

Edit the hadoop-env.sh file-

```
$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

Add the following lines to the file. Then, close and save the file-export
JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

```
export HADOOP_CLASSPATH+=" $HADOOP_HOME/lib/*.jar"
```

Browse to the hadoop lib directory-\$ cd /usr/local/hadoop/lib

Download the Javax activation file- \$ sudo wget
<https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar>

Verify the Hadoop version-\$ hadoop version

Edit the core-site.xml configuration file to specify the URL for your NameNode-\$ sudo nano
\$HADOOP_HOME/etc/hadoop/core-site.xml

Add the following lines. Save and close the file.

```
<configuration>  
  <property>  
    <name>fs.default.name</name>  
    <value>hdfs://0.0.0.0:9000</value>
```

```
<description>The default file system URI</description>
</property>
</configuration>
```

Create a directory for storing node metadata and change the ownership to hadoop.

```
$ sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}
```

```
$ sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

Add the following lines. Close and save the file.

```
<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
<property>
  <name>dfs.name.dir</name>
  <value>file:///home/hadoop/hdfs/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>file:///home/hadoop/hdfs/datanode</value>
</property>
</configuration>
```

Edit mapred-site.xml configuration file to define MapReduce values- \$ sudo nano \$HADOOP_HOME/etc/hadoop/mapred-site.xml

Add the following lines. Save and close the file.

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
```

```
<value>yarn</value>
</property>
</configuration>
```

Edit the yarn-site.xml configuration file and define YARN-related settings- \$ sudo nano
\$HADOOP_HOME/etc/hadoop/yarn-site.xml

Add the following lines. Save and close the file.

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>

</configuration>
```

Log in with hadoop user-\$ sudo su - hadoop

Validate the Hadoop configuration and format the HDFS NameNode-\$ hdfs namenode -format

Step 6: Start the Apache Hadoop Cluster

Start the NameNode and DataNode-\$ start-dfs.sh

Start the YARN resource and node managers-\$ start-yarn.sh

Verify all the running components-\$ jps

Step 7: Access Apache Hadoop Web Interface

<http://localhost:9870>

Screenshots:

```
Taylor Swift - Wildest Dreams - https://www.youtube.com/watch?v=I07y85526f6160600 X Install and Configure Apache Hadoop - How can I delete a user in linux - how to install word - YouTube - Applications Namendie information ... hadoop@ip-172-31-22-77: ~ hadoop@ip-172-31-22-77: ~

File Edit View Search Terminal Help

-p, --pty          create a new pseudo-terminal
-h, --help         display this help
-V, --version     display version

For more details see \[superuser\].
hadoop@ip-172-31-22-77:~$ sudo su - hadoop
hadoop@ip-172-31-22-77:~$ ssh-keygen -t rsa
Generating public key
Enter file in which to save the key ( /home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Overwrite (y/n)? y
Enter password (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:+CfyhslvZtEfmbPngNBB/j6euftM0ZT5qmjVSFw hadoop@ip-172-31-22-77
The key's randomart image is:
+---[RSA 3072]---+
|          .+ |
|          E+ |
|          .o o= |
|          .o o |
| S .. o=%h |
|          .oB0o |
|          o oo=. |
|          o o= |
|          o.B*oo |
+---[SHA256]---

hadoop@ip-172-31-22-77:~$ sudo cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys
hadoop@ip-172-31-22-77:~$ ssh localhost
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.15.0-1017-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Sat Dec 10 13:27:09 UTC 2022

System Load: 0.35           Processes: 237
System Load: 0.35           Processes: 237
Sat Dec 10 13:27:09 UTC 2022
Sat 10 Dec, 13:29 fabuse
```

```
Taylor Swift - Wildest Dreams ... F:\07a857691606000 Install and Configure Apache Hadoop How can I delete a user in linux how to install word - YouTube Applications Namenode information ... hadoop@ip-172-31-22-77: ~ hadoop@ip-172-31-22-77: ~ File Edit View Search Terminal Help hadoop@ip-172-31-22-77: /usr/local/hadoop/include/hadoop-3.3.1/include/hadoopUtilts.h hadoop@ip-172-31-22-77: /usr/local/hadoop/include/hadoop-3.3.1/include/Pipes.h hadoop@ip-172-31-22-77:~$ sudo mv hadoop-3.3.1 /usr/local/hadoop hadoop@ip-172-31-22-77:~$ sudo mkdir -p /usr/local/hadoop/logs hadoop@ip-172-31-22-77:~$ sudo cp /etc/hadoop/hadoop-env /usr/local/hadoop hadoop@ip-172-31-22-77:~$ sudo nano ~/.bashrc hadoop@ip-172-31-22-77:~$ source ~/.bashrc hadoop@ip-172-31-22-77:~$ which java /usr/bin/java hadoop@ip-172-31-22-77:~$ readlink -f /usr/bin/javaavc /usr/lib/jvm/java-11-openjdk-amd64/bin/javaavc hadoop@ip-172-31-22-77:~$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh hadoop@ip-172-31-22-77:~$ cd /usr/local/hadoop/lib hadoop@ip-172-31-22-77:~/lib$ curl -O https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/jar -2022-12-10 12:54:33. https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/jar-resolving javax.activation:jcenter:bintray.com [jcenter.bintray.com]:34.95.74.180 [2022-12-10 12:54:33.556] [application/java-archive] Length: 56674 (55K) [application/java-archive] Saving to: 'javax.activation-api-1.2.0.jar' Savin to: 'javax.activation-api-1.2.0.jar'
```



The screenshot shows a terminal window titled "hadoop@ip-172-31-22-7: ~" running on a Mac OS X desktop. The user is editing the file "/home/hadoop/.bashrc" using the nano editor. The terminal title bar also displays the URL "https://www.wuapro.com/guan.camille/#/client/a50wZDdhODUyYjQ2MDE2MDYwMABg51dmVsaw5r..." and the title "Install and Configure Apache Hadoop". The terminal interface includes standard Mac OS X window controls and a dock at the bottom.

```
File Edit View Search Terminal Help
GNU nano 4.8
/home/hadoop/.bashrc

export HADOOP_HOME=/usr/local/hadoop

export HADOOP_INSTALL=$HADOOP_HOME

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

export YARN_HOME=$HADOOP_HOME

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin

export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native":
```

```

Taylor Swift - Wildest Dreams - i-0d7a8529960160600 https://wsl.nuuvapro.com/guacamole/#/client/a50wZDdhQOULyjK2MDE2MDYwMA BjAG51dmVsaWSr... Applications Namenode information hadoop@ip-172-31-22-77:~ File Edit View Search Terminal Help hadoop@ip-172-31-22-77:~$ start-dfs.sh Starting namenodes on [0.0.0.0] 0.0.0.0: namenode is running as process 6374. Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry. Starting datanodes localhost: datanode is running as process 6538. Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry. Starting secondary namenodes [ip-172-31-22-77] ip-172-31-22-77: namenode is running as process 6815. Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry. hadoop@ip-172-31-22-77:~$ start-yarn.sh Starting resourcemanager resourcemanager is running as process 7056. Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry. Starting nodemanagers localhost: nodemanager is running as process 7216. Stop it first and ensure /tmp/hadoop-hadoop-nodemanager.pid file is empty before retry. hadoop@ip-172-31-22-77:~$ jps 7216 NodeManager 7056 ResourceManager 6374 NameNode 10456 Jps 6538 DataNode 6815 SecondaryNameNode hadoop@ip-172-31-22-77:~$ 

```

Taylor Swift - Wildest Dreams - i-0d7a8529960160600 https://wsl.nuuvapro.com/guacamole/#/client/a50wZDdhQOULyjK2MDE2MDYwMA BjAG51dmVsaWSr... Applications All Applications Google... hadoop@ip-172-31-22-77:~ Applications - Google... Namenode information DataNode information Sat 10 Dec, 13:33 labuser All Applications Namenode information DataNode information localhost:8088/cluster

All Applications

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	Start Time	Launch Time	Finish Time	State	Final Status	Running Containers	Alloc CPU	Alloc VCore
No data available in table														

Show 20 ▾ entries

Showing 0 to 0 of 0 entries

Taylor Swift - Wildest Dreams - i-0d7a8529960160600 https://wsl.nuuvapro.com/guacamole/#/client/a50wZDdhQOULyjK2MDE2MDYwMA BjAG51dmVsaWSr... Applications Namenode information hadoop@ip-172-31-22-77:~ Applications - Google... Namenode information DataNode information localhost:9870/dfshealth.html#tab-overview Sat 10 Dec, 13:33 labuser

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Overview '0.0.0.0:9000' (active)

Started:	Sat Dec 10 12:57:17 +0000 2022
Version:	3.3.1, ra3b9c37a397ad4188041dd80621bdeefc46885f2
Compiled:	Tue Jun 15 05:13:00 +0000 2021 by ubuntu from (HEAD detached at release-3.3.1-RC3)
Cluster ID:	CID-dca4fb9e-2afa-4273-a962-0580f71997cb
Block Pool ID:	BP-1987065403-172.31.22.77-1670677027329

Summary

Security is off.
Safe mode is off.
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).
Heap Memory used 77.92 MB of 315 MB Heap Memory. Max Heap Memory is 1.92 GB.
Non Heap Memory used 49.48 MB of 53.13 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity: 19.31 GB

Taylor Swift - Wildest Dreams - i-007a852b960160600

Install and Configure Apache Hadoop - https://www.nuwavepro.com/guacamole/#/client/s5DwZDihODdyYk2MDc2MDYwMAbjAG51dmVnWSr...

All Applications DataNode Information hadoop@ip-172-31-22-7...

Namenode information DataNode Information

localhost: 9864/datanode.html

Hadoop Overview Utilities

DataNode on ip-172-31-22-77.ap-south-1.compute.internal:9866

Cluster ID: CID-dca4fb9e-2afa-4273-a962-05b0f71997cb

Version: 3.3.1, ra3b9c37a397ad4188041dd80621bdeefc468892

Block Pools

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
0.0.0.0:9000	BP-1987065403-172.31.22.77-1670677027329	RUNNING	1s	36 minutes	0 B (128 MB)

Volume Information

Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
/home/hadoop/hdfs/datanode	DISK	28 KB	9.7 GB	0 B	0 B	0

LAB EXERCISE-2

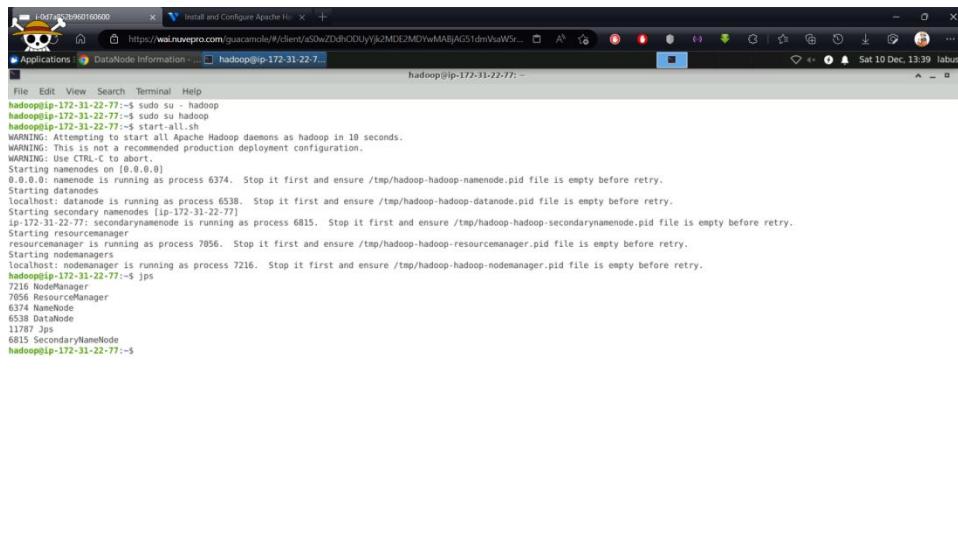
AIM:

Implement the following file management tasks in Hadoop:

- Adding files and directories
- Retrieving files
- Deleting files

FILE OPERATIONS:

Step 1. Open terminal in Ubuntu and start a cluster

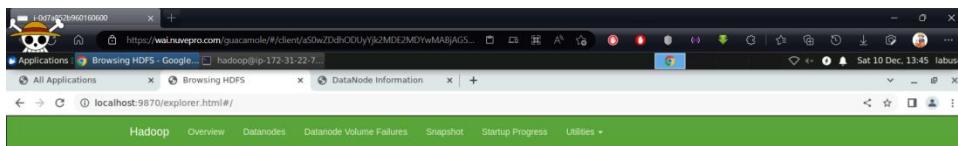


The screenshot shows a terminal window titled "hadoop@ip-172-31-22-7: ~". The window displays the command-line output of starting an Apache Hadoop cluster. The output includes several warning messages about daemon startup and pid file handling. It lists the processes being started: NameNode, DataNode, SecondaryNameNode, ResourceManager, and NodeManager. The process IDs for these daemons are also listed.

```
hadoop@ip-172-31-22-7:~$ sudo su - hadoop
hadoop@ip-172-31-22-7:~$ sudo su hadoop
hadoop@ip-172-31-22-7:~$ sh start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [0.0.0.0]
0.0.0.0 namenode is running as process 6374. Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 6538. Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondarynamenodes [ip-172-31-22-7]
Starting resourcemanager
resourcemanager is running as process 7056. Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 7216. Stop it first and ensure /tmp/hadoop-hadoop-nodemanager.pid file is empty before retry.
hadoop@ip-172-31-22-7:~$ jps
7216 NodeManager
7056 ResourceManager
6374 NameNode
6538 DataNode
11787 Jps
6815 SecondaryNameNode
hadoop@ip-172-31-22-7:~$
```

Step 2: Create a directory in HDFS at given path(s).

```
hdfs dfs -mkdir /hdemo/
```



Browse Directory

/								
<input type="text" value="Go!"/> File Folder New Delete Rename								
Show 25 entries Search:								
□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	drwxr-xr-x	hadoop	supergroup	0 B	Dec 10 13:41	0	0 B	hdemo

Showing 1 to 1 of 1 entries

[Previous](#) [1](#) [Next](#)

Hadoop, 2021.

Step 3: List the contents of a directory.

hdfs dfs -ls /

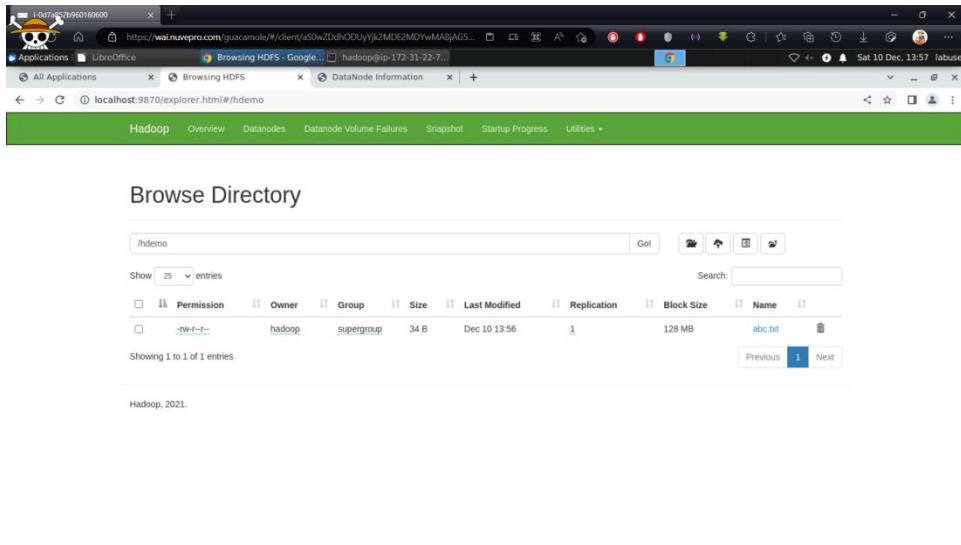
```

https://192.168.1.10:9870/explorer.html#/datanode
File Edit View Search Terminal Help
hadoop@ip-172-31-22-77:~$ sudo su hadoop
hadoop@ip-172-31-22-77:~$ sudo su hadoop
hadoop@ip-172-31-22-77:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: This is not a recommended production deployment configuration.
Starting namenodes on [0.0.0.0].
0.0.0.0: namenode is running as process 6374. Stop it first and ensure /tmp/hadoop-hadoop-namenode.pid file is empty before retry.
Starting datanodes
localhost: datanode is running as process 6538. Stop it first and ensure /tmp/hadoop-hadoop-datanode.pid file is empty before retry.
Starting secondary namenodes [ip-172-31-22-77]
ip-172-31-22-77: secondarynamenode is running as process 6815. Stop it first and ensure /tmp/hadoop-hadoop-secondarynamenode.pid file is empty before retry.
Starting resourcemanager
resourcemanager is running as process 7056. Stop it first and ensure /tmp/hadoop-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
localhost: nodemanager is running as process 7216. Stop it first and ensure /tmp/hadoop-hadoop-nodemanager.pid file is empty before retry.
hadoop@ip-172-31-22-77:~$ jps
7216 NodeManager
7056 ResourceManager
6374 NameNode
6538 DataNode
1737 SecondaryNameNode
6815 SecondaryNameNode
hadoop@ip-172-31-22-77:~$ Step 2: Create a directory in HDFS at given path(s).
bash: syntax error near unexpected token `('
hadoop@ip-172-31-22-77:~$ hdfs dfs -mkdir /hdemo
hadoop@ip-172-31-22-77:~$ hdfs dfs -mkdir /hdemo
hadoop@ip-172-31-22-77:~$ hdfs dfs -mkdir /bdata
hadoop@ip-172-31-22-77:~$ hdfs dfs -ls /
Found 2 items
drwxr-xr-x - hadoop supergroup 0 2022-12-10 13:46 /hdemo
drwxr-xr-x - hadoop supergroup 0 2022-12-10 13:41 /bdata
hadoop@ip-172-31-22-77:~$ 

```

Step 4: Upload and download a file in H Upload

```
hdfs dfs -put '/home/labuser/Documents/abc.txt' /hdemo/
```



Step 5: See contents of a file:

```
hdfs dfs -cat /hdemo/abc.txt
```

```

https://www.geekforge...x HDFS Commands - GeekforGeeks + 
https://www.nuvepro.com/guacamole/#/client/a50wZDhhODUyYjZtMDE2MDYwMABjAG51dmVsWSr... Applications LibreOffice Browsing HDFS - Google hadoop@ip-172-31-22-77: ~ hadoop@ip-172-31-22-77: ~

File Edit View Search Terminal Help
Admin Commands:
daemonlog get/set the log level for each daemon
Client Commands:
archive create a Hadoop archive
checknative check native Hadoop and compression libraries availability
classpath prints the class path needed to get the Hadoop jar and the required libraries
confstore validate configuration XML files
crecord list current record providers
distch distributed metadata changes
distcp copy file or directories recursively
dtutil operations related to delegation tokens
envvars display computed Hadoop environment variables
fs run a generic filesystem user client
gridmix submit a mix of synthetic job, modeling a profiled from production load
jar <jar> run the Java library path
jnipath print the Java library path
kduadmin Diagnostic Kernel Problem
kerbname show auth to local principal conversion
key manage keys via the KeyProvider
runnertrace scale a runner input trace
runnertrace scale a runner in a runner trace
s3guard manage metadata on S3
trace view and modify Hadoop tracing settings
version print the version
Daemon Commands:
kms run KMS, the Key Management Server
registrydns run the registry DNS server
SUBCOMMAND may print help when invoked w/o parameters or with -h.
hadoop@ip-172-31-22-77: ~ hadoop dfs -cat /hdemo/abc.txt
WARNING: Use of this script is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.
Bigdata hadoop cloud file test
hadoop@ip-172-31-22-77: ~ hdfs dfs -cat /hdemo/abc.txt
Bigdata hadoop cloud file test
hadoop@ip-172-31-22-77: ~

```

Step 6: Copy a file from source to destination

`hdfs dfs -cp /hdemo/abc.txt /bdata`

```

https://www.geekforge...x HDFS Commands - GeekforGeeks + 
https://www.nuvepro.com/guacamole/#/client/a50wZDhhODUyYjZtMDE2MDYwMABjAG51dmVsWSr... Applications LibreOffice Browsing HDFS - Google hadoop@ip-172-31-22-77: ~ hadoop@ip-172-31-22-77: ~

File Edit View Search Terminal Help
hadmin run a DFS HA admin client
jmxget get JMX exported values from NameNode or DataNode
oey apply the offline edits viewer to an edits file
oiv apply the offline fsimage viewer to an fsimage
oiv legacy apply the offline fsimage viewer to a legacy fsimage
storagepolicies list/get/set/satisfyStoragePolicy block storage policies
Client Commands:
classpath prints the class path needed to get the hadoop jar and the required libraries
dfs run a filesystem command on the file system
envvars display computed Hadoop environment variables
fetchtoken fetch a delegation token from the NameNode
getconf get configuration values from configuration files
groups get the groups which users belong to
lsSnapshottableDir list all snapshottable dirs owned by the current user
snapshotDiff diff two snapshots of a directory or diff the current directory contents with a snapshot
version print the version
Daemon Commands:
balancer run a cluster balancing utility
datanode run a DFS datanode
dfsrouter run the DFS router
diskbalancer Distributes data evenly among disks on a given node
httpfs run HTTPFS, the HDFS HTTP Gateway
journalnode run the DFS journalnode
mover run a utility to move block replicas across storage types
namenode run the DFS namenode
nfs3 run the HDFS 3 gateway
portmap run a portmap service
secondarynamenode run the DFS secondary namenode
sps run external storagepoliciesatisfier
zkfc run the ZK Failover Controller daemon
SUBCOMMAND may print help when invoked w/o parameters or with -h.
hadoop@ip-172-31-22-77: ~ hdfs dfs -cp /hdemo/abc.txt /bdata
hadoop@ip-172-31-22-77: ~ hdfs dfs -cat /bdata/abc.txt
cat abc.txt Is a directory
hadoop@ip-172-31-22-77: ~ hdfs dfs -cat /bdata/abc.txt
Bigdata hadoop cloud file test
hadoop@ip-172-31-22-77: ~

```

localhost:9870/explorer.html#/bdata

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
abc.txt	-rw-r--r--	hadoop	supergroup	34 B	Dec 10 14:07	1	128 MB	abc.txt

Browse Directory

/bdata								
Show 25 entries		Go!						
Search:								
	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
abc.txt	-rw-r--r--	hadoop	supergroup	34 B	Dec 10 14:07	1	128 MB	abc.txt

Showing 1 to 1 of 1 entries

Previous **1** Next

Hadoop, 2021.

Step 7: Move file from source to destination.

```
hdfs dfs -mv /hdemo/ds.png /bdata/
```

localhost:9870/explorer.html#/hdemo

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
abc.txt	-rw-r--r--	hadoop	supergroup	34 B	Dec 10 13:56	1	128 MB	abc.txt
bl.jpg	-rw-r--r--	hadoop	supergroup	1.6 MB	Dec 10 14:20	1	128 MB	bl.jpg
ds.png	-rw-r--r--	hadoop	supergroup	664.12 KB	Dec 10 14:29	1	128 MB	ds.png

Browse Directory

/hdemo								
		Go!						
Search:								
	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
abc.txt	-rw-r--r--	hadoop	supergroup	34 B	Dec 10 13:56	1	128 MB	abc.txt
bl.jpg	-rw-r--r--	hadoop	supergroup	1.6 MB	Dec 10 14:20	1	128 MB	bl.jpg
ds.png	-rw-r--r--	hadoop	supergroup	664.12 KB	Dec 10 14:29	1	128 MB	ds.png

Showing 1 to 3 of 3 entries

Previous **1** Next

Hadoop, 2021.

1247674.png

Show all X

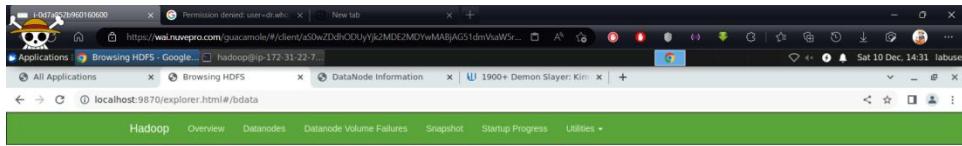
```

hadoop@ip-172-31-22-77: ~ + 
File Edit View Search Terminal Help
[-stat [format] <path> ...]
[-tail [-f] [-s <sleep interval>] <file>]
[-test [-ignorecrc] <src> ...]
[-touch [-a] [-m] [-t TIMESTAMP (yyyyMMddHHmmss)] [-c] <path> ...]
[-truncate [-w] <length> <path> ...]
[-usage [cmd ...]]
Generic options supported are:
-conf <configuration file> specify an application configuration file
-D <property=value> define a value for a given property
-fs <file://>/[hdfs://]namenode:<port> specify default filesystem URL to use, overrides 'fs.defaultFS' property from configurations.
-jt <local>/resourcemanager:<port> specify a ResourceManager
-libs <file1,...> specify a comma-separated list of files to be copied to the map reduce cluster
-libjars <jar1,...> specify a comma-separated list of jar files to be included in the classpath
-archives <archive1,...> specify a comma-separated list of archives to be unarchived on the compute machines

The general command line syntax is:
command [genericOptions] {commandOptions}

hadoop@ip-172-31-22-77: ~ $ sudo -u hdfs hadoop fs -mkdir /user/root
sudo: unable to initialize policy plugin
hadoop@ip-172-31-22-77: ~ $ sudo -u hdfs hadoop fs -chown root /user/root
sudo: unknown user 'root'
sudo: unable to initialize policy plugin
hadoop@ip-172-31-22-77: ~ $ sudo -u hadoop fs -chown root /user/root
sudo: fs: command not found
hadoop@ip-172-31-22-77: ~ $ sudo -u hadoop fs -chown root /user/root
sudo: fs: command not found
hadoop@ip-172-31-22-77: ~ $ hadoop fs -put file /user/root/
put: '/user/root': No such file or directory: hadoop@ip-172-31-22-77:~/hdfs/0.8.0-9000/user/root
hadoop@ip-172-31-22-77: ~ $ hadoop fs -put /home/labuser/Downloads/ds.jpg /hdemo/
put: '/hdemo/ds.jpg': No such file or directory
hadoop@ip-172-31-22-77: ~ $ hadoop fs -put /home/labuser/Downloads/b1.jpg /hdemo/
put: '/hdemo/b1.jpg': File exists
hadoop@ip-172-31-22-77: ~ $ hadoop fs -put /home/labuser/Downloads/ds.png /hdemo/
hadoop@ip-172-31-22-77: ~ $ hadoop fs -mv /hdemo/ds.png /bdemo/
hadoop@ip-172-31-22-77: ~ $ 

```



Browse Directory

/bdata								
Show 25 entries								
	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
✓	-rw-r--r--	hadoop	supergroup	34 B	Dec 10 14:07	1	128 MB	abc.txt
✓	-rw-r--r--	hadoop	supergroup	1.6 MB	Dec 10 14:18	1	128 MB	bl.jpg
✓	-rw-r--r--	hadoop	supergroup	664.12 KB	Dec 10 14:29	1	128 MB	ds.png

Showing 1 to 3 of 3 entries

Previous **1** Next

Hadoop, 2021.

Step 8: Remove a file or directory in HDFS.

`hdfs dfs -rm -r '/hdemo/'`

```

hdfs - How to remove files inside / | New tab | ...
https://www.wavepro.com/guacamole/#/client/a5DwZDdhQOOLyYk2MDE3MDYwMAbjAG51dmVsWSr...
Applications Browsing HDFS - Google... hadoop@ip-172-31-22-7: ~
File Edit View Search Terminal Help
hadoop@ip-172-31-22-7: ~

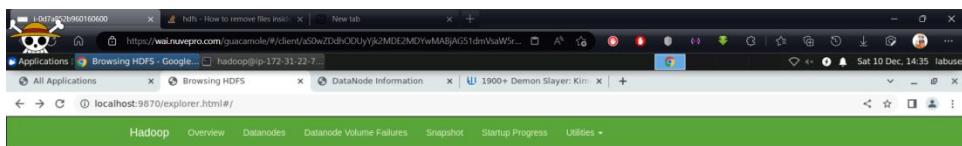
Client Commands:
fck          run a DFS filesystem checking utility
hadmin       run a DFS HA admin command
jmxget      get JMX metrics values from NameNode or DataNode.
ovv         apply the offline edits viewer to an edits file
ovv         apply the offline fsimage viewer to an fsimage
ovv legacy   apply the offline fsimage viewer to a legacy fsimage
storagelolicies list/get/set/satisfyStoragePolicy block storage policies

Client Commands:
classpath    prints the class path needed to get the hadoop jar and the required libraries
dfs          run a filesystem command on the file system
envvars      display computed Hadoop environment variables
fetchtoken   fetch a delegation token from the NameNode
getconf      get configuration from configuration
groups      get the groups which users belong to
lsSnapshottableDir list all snapshottable dirs owned by the current user
snapshotDiff diff two snapshots of a directory or diff the current directory contents with a snapshot
version     print the version

Daemon Commands:
balancer    run a cluster balancing utility
datanode    run a DFS datanode
dfsrouter   run the DFS router
diskbalancer Distributes data evenly among disks on a given node
https       run HTTPS server, the HDFS HTTP Gateway
journalnode run the journal node
mover       run a utility to move block replicas across storage types
namenode    run the DFS namenode
nfs3        run an NFS version 3 gateway
portmap     run a portmap service
secondarynamenode run the DFS secondary namenode
sps         run external storagepoliciesalstifler
zkfc        run the ZK Failover Controller daemon

SUBCOMMAND may print help when invoked w/o parameters or with -h.
hadoop@ip-172-31-22-7: ~$ hdfs dfs -rmdir /hdemo/
rmdir: '/hdemo': Directory is not empty
hadoop@ip-172-31-22-7: ~$ hdfs dfs -rm -r /hdemo/
Deleted /hdemo
hadoop@ip-172-31-22-7: ~$ 

```



Browse Directory

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	drwxr-xr-x	hadoop	supergroup	0 B	Dec 10 14:30	0	0 B	bdata

Showing 1 to 1 of 1 entries

Previous 1 Next

Hadoop, 2021.

Step 9: Display the aggregate length of a file.

hdfs dfs -du /bdata/

hdfs - How to remove files in hdfs | New tab | hadoop@ip-172-31-22-7: ~

```
File Edit View Search Terminal Help
olv apply the offline fsimage viewer to an fsimage
olv_legacy apply the offline fsimage viewer to a legacy fsimage
storagepolicies list/get/set/satisfyStoragePolicy block storage policies

Client Commands:
classpath prints the class path needed to get the hadoop jar and the required libraries
dfs run a filesystem command on the file system
envvars display computed Hadoop environment variables
fetchft fetch a delegation token from the NameNode
getconf get configuration files for configuration
groups get the groups which users belong to
lsSnapshottableDir list all snapshottable dirs owned by the current user
snapshotDiff diff two snapshots of a directory or diff the current directory contents with a snapshot
version print the version

Daemon Commands:
balancer run a cluster balancing utility
datanode run a DFS datanode
dfsrouter run the DFS router
diskbalancer Distributes data evenly among disks on a given node
httpsfs run HttpFS server, the HDFS HTTP Gateway
journalnode run the HDFS journalnode
mover run a utility to move block replicas across storage types
namenode run the DFS namenode
nfs3 run an NFS version 3 gateway
portmap run a portmap service
secondarynamenode run the DFS secondary namenode
sps run external storagepoliciesatisfier
zkfc run the ZK Failover Controller daemon

SUBCOMMAND may print help when invoked w/o parameters or with -h.

hadoop@ip-172-31-22-7:~$ hdfs dfs -rmdir /hdemo/
rmdir: '/hdemo': Directory is not empty
hadoop@ip-172-31-22-7:~$ hdfs dfs -rm -r '/hdemo/'
Deleted /hdemo
hadoop@ip-172-31-22-7:~$ hdfs dfs -du /bdata/
34 /bdata/abc.txt
1682958 1682958 /bdata/d1.jpg
680055 680055 /bdata/d2.png
hadoop@ip-172-31-22-7:~$
```

LAB EXCERSICE -3

WORD COUNT

STEP1-LOGIN TO THE WEB SHELL USING THE USERNAME AND PASSWORD

STEP2-CREATE A FOLDER/ DIRECTORY FOR weather IN HOME

STEP3-MOVE TO THAT FOLDER/DIRECTORY

STEP4-CREATE A input.txt FILE INSIDE THE weather FOLDER

STEP5-PASTE THE INPUT IN THE INPUT FILE

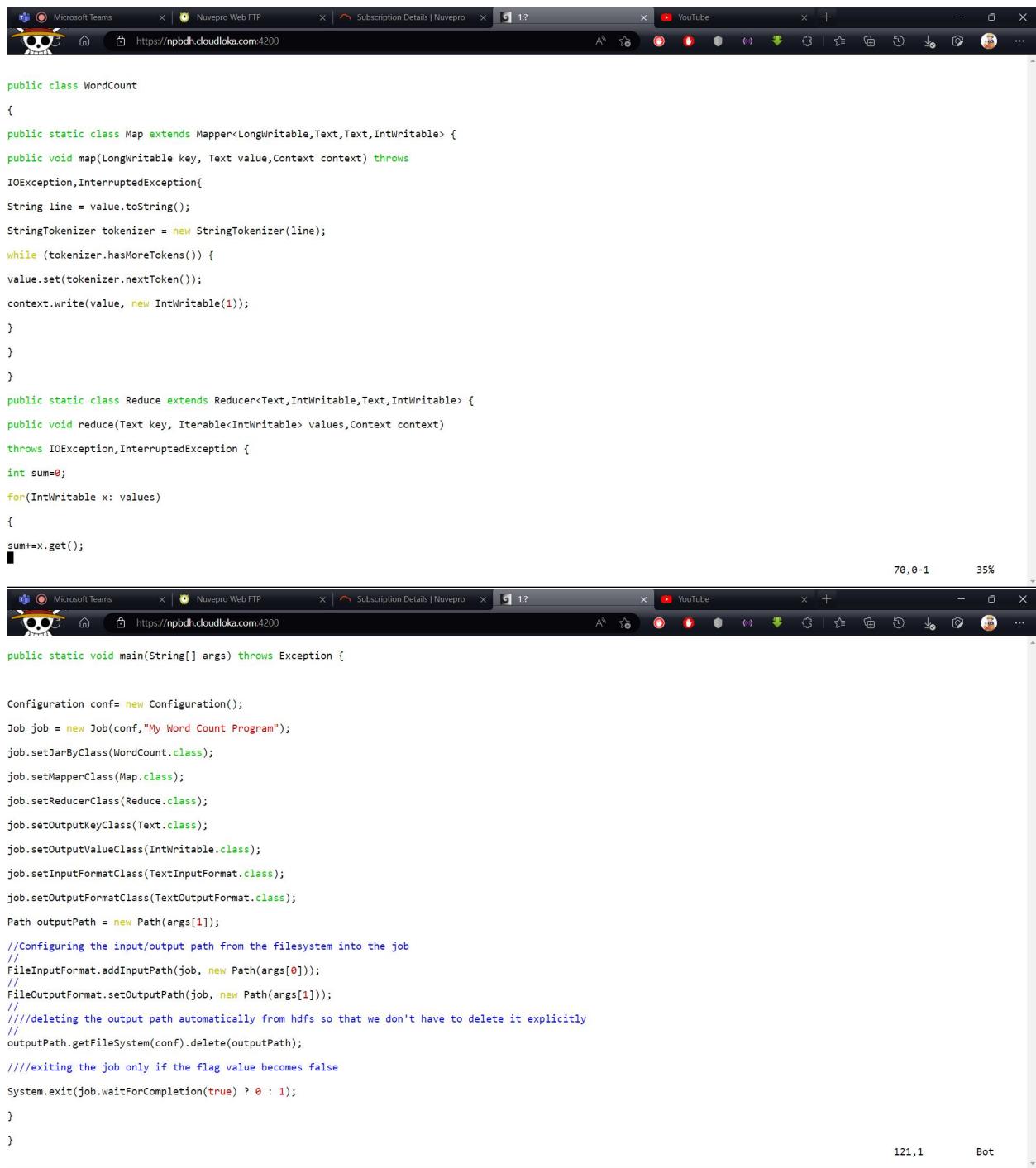


```
npbdh login: kct5thsemcnid031
kct5thsemcnid031@npbdh.cloudloka.com's password:
Last login: Mon Dec 12 14:40:34 2022 from ec2-65-1-45-35.ap-south-1.compute.amazonaws.com
[kct5thsemcnid031@ip-10-1-1-204 ~]$ mkdir wordcount
[kct5thsemcnid031@ip-10-1-1-204 ~]$ cd wordcount
[kct5thsemcnid031@ip-10-1-1-204 wordcount]$ touch input.txt
[kct5thsemcnid031@ip-10-1-1-204 wordcount]$ vi input.txt
```

STEP6-CREATE A WORDCOUNT JAVA FILE

```
npbdh login: kct5thsemcdhid031
kct5thsemcdhid031@npbdh.cloudloka.com's password:
Last login: Mon Dec 12 14:48:34 2022 from ec2-65-1-45-35.ap-south-1.compute.amazonaws.com
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ mkdir wordcount
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ cd wordcount
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ touch input.txt
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ vi input.txt
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ touch WordCount.java
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ vi WordCount.java
```

STEP7-PASTE THE SOURCE CODE IN THE FILE



The image shows a screenshot of a web browser window with two tabs open, both displaying Java code for a WordCount program. The browser interface includes a header with tabs for Microsoft Teams, Nuvepro Web FTP, Subscription Details | Nuvepro, and YouTube. Below the tabs is a toolbar with various icons. The left tab contains the Mapper and Reducer code, and the right tab contains the main() method and job configuration code.

```
public class WordCount {
{
public static class Map extends Mapper<LongWritable,Text,Text,IntWritable> {
public void map(LongWritable key, Text value,Context context) throws
IOException,InterruptedException{
String line = value.toString();
StringTokenizer tokenizer = new StringTokenizer(line);
while (tokenizer.hasMoreTokens()) {
value.set(tokenizer.nextToken());
context.write(value, new IntWritable(1));
}
}
}
}

public static class Reduce extends Reducer<Text,IntWritable,Text,IntWritable> {
public void reduce(Text key, Iterable<IntWritable> values,Context context)
throws IOException,InterruptedException {
int sum=0;
for(IntWritable x: values)
{
sum+=x.get();
}
}
}

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

Configuration conf= new Configuration();
Job job = new Job(conf,"My Word Count Program");
job.setJarByClass(WordCount.class);
job.setMapperClass(Map.class);
job.setReducerClass(Reduce.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
Path outputPath = new Path(args[1]);
//Configuring the input/output path from the filesystem into the job
//FileInputFormat.addInputPath(job, new Path(args[0]));
//FileOutputFormat.setOutputPath(job, new Path(args[1]));
//deleting the output path automatically from hdfs so that we don't have to delete it explicitly
//outputPath.getFileSystem(conf).delete(outputPath);
//exiting the job only if the flag value becomes false
System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}
```

SOURCE CODE:

```
import java.io.IOException;
```

```
import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.fs.Path;
```

```
public class WordCount

{

public static class Map extends Mapper<LongWritable,Text,Text,IntWritable> {

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

        String line = value.toString();

        StringTokenizer tokenizer = new StringTokenizer(line);

        while (tokenizer.hasMoreTokens()) {

            value.set(tokenizer.nextToken());

            context.write(value, new IntWritable(1));

        }

    }

}
```

```
public static class Reduce extends Reducer<Text,IntWritable,Text,IntWritable> {

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

        int sum=0;

        for(IntWritable x: values)

        {

            sum+=x.get();

        }

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

    }

}

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

```
Configuration conf= new Configuration();

Job job = new Job(conf,"My Word Count Program");

job.setJarByClass(WordCount.class);

job.setMapperClass(Map.class);

job.setReducerClass(Reduce.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

job.setInputFormatClass(TextInputFormat.class);

job.setOutputFormatClass(TextOutputFormat.class);

Path outputPath = new Path(args[1]);

//Configuring the input/output path from the filesystem into the job
//
FileInputFormat.addInputPath(job, new Path(args[0]));
//
FileOutputFormat.setOutputPath(job, new Path(args[1]));
//
```

```

///deleting the output path automatically from hdfs so that we don't have to delete it explicitly
//  

outputPath.getFileSystem(conf).delete(outputPath);

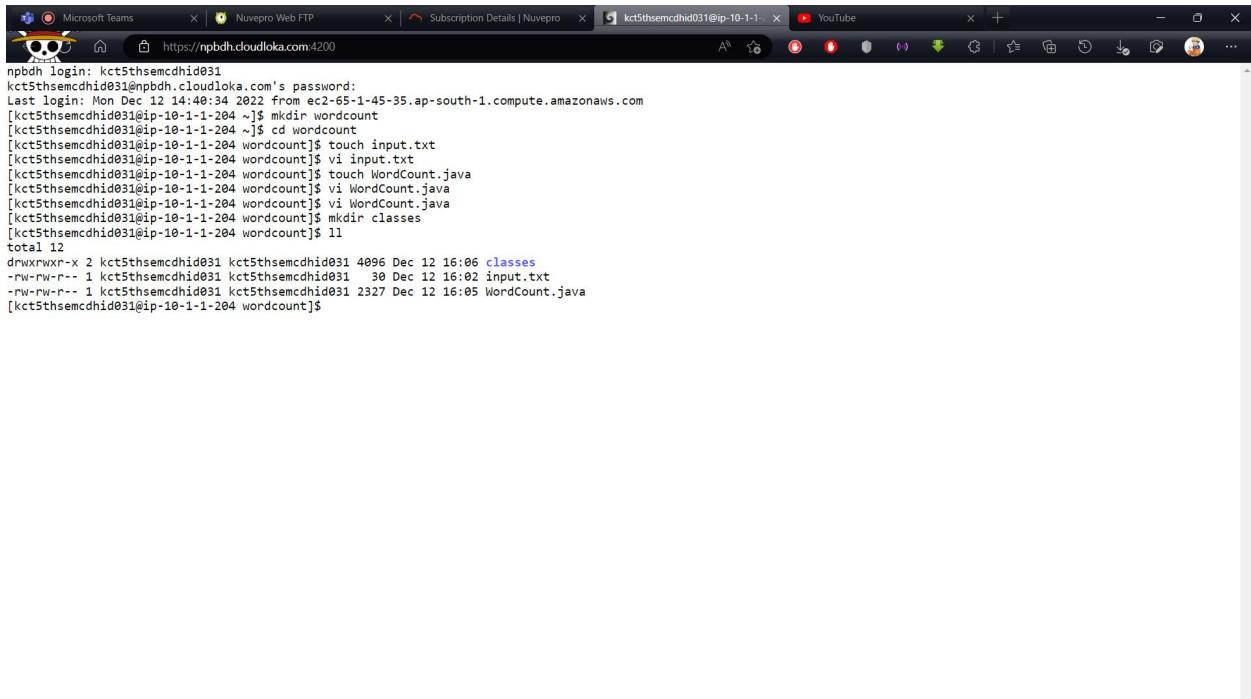
///exiting the job only if the flag value becomes false

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

}

```

STEP8-CREATE A DIRECTORY/ FOLDER CLASSES INSIDE THE WORDCOUNT FOLDER



The screenshot shows a terminal window with the following session:

```

npbdh login: kct5thsemcdhid031
kct5thsemcdhid031@npbdh.cloudloka.com's password:
Last login: Mon Dec 12 14:48:34 2022 from ec2-65-1-45-35.ap-south-1.compute.amazonaws.com
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ mkdir wordcount
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ cd wordcount
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ touch input.txt
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ vi input.txt
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ touch WordCount.java
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ vi WordCount.java
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ mkdir classes
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ ll
total 12
drwxrwxr-x 2 kct5thsemcdhid031 kct5thsemcdhid031 4096 Dec 12 16:06 classes
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 30 Dec 12 16:02 input.txt
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 2327 Dec 12 16:05 WordCount.java
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$

```

STEP9-SET THE PATH FOR JAVA FILE

COMMAND: `export HADOOP_CLASSPATH=$(hadoop classpath)`

```
echo $HADOOP_CLASSPATH
```

```
Microsoft Teams | Nuupro Web FTP | Subscription Details | Nuupro | kct5thsemcdhid031@ip-10-1-1-1 | YouTube | + |  |  |  | 
```

npbdh login: kct5thsemcdhid031
kct5thsemcdhid031@npbdh.cloudloka.com's password:
Last login: Mon Dec 12 14:48:34 2022 from ec2-65-1-45-35.ap-south-1.compute.amazonaws.com
[kct5thsemcdhid031@ip-10-1-1-284 ~]\$ mkdir wordcount
[kct5thsemcdhid031@ip-10-1-1-284 ~]\$ cd wordcount
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ touch input.txt
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ vi input.txt
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ touch WordCount.java
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ vi WordCount.java
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ vi WordCount.java
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ mkdir classes
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ ll
total 12
drwxrwxr-x 2 kct5thsemcdhid031 kct5thsemcdhid031 4096 Dec 12 16:06 classes
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 30 Dec 12 16:02 input.txt
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 2377 Dec 12 16:05 WordCount.java
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ export HADOOP_CLASSPATH=\$(hadoop classpath)
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ echo \$HADOOP_CLASSPATH
/etc/hadoop/conf:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/.../hadoop/lib/*:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/.../hadoop/..:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/.../hadoop-hdfs/..:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop-hdfs/..:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/.../hadoop-mapreduce/..:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/.../hadoop-yarn/..:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/.../hadoop-yarn/*/
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]\$ javac -classpath \${HADOOP_CLASSPATH} -d '/home/kct5thsemcdhid031/wordcount/classes' '/home/kct5thsemcdhid031/wordcount/WordCount.java'
Note: /home/kct5thsemcdhid031@ip-10-1-1-284 wordcount uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
[kct5thsemcdhid031@ip-10-1-1-284 wordcount]

STEP10-COMPILE THE JAVAFILE

COMMAND: `javac -classpath ${HADOOP_CLASSPATH} -d '/home/<NAME>/wordcount/classes' '/home/<NAME>/wordcount/WordCount.java'`

STEP11-CREATE A JAR FILE

COMMAND: jar -cvf WordCount.jar -C '/home/<NAME>/wordcount/classes' .

STEP12-CREATE A DIRECTORY IN HADOOP

```

Microsoft Teams Nuvelpro Web FTP Subscription Details | Nuvelpro YouTube
https://npbhd.cloudloka.com:4200 kct5thsemcdhid031@ip-10-1-1-204 ~$ cd wordcount
kct5thsemcdhid031@ip-10-1-1-204 ~$ mkdir wordcount
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ cd wordcount
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ touch input.txt
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ vi input.txt
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ touch WordCount.java
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ vi WordCount.java
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ vi WordCount.java
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ mkdir classes
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ ls
total 12
drwxrwxr-x 2 kct5thsemcdhid031 kct5thsemcdhid031 4096 Dec 12 16:06 classes
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 30 Dec 12 16:02 input.txt
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 3237 Dec 12 16:05 WordCount.java
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ export HADOOP_CLASSPATH=$(hadoop classpath)
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ echo $HADOOP_CLASSPATH
/etc/hadoop/conf/.opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/../../opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/../../hadoop-hdfs/.:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/../../opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/../../hadoop-hdfs/lib/*:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/../../hadoop-yarn/lib/*:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/../../hadoop-yarn/./*
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ javac -classpath ${HADOOP_CLASSPATH} -d '/home/kct5thsemcdhid031/wordcount/classes' '/home/kct5thsemcdhid031/wordcount/WordCount.java'
Note: /home/kct5thsemcdhid031/wordcount/WordCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ jar -cvf WordCount.jar -C '/home/kct5thsemcdhid031/wordcount/classes' /
added manifest
adding: WordCount.class(in = 1814) (out= 914)(deflated 49%)
adding: WordCount$Map.class(in = 1657) (out= 691)(deflated 58%)
adding: WordCount$Reduce.class(in = 1627) (out= 686)(deflated 57%)
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ ls
total 16
drwxrwxr-x 2 kct5thsemcdhid031 kct5thsemcdhid031 4096 Dec 12 16:09 classes
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 30 Dec 12 16:02 input.txt
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 3021 Dec 12 16:11 WordCount.jar
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 2327 Dec 12 16:05 WordCount.java
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$

```

STEP13-PUT THE INPUT FILE IN LOCAL SYSTEM TO HADOOP DIRECTORY

STEP14-CREATE A OUTPUT DIRECTORY IN HDFS INSIDE THE WORDCOUNT

```

Microsoft Teams Nuvelpro Web FTP Subscription Details | Nuvelpro YouTube
https://npbhd.cloudloka.com:4200 kct5thsemcdhid031@ip-10-1-1-204 ~$ put "/home/kct5thsemcdhid031/WordCount/input.txt"; No such file or directory
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ hdfs dfs -put /home/kct5thsemcdhid031/wordcount/input.txt /user/kct5thsemcdhid031/wordcount/input/
put: '/user/kct5thsemcdhid031/wordcount/input/input.txt': File exists
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ hdfs dfs -mkdir -p /user/kct5thsemcdhid031/wordcount/output
[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ hadoop jar /home/kct5thsemcdhid031/wordcount/WordCount.jar WordCount /user/kct5thsemcdhid031/wordcount/input /user/kct5thsemcdhid031/wordcount/output
WARNING: Use "yarn jar" to launch YARN applications.
22/12/12 16:23:07 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8832
22/12/12 16:23:07 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
22/12/12 16:23:07 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /user/kct5thsemcdhid031/staging/job_1663041244711_21384
22/12/12 16:23:07 INFO input.FileInputFormat: Total input files to process : 1
22/12/12 16:23:08 INFO mapreduce.JobSubmitter: number of splits:1
22/12/12 16:23:08 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
22/12/12 16:23:08 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1663041244711_21384
22/12/12 16:23:08 INFO mapreduce.JobSubmitter: Executing with tokens: []
22/12/12 16:23:08 INFO conf.Configuration: resource-types.xml not found
22/12/12 16:23:08 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
22/12/12 16:23:08 INFO impl.YarnClientImpl: Submitted application application_1663041244711_21384
22/12/12 16:23:08 INFO mapreduce.Job: The url to track the job: http://ip-10-1-1-204.ap-south-1.compute.internal:8086/proxy/application_1663041244711_21384/
22/12/12 16:23:08 INFO mapreduce.Job: Running job: job_1663041244711_21384
22/12/12 16:23:17 INFO mapreduce.Job: Job job_1663041244711_21384 running in uber mode : false
22/12/12 16:23:17 INFO mapreduce.Job: map 0% reduce 0%
22/12/12 16:23:23 INFO mapreduce.Job: map 100% reduce 0%
22/12/12 16:23:39 INFO mapreduce.Job: map 100% reduce 40%
22/12/12 16:23:40 INFO mapreduce.Job: map 100% reduce 100%
22/12/12 16:23:41 INFO mapreduce.Job: Job job_1663041244711_21384 completed successfully
22/12/12 16:23:41 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=185
FILE: Number of bytes written=1338507

```

STEP15-RUN THE MAP REDUCE PROGRAM

COMMAND: hadoop jar /home/<NAME>/wordcount/WordCount.jar' WordCount /user/<NAME>/wordcount/input /user/<NAME>/wordcount/output

STEP16-VERIFY THE OUTPUT:

COMMAND: hdfs dfs -cat /user/<Name>/wordcount/output/*



```
Map output materialized bytes=165
Input split bytes=133
Combine input records=0
Combine output records=0
Reduce input groups=9
Reduce shuffle bytes=165
Reduce input records=9
Reduce output records=9
Spilled Records=18
Shuffled Maps =5
Failed Shuffles=0
Merged Map outputs=5
GC time elapsed (ms)=1594
CPU time spent (ms)=7270
Physical memory (bytes) snapshot=1878593536
Virtual memory (bytes) snapshot=1559064128
Total committed heap usage (bytes)=2833252352
Peak Map Physical memory (bytes)=495263744
Peak Map Virtual memory (bytes)=2576912384
Peak Reduce Physical memory (bytes)=2790993248
Peak Reduce Virtual memory (bytes)=2611478528

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=34
File Output Format Counters
Bytes Written=50

[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$ hdfs dfs -cat /user/kct5thsemcdhid031/wordcount/output/*
hi      1
sh      1
ag      1
hey     1
ise     1
pr      1
ad      1
ee      1
hello   1

[kct5thsemcdhid031@ip-10-1-1-204 wordcount]$
```

LAB EXCERSICE -4

WEATHER

AIM:

Implement a Map Reduce Program to analyse time-temperature statistics and generate report with max/min temperature

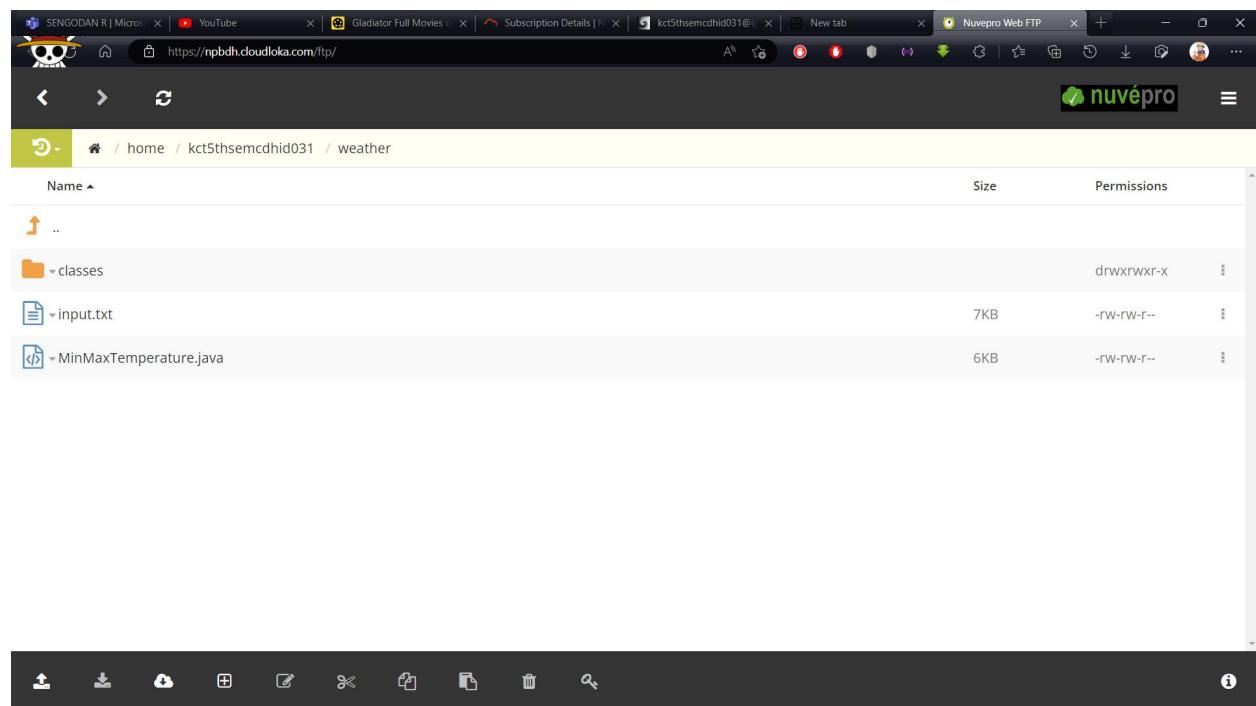
PROCEDURE:

STEP1-LOGIN TO THE WEB SHELL USING THE USERNAME AND PASSWORD

STEP2-CREATE A FOLDER/ DIRECTORY FOR WEATHERIN HOME

STEP3-MOVE TO THAT FOLDER/DIRECTORY

STEP4-CREATE A INPUT FILE AND JAVA FILE INSIDE THE WORDCOUNT FOLDER



STEP5-PASTE THE INPUT IN THE input.txt

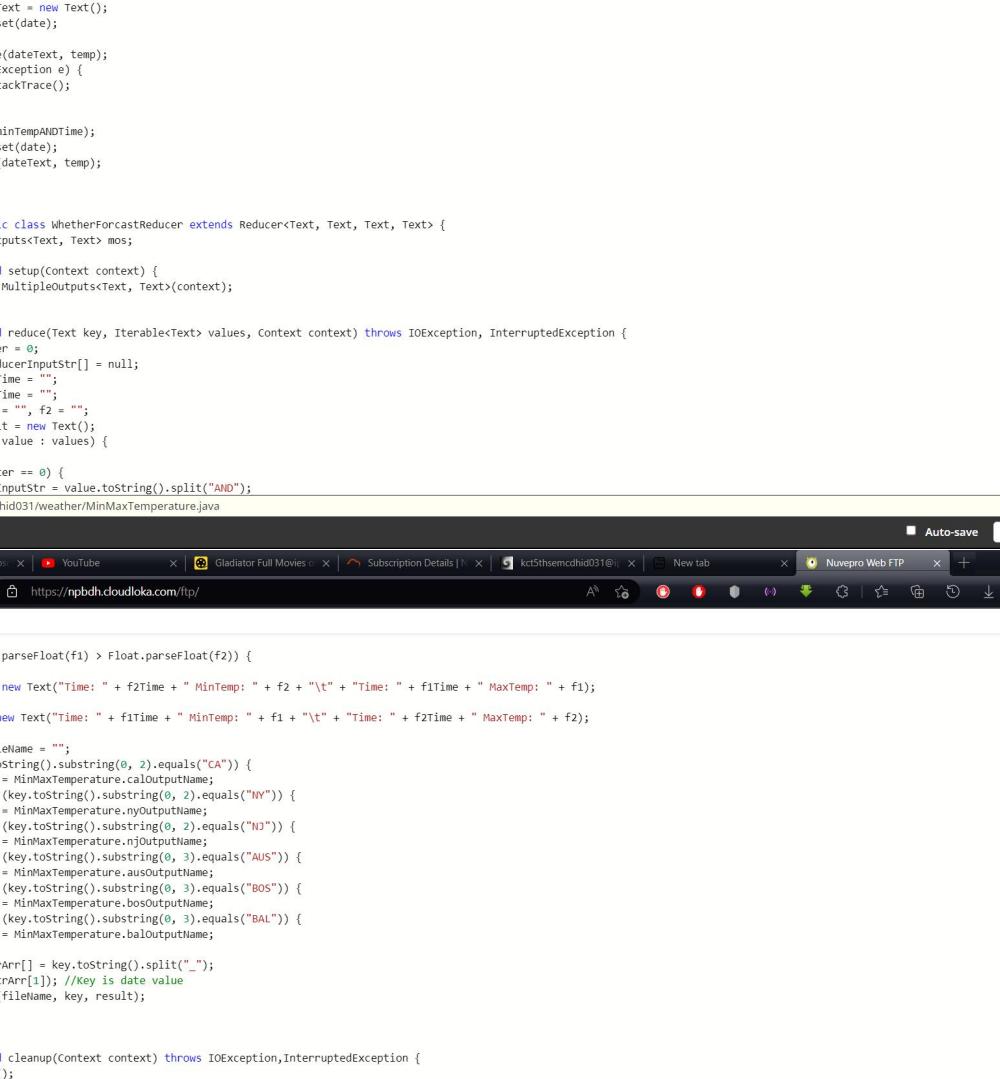
	CA_25-Jan-2014	00:12:345	15.7	01:19:345	23.1	02:34:542	12.3	03:12:187	16	04:00:093	-14	05:12:345	35.7	06:19:345	23.1	07:34:542	12.3	08:12:187	16
1	CA_25-Jan-2014	00:12:345	15.7	01:19:345	23.1	02:34:542	-22.3	03:12:187	16	04:00:093	-7	05:12:345	15.7	06:19:345	23.1	07:34:542	12.3	08:12:187	16
2	CA_26-Jan-2014	00:54:245	15.7	01:19:345	23.1	02:34:542	12.3	03:12:187	16	04:00:093	-14	05:12:345	55.7	06:19:345	23.1	07:34:542	12.3	08:12:187	16
3	CA_27-Jan-2014	00:14:045	35.7	01:19:345	23.1	02:34:542	-22.3	03:12:187	16	04:00:093	-14	05:12:345	35.7	06:19:345	23.1	07:34:542	12.3	08:12:187	16
4	CA_28-Jan-2014	00:22:315	15.7	01:19:345	23.1	02:34:542	12.3	03:12:187	16	04:00:093	-14	05:12:345	35.7	06:19:345	23.1	07:34:542	12.3	08:12:187	16
5	CA_29-Jan-2014	00:15:345	15.7	01:19:345	23.1	02:34:542	52.9	03:12:187	16	04:00:093	-14	05:12:345	45.0	06:19:345	23.1	07:34:542	-2.3	08:12:187	16
6	NJ_29-Jan-2014	00:15:345	15.7	01:19:345	23.1	02:34:542	52.9	03:12:187	16	04:00:093	-14	05:12:345	45.0	06:19:345	23.1	07:34:542	-2.3	08:12:187	16
7	CA_30-Jan-2014	00:22:445	15.7	01:19:345	23.1	02:34:542	12.3	03:12:187	16	04:00:093	-14	05:12:345	35.7	06:19:345	39.6	07:34:542	12.3	08:12:187	16
8	CA_31-Jan-2014	00:42:245	15.7	01:19:345	23.1	02:34:542	12.3	03:12:187	16	04:00:093	-14	05:12:345	49.2	06:19:345	23.1	07:34:542	12.3	08:12:187	16
9	NY_29-Jan-2014	00:15:345	15.7	01:19:345	23.1	02:34:542	52.9	03:12:187	16	04:00:093	-14	05:12:345	45.0	06:19:345	23.1	07:34:542	-2.3	08:12:187	16
10	NY_30-Jan-2014	00:22:445	15.7	01:19:345	23.1	02:34:542	12.3	03:12:187	16	04:00:093	-14	05:12:345	35.7	06:19:345	39.6	07:34:542	12.3	08:12:187	16
11	NY_31-Jan-2014	00:42:245	15.7	01:19:345	23.1	02:34:542	12.3	03:12:187	16	04:00:093	-14	05:12:345	49.2	06:19:345	23.1	07:34:542	12.3	08:12:187	16
12	NJ_30-Jan-2014	00:22:445	15.7	01:19:345	23.1	02:34:542	12.3	03:12:187	16	04:00:093	-14	05:12:345	35.7	06:19:345	39.6	07:34:542	12.3	08:12:187	16
13	AUS_25-Jan-2014	00:12:345	15.7	01:19:345	23.1	02:34:542	12.2	03:12:187	16	04:00:093	-14	05:12:345	35.7	06:19:345	23.1	07:34:542	12.3	08:12:187	16

STEP6-PASTE THE SOURCE CODE IN THE FILE

```

23
24 public static class WhetherForecastMapper extends Mapper<Object, Text, Text, Text> {
25     public void map(Object keyoffset, Text dayReport, Context con) throws IOException, InterruptedException {
26         StringTokenizer strTokens = new StringTokenizer(dayReport.toString(), "\t");
27         StringTokenizer strTokens = new StringTokenizer(dayReport.toString(), "\n");
28         int counter = 0;
29         Float currnetTemp = null;
30         Float minTemp = Float.MAX_VALUE;
31         Float maxTemp = Float.MIN_VALUE;
32         String date = null;
33         String currentTime = null;
34         String minTempANDtime = null;
35         String maxTempANDtime = null;
36
37         while (strTokens.hasMoreElements()) {
38             if (counter == 0) {
39                 date = strTokens.nextToken();
40             } else {
41                 if (counter % 2 == 1) {
42                     currentTime = strTokens.nextToken();
43                 }
44                 else {
45                     currnetTemp = Float.parseFloat(strTokens.nextToken());
46                     /* DecimalFormat df = new DecimalFormat();
47                     currnetTemp = df.parse(strTokens.nextToken()).floatValue(); */
48                     if (minTemp > currnetTemp) {
49                         minTemp = currnetTemp;
50                         minTempANDtime = minTemp + " AND " + currentTime;
51                     }
52                     if (maxTemp < currnetTemp) {
53                         maxTemp = currnetTemp;
54                         maxTempANDtime = maxTemp + " AND " + currentTime;
55                     }
56                 }
57             }
58             counter++;
59         }
60     }
61 }

```



```
59     counter++;
60   }
61   // Write to context - MinTemp, MaxTemp and corresponding time
62   Text temp = new Text();
63   temp.set(maxTempANDtime);
64   Text dateText = new Text();
65   dateText.set(date);
66   try {
67     con.write(dateText, temp);
68   } catch (Exception e) {
69     e.printStackTrace();
70   }
71
72   temp.set(minTempANDtime);
73   dateText.set(date);
74   con.write(dateText, temp);
75 }
76
77 public static class WhetherForcastReducer extends Reducer<Text, Text, Text, Text> {
78   MultipleOutputs<Text, Text> mos;
79
80   public void setup(Context context) {
81     mos = new MultipleOutputs<Text, Text>(context);
82   }
83
84   public void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
85     int counter = 0;
86     String reducerInputStr[] = null;
87     String f1Time = "";
88     String f2Time = "";
89     String f1 = "", f2 = "";
90     Text result = new Text();
91     for (Text value : values) {
92
93       if (counter == 0) {
94         reducerInputStr = value.toString().split("AND");
95       }
96
97       if (Float.parseFloat(f1) > Float.parseFloat(f2)) {
98
99         result = new Text("Time: " + f2Time + " MinTemp: " + f2 + "\t" + "Time: " + f1Time + " MaxTemp: " + f1);
100      } else {
101        result = new Text("Time: " + f1Time + " MinTemp: " + f1 + "\t" + "Time: " + f2Time + " MaxTemp: " + f2);
102      }
103
104      String fileName = "";
105      if (key.toString().substring(0, 2).equals("CA")) {
106        fileName = MinMaxTemperature.caOutputName;
107      } else if (key.toString().substring(0, 2).equals("NY")) {
108        fileName = MinMaxTemperature.nyOutputName;
109      } else if (key.toString().substring(0, 2).equals("NJ")) {
110        fileName = MinMaxTemperature.njOutputName;
111      } else if (key.toString().substring(0, 3).equals("AUS")) {
112        fileName = MinMaxTemperature.ausOutputName;
113      } else if (key.toString().substring(0, 3).equals("BOS")) {
114        fileName = MinMaxTemperature.bosOutputName;
115      } else if (key.toString().substring(0, 3).equals("BAL")) {
116        fileName = MinMaxTemperature.balOutputName;
117      }
118
119      String strArr[] = key.toString().split("_");
120      key.set(strArr[1]); //Key is date value
121      mos.write(fileName, key, result);
122    }
123
124    @Override
125    public void cleanup(Context context) throws IOException, InterruptedException {
126      mos.close();
127    }
128  }
129
130  public static void main(String[] args) throws IOException,
131  ClassNotFoundException, InterruptedException {
132    Configuration conf = new Configuration();
133    Job job = Job.getInstance(conf, "Weather Statistics of USA");
134    job.setJarByClass(MinMaxTemperature.class);
135    job.setMapperClass(Map.class);
136    job.setReducerClass(WhetherForcastReducer.class);
137    job.setOutputKeyClass(Text.class);
138    job.setOutputValueClass(Text.class);
139    FileInputFormat.addInputPath(job, new Path(args[0]));
140    FileOutputFormat.setOutputPath(job, new Path(args[1]));
141    job.waitForCompletion(true);
142  }
143 }
```

```

141 Configuration conf = new Configuration();
142 Job job = Job.getInstance(conf, "weather Statistics of USA");
143 job.setJarByClass(MinMaxTemperature.class);
144
145 job.setMapperClass(whetherForcastMapper.class);
146 job.setReducerClass(whetherForcastReducer.class);
147
148 job.setMapOutputKeyClass(Text.class);
149 job.setMapOutputValueClass(Text.class);
150
151 job.setOutputKeyClass(Text.class);
152 job.setOutputValueClass(Text.class);
153
154 MultipleOutputs.addNamedOutput(job, calOutputName, TextOutputFormat.class, Text.class, Text.class);
155 MultipleOutputs.addNamedOutput(job, nyOutputName, TextOutputFormat.class, Text.class, Text.class);
156 MultipleOutputs.addNamedOutput(job, njOutputName, TextOutputFormat.class, Text.class, Text.class);
157 MultipleOutputs.addNamedOutput(job, bosOutputName, TextOutputFormat.class, Text.class, Text.class);
158 MultipleOutputs.addNamedOutput(job, ausOutputName, TextOutputFormat.class, Text.class, Text.class);
159 MultipleOutputs.addNamedOutput(job, balOutputName, TextOutputFormat.class, Text.class, Text.class);
160
161 // FileInputFormat.addInputPath(job, new Path(args[0]));
162 // FileOutputFormat.setOutputPath(job, new Path(args[1]));
163 Path outputPath = new Path(args[1]);
164 //Configuring the input/output path from the filesystem into the job
165 FileInputFormat.addInputPath(job, new Path(args[0]));
166 FileOutputFormat.setOutputPath(job, new Path(args[1]));
167 //deleting the output path automatically from hdfs so that we don't have to
168 //delete it explicitly
169 outputPath.getFileSystem(conf).delete(outputPath);
170 //exiting the job only if the flag value becomes false
171 try {
172     System.exit(job.waitForCompletion(true) ? 0 : 1);
173 } catch (Exception e) {
174     // TODO Auto-generated catch block
175     e.printStackTrace();
176 }

```

▲ /home/kct5thsemcdhid031/weather/MinMaxTemperature.java

nuvépro

■ Auto-save Save Close

SOURCE CODE:

```

import java.io.IOException;
import java.util.StringTokenizer;
import java.text.DecimalFormat;

import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.output.MultipleOutputs;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class MinMaxTemperature {
    public static String calOutputName = "California";
    public static String nyOutputName = "Newyork";
    public static String njOutputName = "Newjersy";
    public static String ausOutputName = "Austin";
}

```

```
public static String bosOutputName = "Boston";
public static String balOutputName = "Baltimore";

public static class WhetherForcastMapper extends Mapper<Object, Text, Text, Text> {
    public void map(Object keyOffset, Text dayReport, Context con) throws IOException,
    InterruptedException {
        //StringTokenizer strTokens = new StringTokenizer(dayReport.toString(), "\t");
        StringTokenizer strTokens = new StringTokenizer(dayReport.toString(),"\t");
        int counter = 0;
        Float currnetTemp = null;
        Float minTemp = Float.MAX_VALUE;
        Float maxTemp = Float.MIN_VALUE;
        String date = null;
        String currentTime = null;
        String minTempANDTime = null;
        String maxTempANDTime = null;

        while (strTokens.hasMoreElements()) {
            if (counter == 0) {
                date = strTokens.nextToken();
            } else {
                if (counter % 2 == 1) {
                    currentTime = strTokens.nextToken();
                }
            }
            else {
                currnetTemp = Float.parseFloat(strTokens.nextToken());
                /* DecimalFormat df = new DecimalFormat();
                currnetTemp = df.parse(strTokens.nextToken()).floatValue(); */
                if (minTemp > currnetTemp) {
                    minTemp = currnetTemp;
                    minTempANDTime = minTemp + "AND" + currentTime;
                }
                if (maxTemp < currnetTemp) {
                    maxTemp = currnetTemp;
                    maxTempANDTime = maxTemp + "AND" + currentTime;
                }
            }
            counter++;
        }
        // Write to context - MinTemp, MaxTemp and corresponding time
        Text temp = new Text();
        temp.set(maxTempANDTime);
        Text dateText = new Text();
```

```

dateText.set(date);
try {
con.write(dateText, temp);
} catch (Exception e) {
e.printStackTrace();
}
temp.set(minTempANDTime);
dateText.set(date);
con.write(dateText, temp);
}
}

public static class WhetherForcastReducer extends Reducer<Text, Text, Text, Text> {
MultipleOutputs<Text, Text> mos;

public void setup(Context context) {
mos = new MultipleOutputs<Text, Text>(context);
}

public void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
InterruptedException {
int counter = 0;
String reducerInputStr[] = null;
String f1Time = "";
String f2Time = "";
String f1 = "", f2 = "";
Text result = new Text();
for (Text value : values) {
if (counter == 0) {
reducerInputStr = value.toString().split("AND");
f1 = reducerInputStr[0];
f1Time = reducerInputStr[1];
}
else {
reducerInputStr = value.toString().split("AND");
f2 = reducerInputStr[0];
f2Time = reducerInputStr[1];
}
counter = counter + 1;
}
if (Float.parseFloat(f1) > Float.parseFloat(f2)) {

```

```

result = new Text("Time: " + f2Time + " MinTemp: " + f2 + "\t" + "Time: " + f1Time + " MaxTemp:
" + f1);
} else {
result = new Text("Time: " + f1Time + " MinTemp: " + f1 + "\t" + "Time: " + f2Time + " MaxTemp:
" + f2);
}
String fileName = "";
if (key.toString().substring(0, 2).equals("CA")) {
fileName = MinMaxTemperature.caOutputName;
} else if (key.toString().substring(0, 2).equals("NY")) {
fileName = MinMaxTemperature.nyOutputName;
} else if (key.toString().substring(0, 2).equals("NJ")) {
fileName = MinMaxTemperature.njOutputName;
} else if (key.toString().substring(0, 3).equals("AUS")) {
fileName = MinMaxTemperature.ausOutputName;
} else if (key.toString().substring(0, 3).equals("BOS")) {
fileName = MinMaxTemperature.bosOutputName;
} else if (key.toString().substring(0, 3).equals("BAL")) {
fileName = MinMaxTemperature.balOutputName;
}
String strArr[] = key.toString().split("_");
key.set(strArr[1]); //Key is date value
mos.write(fileName, key, result);
}

@Override
public void cleanup(Context context) throws IOException, InterruptedException {
mos.close();
}

public static void main(String[] args) throws IOException,
ClassNotFoundException, InterruptedException {
Configuration conf = new Configuration();
Job job = Job.getInstance(conf, "Wheather Statistics of USA");
job.setJarByClass(MinMaxTemperature.class);

job.setMapperClass(WhetherForcastMapper.class);
job.setReducerClass(WhetherForcastReducer.class);

job.setMapOutputKeyClass(Text.class);
job.setMapOutputValueClass(Text.class);

job.setOutputKeyClass(Text.class);

```

```

job.setOutputValueClass(Text.class);

MultipleOutputs.addNamedOutput(job, calOutputName, TextOutputFormat.class, Text.class,
Text.class);
MultipleOutputs.addNamedOutput(job, nyOutputName, TextOutputFormat.class, Text.class,
Text.class);
MultipleOutputs.addNamedOutput(job, njOutputName, TextOutputFormat.class, Text.class,
Text.class);
MultipleOutputs.addNamedOutput(job, bosOutputName, TextOutputFormat.class, Text.class,
Text.class);
MultipleOutputs.addNamedOutput(job, ausOutputName, TextOutputFormat.class, Text.class,
Text.class);
MultipleOutputs.addNamedOutput(job, balOutputName, TextOutputFormat.class, Text.class,
Text.class);

// FileInputFormat.addInputPath(job, new Path(args[0]));
// FileOutputFormat.setOutputPath(job, new Path(args[1]));
Path outputPath = new Path(args[1]);
//Configuring the input/output path from the filesystem into the job
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
//deleting the output path automatically from hdfs so that we don't have to
//delete it explicitly
outputPath.getFileSystem(conf).delete(outputPath);
//exiting the job only if the flag value becomes false
try {
System.exit(job.waitForCompletion(true) ? 0 : 1);
} catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}

}

```

STEP7-CREATE A DIRECTORY/ FOLDER CLASSES INSIDE THE WEATHER FOLDER

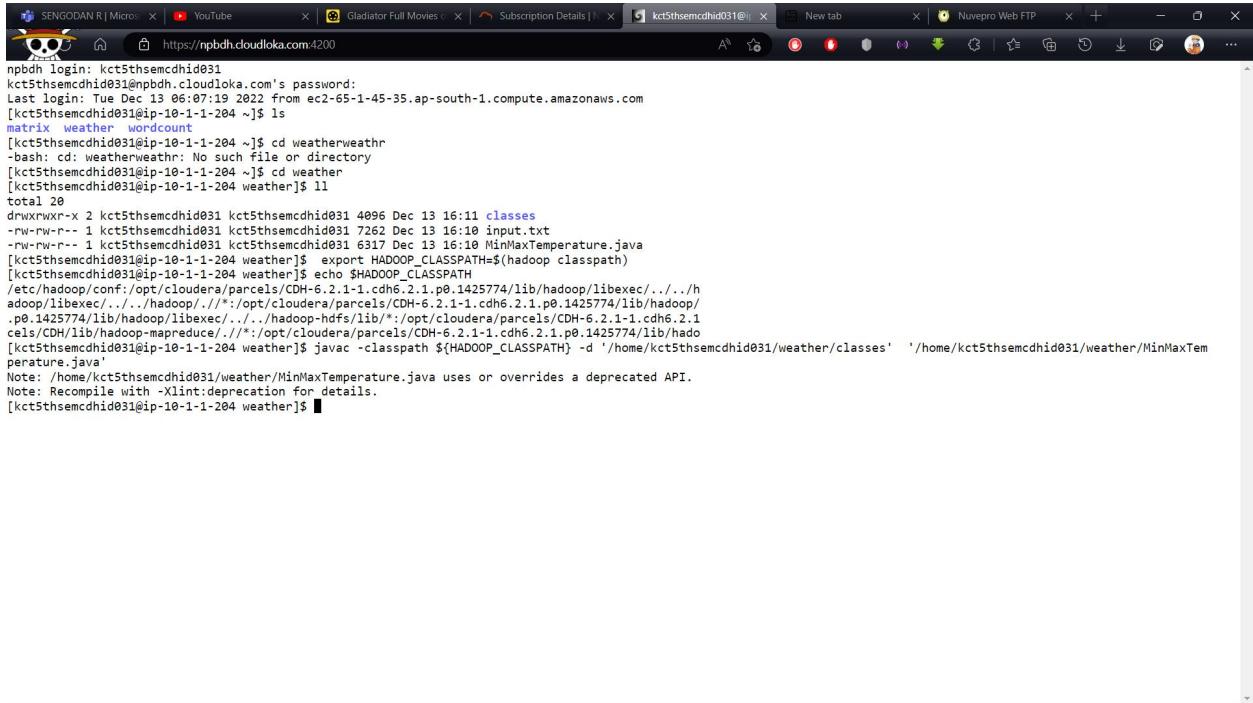
STEP8-SET THE PATH FOR JAVA FILE

COMMAND: export HADOOP_CLASSPATH=\$(hadoop classpath)

```
echo $HADOOP_CLASSPATH
```

STEP9-COMPILE THE JAVAFILE

COMMAND: javac -classpath \${HADOOP_CLASSPATH} -d '/home/<NAME>/wordcount/classes' '/home/<NAME>/wordcount/WordCount.java'

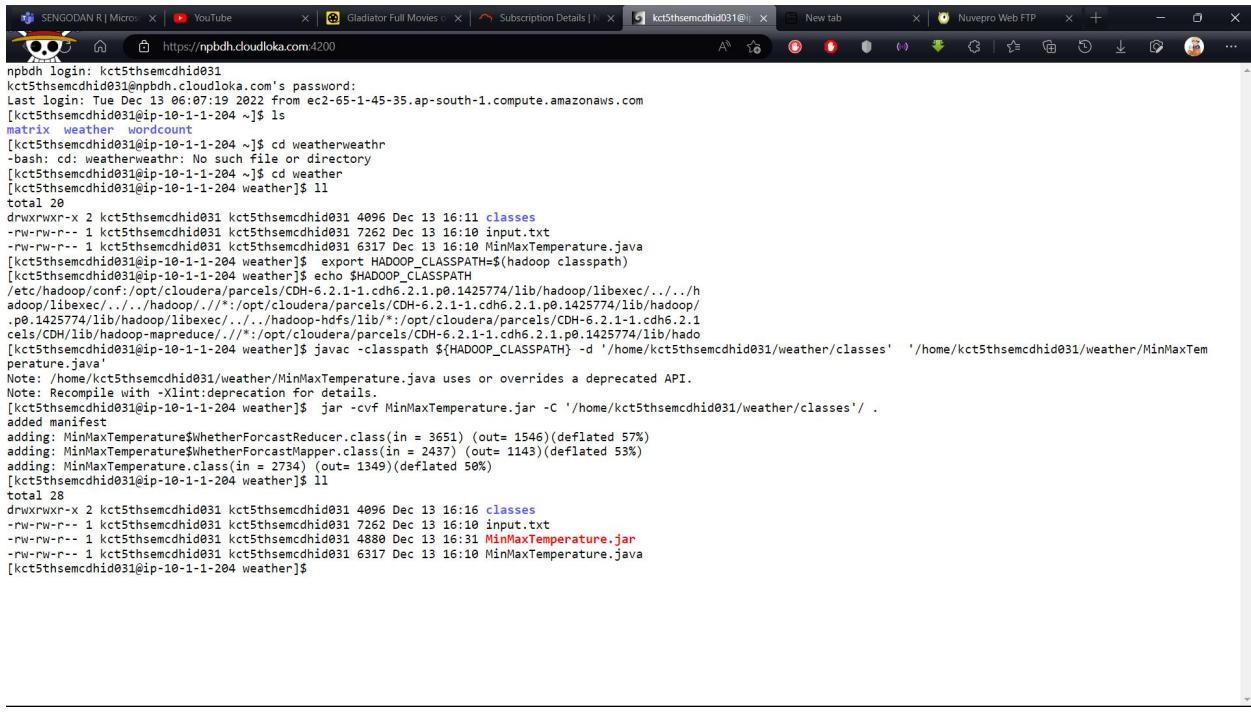


The screenshot shows a terminal window with a black background and white text. It displays the following command and its execution:

```
npbdh:~ login: kct5thsemcdhid031
kct5thsemcdhid031@npbdh.cloudloka.com's password:
Last login: Tue Dec 13 06:07:19 2022 from ec2-65-1-45-35.ap-south-1.compute.amazonaws.com
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ ls
matrix weather wordcount
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ cd weatherweather
-bash: cd: weatherweather: No such file or directory
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ cd weather
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ ls
total 20
drwxrwxr-x 2 kct5thsemcdhid031 kct5thsemcdhid031 4096 Dec 13 16:11 classes
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 7262 Dec 13 16:10 input.txt
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 6317 Dec 13 16:10 MinMaxTemperature.java
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ export HADOOP_CLASSPATH=$(hadoop classpath)
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ echo $HADOOP_CLASSPATH
/etc/hadoop/conf:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/../../hadoop/libexec/../../hadoop/libexec/../../../../hadoop-hdfs/lib/*:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1/cells/CDH-6.2.1-1.hdfs.mapreduce/../../../../hadoop/libexec/../../../../hadoop-hdfs/lib/*:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/[kct5thsemcdhid031@ip-10-1-1-204 weather]$ javac -classpath ${HADOOP_CLASSPATH} -d '/home/kct5thsemcdhid031/weather/classes' '/home/kct5thsemcdhid031/weather/MinMaxTemperature.java'
Note: /home/kct5thsemcdhid031/weather/MinMaxTemperature.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
[kct5thsemcdhid031@ip-10-1-1-204 weather]$
```

STEP10-CREATE A JAR FILE

COMMAND: jar -cvf WordCount.jar -C '/home/<NAME>/wordcount/classes' .



```
SENGODAN R | Micro: | YouTube | Gladiator Full Movies | Subscription Details | kct5thsemcdhid031@ip-10-1-1-204 ~ | New tab | Nuvelpro Web FTP | + | - | × |
```

```
pbdh login: kct5thsemcdhid031
kct5thsemcdhid031@npbdh.cloudloka.com's password:
Last login: Tue Dec 13 06:07:19 2022 from ec2-65-1-45-35.ap-south-1.compute.amazonaws.com
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ ls
matrix weather wordcount
[kct5thsemcdhid031@ip-10-1-1-204 ~]$ cd weather
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ ll
total 20
drwxrwxr-x 2 kct5thsemcdhid031 kct5thsemcdhid031 4096 Dec 13 16:11 classes
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 7262 Dec 13 16:10 input.txt
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 6317 Dec 13 16:10 MinMaxTemperature.java
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ export HADOOP_CLASSPATH=$(/usr/lib/hadoop/bin/hadoop classpath)
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ echo $HADOOP_CLASSPATH
/etc/hadoop/conf:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/libexec/../../hadoop/libexec/../../../../opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/p0.1425774/lib/hadoop/libexec/../../../../opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/p0.1425774/lib/hadoop-hdfs/lib/../../../../opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop-mapreduce/../../../../opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop-mapreduce/lib
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ javac -classpath ${HADOOP_CLASSPATH} -d '/home/kct5thsemcdhid031/weather/classes' '/home/kct5thsemcdhid031/weather/MinMaxTemperature.java'
Note: /home/kct5thsemcdhid031/weather/MinMaxTemperature.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ jar -cvf MinMaxTemperature.jar -C '/home/kct5thsemcdhid031/weather/classes' .
added manifest
adding: MinMaxTemperature$WhetherForcastReducer.class(in = 3651) (out= 1546)(deflated 57%)
adding: MinMaxTemperature$WhetherForcastMapper.class(in = 2437) (out= 1143)(deflated 53%)
adding: MinMaxTemperature.class(in = 2734) (out= 1349)(deflated 50%)
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ ll
total 28
drwxrwxr-x 2 kct5thsemcdhid031 kct5thsemcdhid031 4096 Dec 13 16:16 classes
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 7262 Dec 13 16:10 input.txt
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 4880 Dec 13 16:31 MinMaxTemperature.jar
-rw-rw-r-- 1 kct5thsemcdhid031 kct5thsemcdhid031 6317 Dec 13 16:10 MinMaxTemperature.java
[kct5thsemcdhid031@ip-10-1-1-204 weather]$
```

STEP11-CREATE A DIRECTORY IN HADOOP

STEP12-CREATE AND PUT THE INPUT FILE IN LOCAL SYSTEM TO HADOOP DIRECTORY

STEP13-CREATE A OUTPUT DIRECTORY IN HDFS INSIDE THE WEATHER

STEP14-RUN THE MAP REDUCE PROGRAM

COMMAND: hadoop jar /home/<NAME>/wordcount/WordCount.jar' WordCount /user/<NAME>/wordcount/input /user/<NAME>/wordcount/output

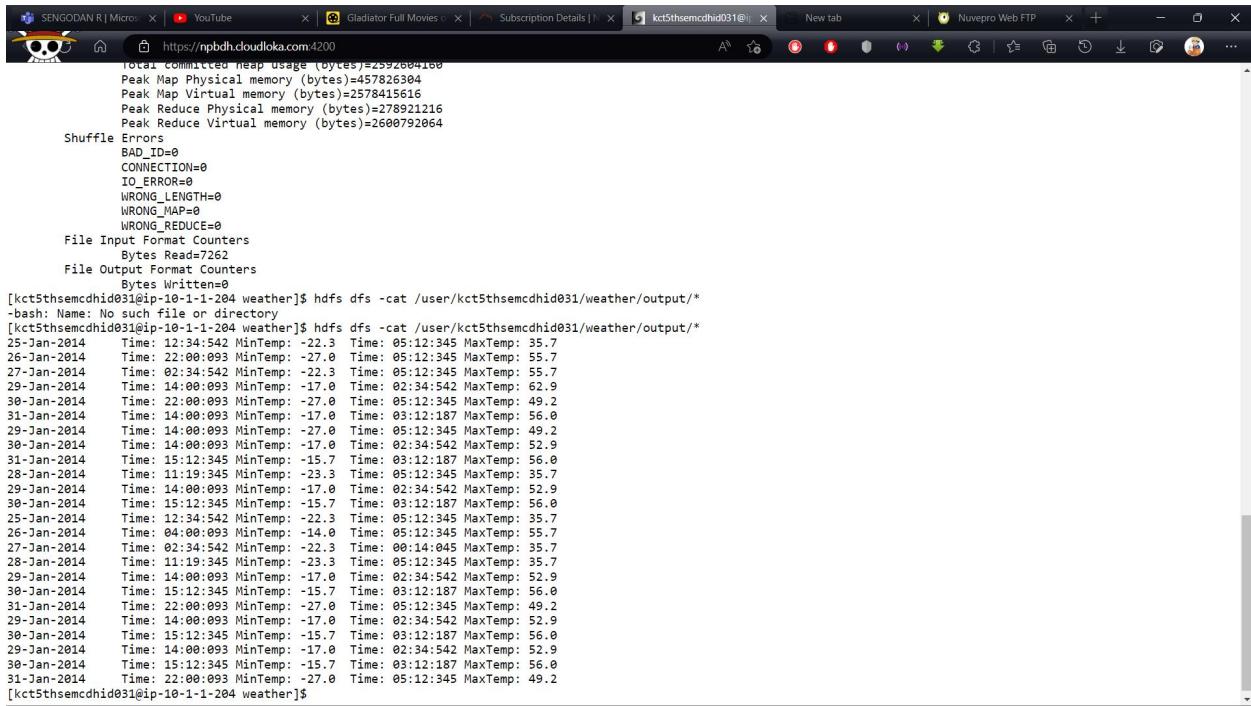
```

SENGODAN R | Micro: https://npbdh.cloudloka.com:4200
YouTube | Gladiator Full Movies | Subscription Details | kct5thsemcdhid031@ip | New tab | Nuvero Web FTP | ...
31/weather/output
[kct5thsemcdhid031@ip-10-1-1-204 weather]$ hadoop jar /home/kct5thsemcdhid031/weather/MinMaxTemperature.jar MinMaxTemperature /user/kct5thsemcdhid031/weather/input /user/kct5thsemcdhid031/weather/output
WARNING: Use "yarn jar" to launch YARN applications.
22/12/13 16:41:45 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
22/12/13 16:41:46 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
22/12/13 16:41:46 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /user/kct5thsemcdhid031/.staging/job_1663041244711_22058
22/12/13 16:41:46 INFO input.FileInputFormat: Total input files to process : 1
22/12/13 16:41:46 INFO mapreduce.JobSubmitter: number of splits:1
22/12/13 16:41:46 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
22/12/13 16:41:47 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1663041244711_22058
22/12/13 16:41:47 INFO mapreduce.JobSubmitter: Executing with tokens: []
22/12/13 16:41:47 INFO conf.Configuration: resource-types.xml not found
22/12/13 16:41:47 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
22/12/13 16:41:47 INFO impl.YarnClientImpl: Submitted application application_1663041244711_22058
22/12/13 16:41:47 INFO mapreduce.Job: The url to track the job: http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1663041244711_22058/
22/12/13 16:42:32 INFO mapreduce.Job: Job job_1663041244711_22058 running in uber mode : false
22/12/13 16:42:32 INFO mapreduce.Job: map 0% reduce 0%
22/12/13 16:43:09 INFO mapreduce.Job: map 100% reduce 0%
22/12/13 16:43:23 INFO mapreduce.Job: map 100% reduce 100%
22/12/13 16:43:25 INFO mapreduce.Job: Job job_1663041244711_22058 completed successfully
22/12/13 16:43:25 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=906
FILE: Number of bytes written=1364303
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=7393
HDFS: Number of bytes written=1752
HDFS: Number of read operations=49
HDFS: Number of large read operations=0
HDFS: Number of write operations=34
HDFS: Number of bytes read erasure-coded=0
Job Counters
Launched map tasks=1
Launched reduce tasks=5
Data-local map tasks=1
Total time spent by all maps in occupied slots (ms)=32288
Total time spent by all reduces in occupied slots (ms)=57813
Total time spent by all map tasks (ms)=32288
Map-Reduce Framework
Total time spent by all reduces in occupied slots (ms)=57813
Total time spent by all map tasks (ms)=32288
Total time spent by all reduce tasks (ms)=57813
Total vcore-milliseconds taken by all map tasks=32288
Total vcore-milliseconds taken by all reduce tasks=57813
Total megabyte-milliseconds taken by all map tasks=33062912
Total megabyte-milliseconds taken by all reduce tasks=59200512
Map-Reduce Framework
Map input records=24
Map output records=48
Map output bytes=1584
Map output materialized bytes=886
Input split bytes=131
Combine input records=0
Combine output records=0
Reduce input groups=24
Reduce input bytes=886
Reduce input records=48
Reduce output records=0
Spilled Records=96
Shuffled Maps =5
Failed Shuffles=0
Merged Map outputs=5
GC time elapsed (ms)=1640
CPU time spent (ms)=8140
Physical memory (bytes) snapshot=1813049344
Virtual memory (bytes) snapshot=15573565440
Total committed heap usage (bytes)=2592604160
Peak Map Physical memory (bytes)=457826304
Peak Map Virtual memory (bytes)=2578415616
Peak Reduce Physical memory (bytes)=278921216
Peak Reduce Virtual memory (bytes)=2600792064
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=7262
File Output Format Counters
Bytes Written=0
[kct5thsemcdhid031@ip-10-1-1-204 weather]$

```

STEP15-VERIFY THE OUTPUT:

COMMAND: hdfs dfs -cat /user/<Name>/wordcount/output/*



The screenshot shows a web browser window with multiple tabs open. The active tab displays a terminal session on a Linux system. The session starts with memory usage statistics:

```
total committed heap usage (bytes)=2592684100
Peak Map Physical memory (bytes)=457826304
Peak Map Virtual memory (bytes)=2578415616
Peak Reduce Physical memory (bytes)=278921216
Peak Reduce Virtual memory (bytes)=2660792064
```

It then lists various error counters for shuffle operations:

```
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
```

File input and output counters are shown next:

```
File Input Format Counters
Bytes Read=7262
File Output Format Counters
Bytes Written=0
```

The session continues with a command to list files in a directory:

```
[kct5thsemcdhid031@ip-10-1-1-284 weather]$ hdfs dfs -cat /user/kct5thsemcdhid031/weather/output/*
-bash: Name: No such file or directory
```

Then it lists files again:

```
[kct5thsemcdhid031@ip-10-1-1-284 weather]$ hdfs dfs -cat /user/kct5thsemcdhid031/weather/output/*
25-Jan-2014 Time: 12:34:542 MintTemp: -22.3 Time: 05:12:345 MaxTemp: 35.7
26-Jan-2014 Time: 22:00:093 MintTemp: -27.0 Time: 05:12:345 MaxTemp: 55.7
27-Jan-2014 Time: 02:34:542 MintTemp: -22.3 Time: 05:12:345 MaxTemp: 55.7
29-Jan-2014 Time: 14:00:093 MintTemp: -17.0 Time: 02:34:542 MaxTemp: 62.9
30-Jan-2014 Time: 22:00:093 MintTemp: -27.0 Time: 05:12:345 MaxTemp: 49.2
31-Jan-2014 Time: 14:00:093 MintTemp: -17.0 Time: 03:12:187 MaxTemp: 56.0
29-Jan-2014 Time: 14:00:093 MintTemp: -27.0 Time: 05:12:345 MaxTemp: 49.2
30-Jan-2014 Time: 14:00:093 MintTemp: -17.0 Time: 02:34:542 MaxTemp: 52.9
31-Jan-2014 Time: 15:12:345 MintTemp: -15.7 Time: 03:12:187 MaxTemp: 56.0
28-Jan-2014 Time: 11:19:345 MintTemp: -23.3 Time: 05:12:345 MaxTemp: 35.7
29-Jan-2014 Time: 14:00:093 MintTemp: -17.0 Time: 02:34:542 MaxTemp: 52.9
30-Jan-2014 Time: 15:12:345 MintTemp: -15.7 Time: 03:12:187 MaxTemp: 52.9
30-Jan-2014 Time: 15:12:345 MintTemp: -15.7 Time: 03:12:187 MaxTemp: 56.0
25-Jan-2014 Time: 12:34:542 MintTemp: -22.3 Time: 05:12:345 MaxTemp: 35.7
26-Jan-2014 Time: 04:00:093 MintTemp: -14.0 Time: 05:12:345 MaxTemp: 55.7
27-Jan-2014 Time: 02:34:542 MintTemp: -22.3 Time: 06:14:045 MaxTemp: 35.7
28-Jan-2014 Time: 11:19:345 MintTemp: -23.3 Time: 05:12:345 MaxTemp: 35.7
29-Jan-2014 Time: 14:00:093 MintTemp: -17.0 Time: 02:34:542 MaxTemp: 52.9
30-Jan-2014 Time: 15:12:345 MintTemp: -15.7 Time: 03:12:187 MaxTemp: 56.0
31-Jan-2014 Time: 22:00:093 MintTemp: -27.0 Time: 05:12:345 MaxTemp: 49.2
29-Jan-2014 Time: 14:00:093 MintTemp: -17.0 Time: 02:34:542 MaxTemp: 52.9
30-Jan-2014 Time: 15:12:345 MintTemp: -15.7 Time: 03:12:187 MaxTemp: 56.0
29-Jan-2014 Time: 14:00:093 MintTemp: -17.0 Time: 02:34:542 MaxTemp: 52.9
30-Jan-2014 Time: 15:12:345 MintTemp: -15.7 Time: 03:12:187 MaxTemp: 56.0
31-Jan-2014 Time: 22:00:093 MintTemp: -27.0 Time: 05:12:345 MaxTemp: 49.2
```

The session ends with a final command:

```
[kct5thsemcdhid031@ip-10-1-1-284 weather]$
```

LAB EXERCISE-8

SPARK

AIM:

To Perform simple join using Mapper in Spark.

THEORY:

Spark DataFrame supports all basic SQL Join Types like INNER, LEFT OUTER, RIGHT OUTER, LEFT ANTI, LEFT SEMI, CROSS,

SELF JOIN. Spark SQL Joins are wider transformations that result in data shuffling over the network hence they have huge performance issues when not designed with care.

INSTALLATION OF SPARK

```
hadoop@ip-172-31-22-77: ~
File Edit View Search Terminal Help
Unpacking scala-xml (1.0.3-3) ...
Selecting previously unselected package scala.
Preparing to unpack .../10-scala_2.11.12-4_all.deb ...
Unpacking scala (2.11.12-4) ...
Setting up scala-library (2.11.12-4) ...
Setting up scala-xml (1.0.3-3) ...
Setting up scala-parser-combinators (1.0.3-3) ...
Setting up git (1:2.25.1-1ubuntu3.6) ...
Setting up libcurl4-openssl4 (7.68.0-1ubuntu2.14) ...
Setting up curl (7.68.0-1ubuntu2.14) ...
Setting up libhawtjni-runtime-java (1.17-1) ...
Setting up libjansi-native-java (1.8-1) ...
Setting up libjansi-java (1.18-1) ...
Setting up libjline2-java (2.14.6-3) ...
Setting up scala (2.11.12-4) ...
update-alternatives: using /usr/share/scala-2.11/bin/scala to provide /usr/bin/scala (scala) in auto mode
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
hadoop@ip-172-31-22-77:~$ curl -O https://archive.apache.org/dist/spark/spark-3.2.0/spark-3.2.0-bin-hadoop3.2.tgz
Total  Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
4 287M  4 12.4M    0     0 1621k   0 0:03:01 0:00:07 0:02:54 1830k
                         0:00:00 0:00:00 0:00:00
more required packages.
$ sudo apt install curl mlocate git scala -y
Download Apache Spark. Find the latest release from the downloads page.
$ curl -O https://archive.apache.org/dist/spark/spark-3.2.0/spark-3.2.0-bin-hadoop3.2.tgz
Extract the Spark tarball.
$ sudo tar xvf spark-3.2.0-bin-hadoop3.2.tgz
```

Was this article helpful?

Try Vultr Today with
\$50 Free on Us!

```
Preparing to unpack .../03-libhawtjni-runtime-java_1.17-1_all.deb ...
Unpacking libhawtjni-runtime-java (1.17-1) ...
Selecting previously unselected package libjansi-native-java.
Preparing to unpack .../04-libjansi-native-java_1.8-1_all.deb ...
Unpacking libjansi-native-java (1.8-1) ...
Selecting previously unselected package libjansi-java.
Preparing to unpack .../05-libjansi-java_1.18-1_all.deb ...
Unpacking libjansi-java (1.18-1) ...
Selecting previously unselected package libjline2-java.
Preparing to unpack .../06-libjline2-java_2.14.6-3_all.deb ...
Unpacking libjline2-java (2.14.6-3) ...
Selecting previously unselected package scala-library.
Preparing to unpack .../07-scala-library_2.11.12-4_all.deb ...
Unpacking scala-library (2.11.12-4) ...
Selecting previously unselected package scala-parser-combinators.
Preparing to unpack .../08-scala-parser-combinators_1.0.3-3_all.deb ...
Unpacking scala-parser-combinators (1.0.3-3) ...
Selecting previously unselected package scala-xml.
Preparing to unpack .../09-scala-xml_1.0.3-3_all.deb ...
Unpacking scala-xml (1.0.3-3) ...
Selecting previously unselected package scala.
Preparing to unpack .../10-scala_2.11.12-4_all.deb ...
Unpacking scala (2.11.12-4) ...
Setting up scala-library (2.11.12-4) ...
Setting up scala-xml (1.0.3-3) ...
Setting up scala-parser-combinators (1.0.3-3) ...
Setting up git (1:2.25.1-1ubuntu3.6) ...
Setting up libcurl4:amd64 (7.68.0-1ubuntu2.14) ...
Setting up curl (7.68.0-1ubuntu2.14) ...
Setting up libhawtjni-runtime-java (1.17-1) ...
Setting up libjansi-native-java (1.8-1) ...
Setting up libjansi-java (1.18-1) ...
Setting up libjline2-java (2.14.6-3) ...
Setting up scala (2.11.12-4) ...
update-alternatives: using /usr/share/scala-2.11/bin/scala to provide /usr/bin/scala (scala) in auto mode
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
hadoop@ip-172-31-22-77:~$ curl -O https://archive.apache.org/dist/spark/spark-3.2.0/spark-3.2.0-bin-hadoop3.2.tgz
```

JOIN QUERY:

Queries can access multiple tables at once, or access the same table in such a way that multiple rows of the table are being processed at the same time. A query that accesses multiple rows of the same or different tables at one time is called a join query.

Parameters

Relation:Specifies the relation to be joined. join_type:Specifies the join type.

Syntax:

```
[ INNER ] | CROSS | LEFT [ OUTER ] | [ LEFT ] SEMI | RIGHT [ OUTER ] | FULL [ OUTER ] | [ LEFT ]  
ANTI
```

join_criteria :Specifies how the rows from one relation will be combined with the rows of another relation.

Syntax:

```
ON boolean_expression | USING ( column_name [ , ... ] )
```

boolean_expression: Specifies an expression with a return type of boolean.

join Operators

```
join(right: Dataset[_]): DataFrame (1)
```

```
join(right: Dataset[_], usingColumn: String): DataFrame (2)
```

```
join(right: Dataset[_], usingColumns: Seq[String]): DataFrame (3)
```

```
join(right: Dataset[_], usingColumns: Seq[String], joinType: String): DataFrame (4)
```

```
join(right: Dataset[_], joinExprs: Column): DataFrame (5)
```

```
join(right: Dataset[_], joinExprs: Column, joinType: String): DataFrame (6) Condition-less inner  
join
```

Inner join with a single column that exists on both sides Inner join with columns that exist on both sides

Equi-join with explicit join type Inner join

Join with explicit join type. Self-joins are acceptable.

1.INNER JOIN:

The inner join is the default join in Spark SQL. It selects rows that have matching values in both relations.

Syntax:

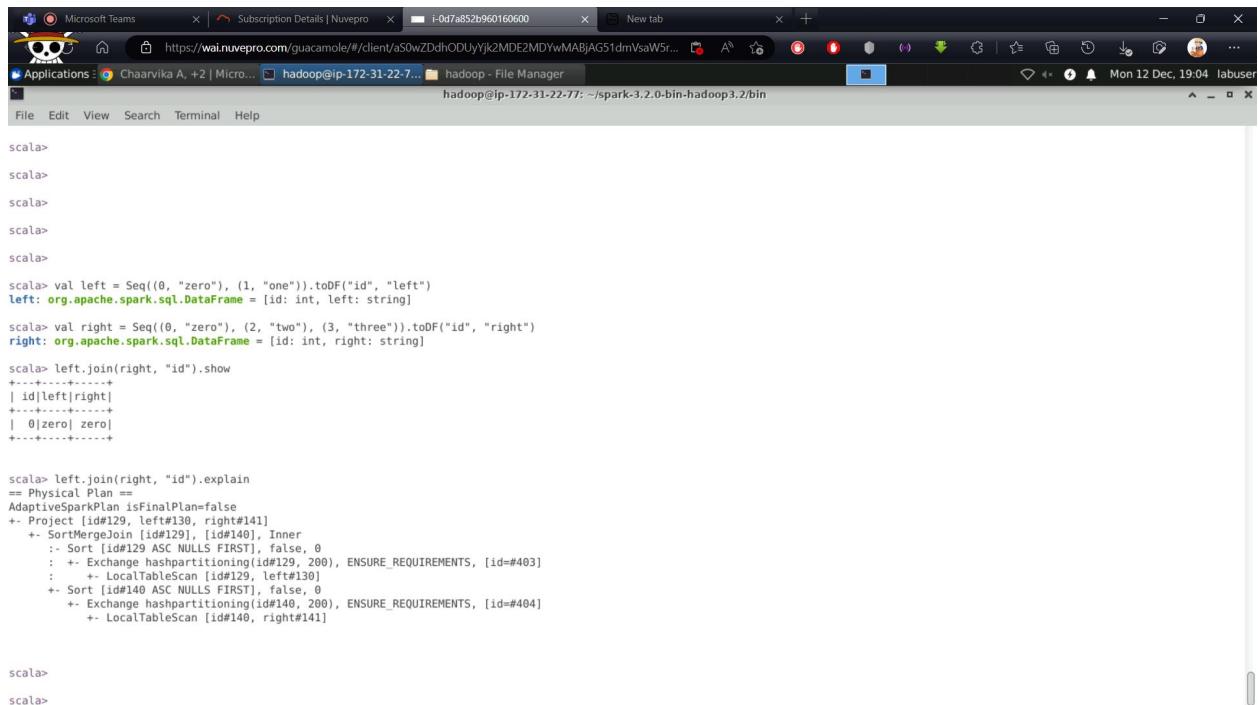
```
relation [ INNER ] JOIN relation [ join_criteria ]
```

Commands

```
1.val left = Seq((0, "zero"), (1, "one")).toDF("id", "left")
```

```
val right = Seq((0, "zero"), (2, "two"), (3, "three")).toDF("id", "right") 3.left.join(right, "id").show
```

```
left.join(right, "id").explain
```



A screenshot of a terminal window titled 'hadoop@ip-172-31-22-77: ~/spark-3.2.0-bin-hadoop3.2/bin'. The window shows Scala code being run. The code defines two DataFrames, 'left' and 'right', and then performs an inner join on 'id'. Finally, it calls 'explain' on the joined DataFrame to show the physical plan. The output shows a detailed logical plan with various stages like SortMergeJoin, Sort, Exchange, and LocalTableScan.

```
scala>
scala>
scala>
scala>
scala>
scala> val left = Seq((0, "zero"), (1, "one")).toDF("id", "left")
left: org.apache.spark.sql.DataFrame = [id: int, left: string]
scala> val right = Seq((0, "zero"), (2, "two"), (3, "three")).toDF("id", "right")
right: org.apache.spark.sql.DataFrame = [id: int, right: string]
scala> left.join(right, "id").show
+---+---+
| id|left|right|
+---+---+
| 0|zero| zero|
+---+---+
scala> left.join(right, "id").explain
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- Project [id#129, left#130, right#141]
  +- SortMergeJoin [id#129], [id#140], Inner
    :- Sort [id#129 ASC NULLS FIRST], false, 0
    :  +- Exchange hashpartitioning(id#129, 200), ENSURE_REQUIREMENTS, [id#403]
    :    +- LocalTableScan [id#129, left#130]
    +- Sort [id#140 ASC NULLS FIRST], false, 0
       +- Exchange hashpartitioning(id#140, 200), ENSURE_REQUIREMENTS, [id#404]
          +- LocalTableScan [id#140, right#141]

scala>
scala>
```

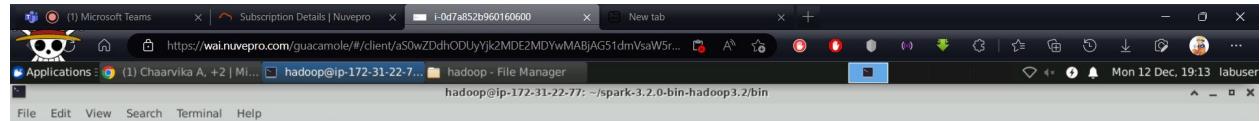
2.FULL OUTER JOIN:

In SQL the FULL OUTER JOIN combines the results of both left and right outer joins and returns all (matched or unmatched) rows from the tables on both sides of the join clause.

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

Commands

```
left.join(right, Seq("id"), "fullouter").show  
left.join(right, Seq("id"), "fullouter").explain
```



```
scala> left.join(right, Seq("id"), "fullouter").show
+---+---+---+
| id|left|right|
+---+---+---+
| 0|zero| zero|
| 1|one| null|
| 2|null| two|
| 3|null|three|
+---+---+---+

scala> left.join(right, Seq("id"), "fullouter").explain
= Physical Plan =
AdaptiveSparkPlan isFinalPlan=false
+- Project [coalesce(id#129, id#140) AS id#180, left#130, right#141]
   +- SortMergeJoin [id#129, id#140], FullOuter
     :- Sort [id#129 ASC NULLS FIRST, false, 0]
     :  +- Exchange hashpartitioning(id#129, 200), ENSURE_REQUIREMENTS, [id#505]
     :    +- LocalTableScan [id#129, left#130]
     +- Sort [id#140 ASC NULLS FIRST], false, 0
        +- Exchange hashpartitioning(id#140, 200), ENSURE_REQUIREMENTS, [id#506]
          +- LocalTableScan [id#140, right#141]

scala> case class City(id: Long, name: String)
defined class City

scala> val family = Seq(
  | Person(0, "Tenz", 0),
  | Person(1, "Ben", 1),
  | Person(2, "Gent", 2),
  | Person(3, "Gaara", 3)).toDS
<console>:23: error: not found: value Person
  Person(0, "Tenz", 0),
               ^
<console>:24: error: not found: value Person
```

3.LEFT ANTI-JOIN

A left anti join returns that all rows from the first table which do not have a match in the second table.

Commands

```
left.join(right, Seq("id"), "leftanti").show  
left.join(right, Seq("id"), "leftanti").explain
```

```

(l) Microsoft Teams | Subscription Details | Nuvepro | i-0d7a852b960160600 | New tab | + | 
https://wa.inuvepro.com/quacamole/#/client/aS0wZDdhODUyjk2MDE2MDYwMABjAG51dmVsaW5r... | 
Applications (1) Chaarviika A, +2 | Mi... | hadoop@ip-172-31-22-7... | hadoop - File Manager | 
File Edit View Search Terminal Help | hadoop@ip-172-31-22-77: ~/spark-3.2.0-bin-hadoop3.2/bin | 
Mon 12 Dec, 19:26 labuser | 
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala> left.join(right, Seq("id"), "leftanti").show
+---+---+
| id|left|
+---+---+
| 1| one|
+---+---+
scala> left.join(right, Seq("id"), "leftanti").explain
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- SortMergeJoin [id#129], [id#140], LeftAnti
  :- Sort [id#129 ASC NULLS FIRST], false, 0
    :+ Exchange hashpartitioning(id#129, 200), ENSURE_REQUIREMENTS, [id#684]
    :  +- LocalTableScan [id#129, left#130]
  +- Sort [id#140 ASC NULLS FIRST], false, 0
    :+ Exchange hashpartitioning(id#140, 200), ENSURE_REQUIREMENTS, [id#685]
    :  +- LocalTableScan [id#140]
|| teams.microsoft.com is sharing your screen. Stop sharing Hide

```

4. BROADCAST JOIN

Spark SQL uses broadcast join (aka broadcast hash join) instead of hash join to optimize join queries when the size of one side data is below spark.sql.autoBroadcastJoinThreshold.

Broadcast join can be very efficient for joins between a large table (fact) with relatively small tables (dimensions) that could then be used to perform a star- schema join. It can avoid sending all data of the large table over the network.

You can use broadcast function or SQL's broadcast hints to mark a dataset to be broadcast when used in a join query.

broadcast join is also called a replicated join (in the distributed system community) or a map-side join (in the Hadoop community).

JoinSelection execution planning strategy uses spark.sql.autoBroadcastJoinThreshold property (default: 10M) to control the size of a dataset before broadcasting it to all worker nodes when performing a join.

Commands

1. val threshold = spark.conf.get("spark.sql.autoBroadcastJoinThreshold").toInt
2. threshold / 1024 / 1024
3. val q = spark.range(100).as("a").join(spark.range(100).as("b")).where(\$"a.id" === \$"b.id")

```
4.println(q.queryExecution.logical.numberedTreeString)
5.q.explain
6.spark.conf.set("spark.sql.autoBroadcastJoinThreshold", -1)
7.spark.conf.get("spark.sql.autoBroadcastJoinThreshold")
```

```
1.q.explain
2.val qBroadcast =
spark.range(100).as("a").join(broadcast(spark.range(100)).as("b")).where($"a.id"
===$"b.id")
3.qBroadcast.explain
4.val qBroadcastLeft = """
SELECT /*+ BROADCAST (lf) */ *
FROM range(100) lf, range(1000) rt WHERE lf.id = rt.id
"""
scala> sql(qBroadcastLeft).explain
```

```
1. val qBroadcastRight = """
SELECT /*+ MAPJOIN (rt) */ *
FROM range(100) lf, range(1000) rt
WHERE lf.id = rt.id
"""
scala> sql(qBroadcastRight).explain
```

```

Microsoft Teams | Subscription Details | Nuvepro | i-0d7a852b960160600 | New tab | + | - | x
https://wai.nuvepro.com/guacamole/#/client/a50wZDdhODUyjk2MDE2MDYwMABjAG5... | Applications: Chaarvika A, +2 | Micro... | hadoop@ip-172-31-22-7... | hadoop - File Manager | hadoop@ip-172-31-22-77: ~/spark-3.2.0-bin-hadoop3.2/bin | Mon 12 Dec, 19:29 labuser
File Edit View Search Terminal Help

scala>
scala> val threshold = spark.conf.get("spark.sql.autoBroadcastJoinThreshold").toInt
threshold: Int = 2
scala> threshold / 1024 / 1024
res38: Int = 0
scala> val q = spark.range(100).as("a").join(spark.range(100).as("b")).where($"a.id" ===
| $"b.id")
q: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [id: bigint, id: bigint]
scala> println(q.queryExecution.logical.numberedTreeString)
00 'Filter ('a.id = 'b.id)
01 +- Join Inner
02   :: SubqueryAlias a
03     : +- Range (0, 100, step=1, splits=Some(2))
04     +- SubqueryAlias b
05       +- Range (0, 100, step=1, splits=Some(2))

scala> q.explain
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- SortMergeJoin [id#232L, [id#236L], Inner
  :- Sort [id#232L ASC NULLS FIRST], false, 0
  :  +- Exchange hashpartitioning(id#232L, 200), ENSURE_REQUIREMENTS, [id#702]
  :    +- Range (0, 100, step=1, splits=2)
  +- Sort [id#236L ASC NULLS FIRST], false, 0
    +- Exchange hashpartitioning(id#236L, 200), ENSURE_REQUIREMENTS, [id#703]
      +- Range (0, 100, step=1, splits=2)

scala> spark.conf.set("spark.sql.autoBroadcastJoinThreshold", -1)
scala> spark.conf.get("spark.sql.autoBroadcastJoinThreshold")
res42: String = -1
scala> q.explain
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- SortMergeJoin [id#232L, [id#236L], Inner
  :- Sort [id#232L ASC NULLS FIRST], false, 0
  :  +- Exchange hashpartitioning(id#232L, 200), ENSURE_REQUIREMENTS, [id#702]
  :    +- Range (0, 100, step=1, splits=2)
  +- Sort [id#236L ASC NULLS FIRST], false, 0
    +- Exchange hashpartitioning(id#236L, 200), ENSURE_REQUIREMENTS, [id#703]
      +- Range (0, 100, step=1, splits=2)

scala> spark.conf.set("spark.sql.autoBroadcastJoinThreshold", -1)
scala> spark.conf.get("spark.sql.autoBroadcastJoinThreshold")
res42: String = -1
scala> q.explain
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- SortMergeJoin [id#232L, [id#236L], Inner
  :- Sort [id#232L ASC NULLS FIRST], false, 0
  :  +- Exchange hashpartitioning(id#232L, 200), ENSURE_REQUIREMENTS, [id#702]
  :    +- Range (0, 100, step=1, splits=2)
  +- Sort [id#236L ASC NULLS FIRST], false, 0
    +- Exchange hashpartitioning(id#236L, 200), ENSURE_REQUIREMENTS, [id#703]
      +- Range (0, 100, step=1, splits=2)

scala> val qBroadcast =
|   spark.range(100).as("a").join(broadcast(spark.range(100)).as("b")).where($"a.id" ===
|   $"b.id")
qBroadcast: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [id: bigint, id: bigint]
scala> qBroadcast.explain
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- BroadcastHashJoin [id#242L, [id#246L], Inner, BuildRight, false
  :- Range (0, 100, step=1, splits=2)

```

```
Microsoft Teams | Subscription Details | Nuvepro | i-0d7a852b960160600 | New tab | + | - | x | https://wa.inuvepro.com/guacamole/#/client/aS0wZDdhODUyjk2MDE2MDYwMABjAG5... | Applications: Chaarvika A, +2 | Microsoft Edge | hadoop@ip-172-31-22-7... | hadoop - File Manager | hadoop@ip-172-31-22-77: ~/spark-3.2.0-bin-hadoop3.2/bin | Mon 12 Dec, 19:30 labuser | ... |
```

```
File Edit View Search Terminal Help
| WHERE lf.id = rt.id
|   """
qBroadcastLeft: String =
"
SELECT /*+ BROADCAST (lf) */ *
FROM range(100) lf, range(1000) rt
WHERE lf.id = rt.id
"

scala> sql(qBroadcastLeft).explain
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- BroadcastHashJoin [id#254L], [id#255L], Inner, BuildLeft, false
  :- BroadcastExchange HashedRelationBroadcastMode(List(input[0, bigint, false]),false), [id#733]
    +- Range (0, 100, step=1, splits=2)
  +- Range (0, 1000, step=1, splits=2)

scala> val qBroadcastRight = """
| SELECT /*+ MAPJOIN (rt) */ *
| FROM range(100) lf, range(1000) rt
| WHERE lf.id = rt.id
|   """
qBroadcastRight: String =
"
SELECT /*+ MAPJOIN (rt) */ *
FROM range(100) lf, range(1000) rt
WHERE lf.id = rt.id
"

scala> sql(qBroadcastRight).explain
== Physical Plan ==
AdaptiveSparkPlan isFinalPlan=false
+- BroadcastHashJoin [id#258L], [id#259L], Inner, BuildRight, false
  :- Range (0, 100, step=1, splits=2)
  +- BroadcastExchange HashedRelationBroadcastMode(List(input[0, bigint, false]),false), [id#747]
    +- Range (0, 1000, step=1, splits=2)

scala>
```

LAB EXCERSICE: 9

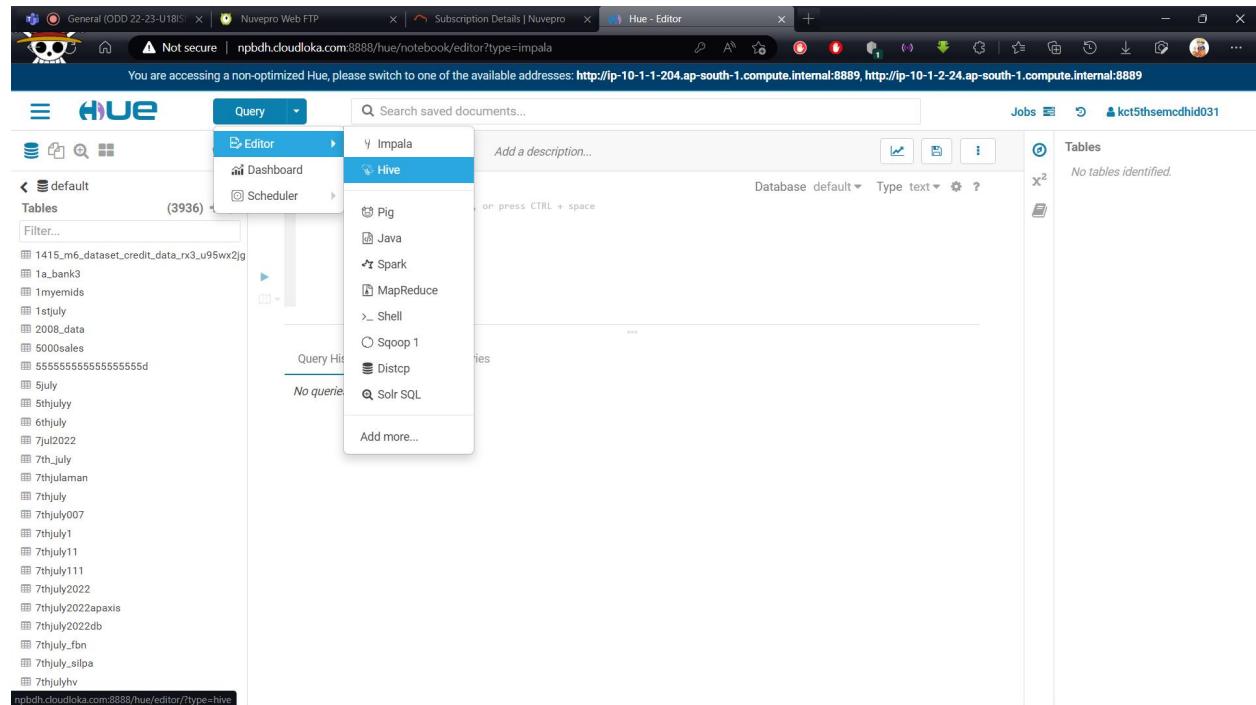
HIVE

AIM:

Install and Run Hive then use Hive to create, alter, and drop databases, tables, views,functions, and indexes

PROCEDURE:

STEP1:Opening HIVE from HUE portal cloudera



You are accessing a non-optimized Hue, please switch to one of the available addresses: <http://ip-10-1-1-204.ap-south-1.compute.internal:8889>, <http://ip-10-1-2-24.ap-south-1.compute.internal:8889>

HUE

Query Add a name... Add a description... Database default Type text

default Tables

Example: SELECT * FROM tablename, or press CTRL + space

Query History Saved Queries You don't have any saved query.

npbdh.cloudloka.com:888/hue/notebook/editor?type=hive

STEP2: Create database

```
create database 20bis031bigdatahive; show databases;
```

```
use 20bis031bigdatahive;
```

You are accessing a non-optimized Hue, please switch to one of the available addresses: <http://ip-10-1-1-204.ap-south-1.compute.internal:8889>, <http://ip-10-1-2-24.ap-south-1.compute.internal:8889>

HUE

Query Add a name... Add a description... Database default Type text

default Tables (3938) Filter...

1 show databases;
2 create database 20bis031bigdatahive;
3 show databases;
4 use 20bis031bigdatahive;
5 |

use 20bis031bigdatahive
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20221213072029_0b5001fa-cc2f-478c-932b-d2281fc252c2); Time taken: 0.004 sec
INFO : OK

Success.

Query History Saved Queries

2 minutes ago ✓ use 20bis031bigdatahive
12 minutes ago ✓ show databases
12 minutes ago show databases
13 minutes ago ✓ show databases; create database 20bis031bigdatahive;
14 minutes ago ✓ show databases;

npbdh.cloudloka.com:888/hue/notebook/editor?type=hive

The screenshot shows the Hue interface with the title bar "Hue - Editor". The left sidebar shows the "Tables" section for the "default" database, listing 3938 tables. The main area displays a list of 30 table names, including "20bis031bigdatahive" which is currently selected. The right sidebar shows the "Jobs" and "Tables" sections, both indicating "Statement 3/3" and "No tables identified".

STEP3:Create table

The screenshot shows the Hue interface with the title bar "Hive - Create Table". The left sidebar shows the "Tables" section for the "default" database. The main area has a query editor with the command "create table studentskct(rollno varchar(10) , subject varchar(10) , mark int , grade varchar(10))". Below the editor, the logs show the execution of the command: "INFO : Starting task [Stage-0:DDL] in serial mode", "INFO : Completed executing command(queryId=hive_20221219073116_5b34a723-c3d5-4379-9585-44a03966758b); Time taken: 0.059 seconds", and "INFO : OK". The right sidebar shows the "Tables" section, which also indicates "No tables identified".

STEP4:Insert values in the table

```
insert into studentskct values('20bis100','cs',90,'A'),  
                            ('20bis101','cs',90,'A'),  
                            ('20bis102','i5',80,'B'),  
                            ('20bis103','cs',80,'B'),  
                            ('20bis104','is',60,'D'),  
                            ('20bis105','is',70,'C');
```

```
select * from studentskct;
```

The screenshot shows the Hue interface with the following details:

- Left Sidebar:** Shows a list of databases and tables, with "default" selected.
- Hive Editor Tab:** Contains the following code:

```
1 create table studentskct(rollno varchar(10) , subject varchar(10) , mark int , grade varchar(10));  
2 insert into studentskct values('20bis100','cs',90,'A'),  
   ('20bis101','cs',90,'A'),  
   ('20bis102','i5',80,'B'),  
   ('20bis103','cs',80,'B'),  
   ('20bis104','is',60,'D'),  
   ('20bis105','is',70,'C');  
7  
8 select * from studentskct;  
9
```
- Output Log:** Displays the execution log:

```
INFO : Executing command(queryId=hive_20221213073735_0d921410-2039-48c9-aa46-3403beaa07cd); Time taken: 0.8 seconds  
INFO : OK
```
- Results Table:** Shows the data inserted into the table:

rollno	subject	mark	grade
20bis100	cs	90	A
20bis101	cs	90	A
20bis102	i5	80	B
20bis103	cs	80	B
20bis104	is	60	D
20bis105	is	70	C

STEP5:Filtering records with condition

```
select * from studentskct where mark<80;
```

You are accessing a non-optimized Hue, please switch to one of the available addresses: <http://ip-10-1-1-204.ap-south-1.compute.internal:8889>, <http://ip-10-1-2-24.ap-south-1.compute.internal:8889>

HUE Query Search saved documents...

Hive Add a name... Add a description...

Tables (3944) Filter...

```
1 select * from studentskct where mark<80;
2
```

24.1s Database default Type text

INFO : Total MapReduce CPU Time Spent: 3 seconds 910 msec job_1663041244711_21664
INFO : Completed executing command(queryId=hive_20221213074010_018eaf9b-067c-4118-be3d-b7cf1cb00000), Time taken: 23.001 seconds
INFO : OK

Query History Saved Queries Results (2)

	studentskct.rollno	studentskct.subject	studentskct.mark	studentskct.grade
1	20bis104	is	60	D
2	20bis105	is	70	C

Tables Filter... default.studentskct

STEP6:Drop table

drop table studentskct;

You are accessing a non-optimized Hue, please switch to one of the available addresses: <http://ip-10-1-1-204.ap-south-1.compute.internal:8889>, <http://ip-10-1-2-24.ap-south-1.compute.internal:8889>

HUE Query Search saved documents...

Hive Add a name... Add a description...

Tables (3945) Filter...

```
1 drop table studentskct;
```

1.25s Database default Type text

INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20221213074642_346f4a89-aabd-412e-badc-f31e6ad699e4); Time taken: 0.13 seconds
INFO : OK

Success.

Query History Saved Queries

a few seconds ago	drop table studentskct
7 minutes ago	select * from studentskct where mark<80
10 minutes ago	select * from studentskct
11 minutes ago	insert into studentskct values('20bis100','cs',90,'A'), ('20bis101','cs',90,'A'), ('20bis102','is',80,'B'), ('20bis103','cs',80,'B'), ('20bis104','is',60,'D'), ('20bis105','is',70,'C')
16 minutes ago	create table studentskct(rollno varchar(10) , subject varchar(10) , mark int , grade varchar(10))
16 minutes ago	create table studentkct(rollno varchar(10) , subject varchar(10) , mark int , grade varchar(10))

Tables Filter... default.studentskct

LAB EXERCISE-10

SPARK

AIM:

To Verify, Sparse and perform advance join of data using spark

THEORY:

Join operations in Apache Spark is often a biggest source of performance problems and even full-blown exceptions in Spark. After this talk, you will understand the two most basic methods Spark employs for joining dataframes – to the level of detail of how Spark distributes the data within the cluster. You'll also find out how to work out common errors and even handle the trickiest corner cases we've encountered! After this talk, you should be able to write performance joins in Spark SQL that scale and are zippy fast!

This session will cover different ways of joining tables in Apache Spark.

ShuffleHashJoin

–A ShuffleHashJoin is the most basic way to join tables in Spark – we'll diagram how Spark shuffles the dataset to make this happen.

BroadcastHashJoin

–A BroadcastHashJoin is also a very common way for Spark to join two tables under the special condition that one of your tables is small.

Dealing with Key Skew in a ShuffleHashJoin

–Key Skew is a common source of slowness for a Shuffle Hash Join – we'll describe what this is and how you might work around this.

CartesianJoin

–Cartesian Joins is a hard problem – we'll describe why it's difficult as well as what you need to do to make that work and what to look out for.

One to Many Joins

–When a single row in one table can match to many rows in your other table, the total number of output rows in your joined table can be really high. We'll let you know how to deal with this.

Theta Joins

–If you aren't joining two tables strictly by key, but instead checking on a condition for your tables, you may need to provide some hints to Spark SQL to get this to run well.

1. Broadcast Hash Join

COMMANDS

```
scala> spark.conf.get("spark.sql.autoBroadcastJoinThreshold") res1: String = 10485760
scala> val data1 = Seq(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
data1: Seq[Int] = List(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
scala> val df1 = data1.toDF("id1")
df1: org.apache.spark.sql.DataFrame = [id1: int]
scala> val data2 = Seq(30, 20, 40, 50)
data2: Seq[Int] = List(30, 20, 40, 50)
scala> val df2 = data2.toDF("id2")
df2: org.apache.spark.sql.DataFrame = [id2: int]

val dfJoined = df1.join(df2, $"id1" === $"id2")
```

```
scala> dfJoined.queryExecution.executedPlan
```

```
scala> dfJoined.queryExecution.executedPlan  
scala> dfJoined.show
```



Microsoft Teams | Subscription Details | Nuvelo | i-0d7a852b960160600 | New tab | +

Applications : Chaarvik A. +2 | Micro... | hadoop@ip-172-31-22-77: ~ | hadoop - File Manager | hadoop@ip-172-31-22-77: ~/spark-3.2.0-bin-hadoop3.2/bin

File Edit View Search Terminal Help

Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

22/12/12 18:33:08 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

Spark context Web UI available at http://ip-172-31-22-77.ap-south-1.compute.internal:4040

Spark context available as 'sc' (master = local[*], app id = local-1670869990240).

Spark session available as 'spark'.

Welcome to

version 3.2.0

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 11.0.17)

Type in expressions to have them evaluated.

Type :help for more information.

```
scala> spark.conf.get("spark.sql.autoBroadcastJoinThreshold")
res0: String = 10485760b

scala> val data1 = Seq(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
data1: Seq[Int] = List(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)

scala> val df1 = data1.toDF("id1")
df1: org.apache.spark.sql.DataFrame = [id1: int]

scala> val data2 = Seq(30, 20, 40, 50)
data2: Seq[Int] = List(30, 20, 40, 50)

scala> val df2 = data2.toDF("id2")
df2: org.apache.spark.sql.DataFrame = [id2: int]

scala> val dfJoined = df1.join(df2, $"id1" === $"id2")
dfJoined: org.apache.spark.sql.DataFrame = [id1: int, id2: int]

scala> dfJoined.explain()
res1: org.apache.spark.sql.execution.SparkPlan =
  AdaptiveSparkPlan isFinalPlan=false
  +- BroadcastHashJoin [id1#4, [id2#10], Inner, BuildRight, false
    +- LocalTableScan [id1#4]
    +- BroadcastExchange HashedRelationBroadcastMode(List[cast(inout#0, int, false) as bigint]), [id#12]
```

Microsoft Teams | Subscription Details | Nuvepro | New tab | +

https://wai.nuvepro.com/guacamole/#/client/a50wZDdhODUyjk2MDE2MDYwMABjAG51dmVsaW5r... | Mon 12 Dec, 18:50 labuser

File Edit View Search Terminal Help

```
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala>
scala> dfJoined.queryExecution.executedPlan
res4: org.apache.spark.sql.execution.SparkPlan =
AdaptiveSparkPlan isFinalPlan=false
+- BroadcastHashJoin [id1#4], [id2#10], Inner, BuildRight, false
  :- LocalTableScan [id1#4]
  +- BroadcastExchange HashedRelationBroadcastMode(List(cast(input[0, int, false] as bigint)),false), [id=#12]
    +- LocalTableScan [id2#10]

scala> dfJoined.show
+---+---+
|id1|id2|
+---+---+
| 20| 20|
| 20| 20|
| 30| 30|
| 40| 40|
| 40| 40|
| 20| 20|
| 20| 20|
| 20| 20|
| 50| 50|
+---+---+
```

2. Shuffle Hash Join

COMMANDS

```
scala> spark.conf.set("spark.sql.autoBroadcastJoinThreshold", 2) scala>
spark.conf.set("spark.sql.join.preferSortMergeJoin", "false")
```

```
scala> spark.conf.get("spark.sql.join.preferSortMergeJoin") res2: String = false
```

```
scala> spark.conf.get("spark.sql.autoBroadcastJoinThreshold") res3: String = 2
```

```
scala> val data1 = Seq(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
```

```
data1: Seq[Int] = List(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
```

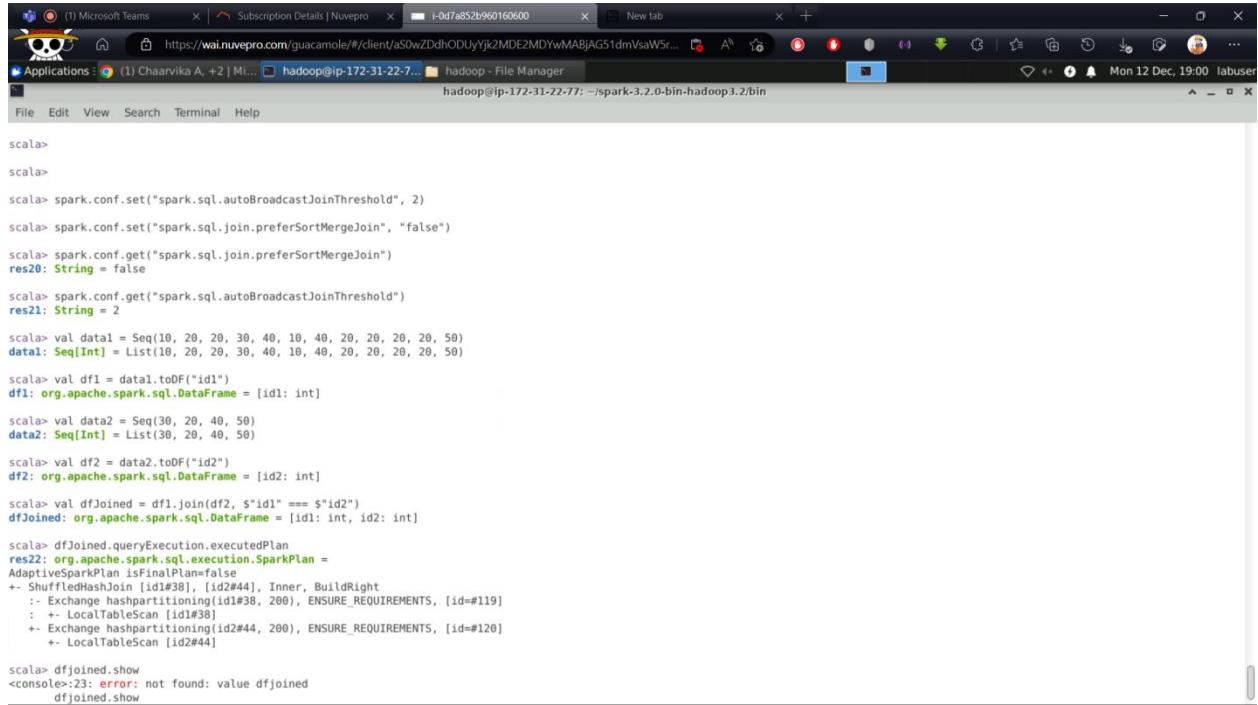
```
scala> val df1 = data1.toDF("id1")
df1: org.apache.spark.sql.DataFrame = [id1: int]
```

```
scala> val data2 = Seq(30, 20, 40, 50)
data2: Seq[Int] = List(30, 20, 40, 50)
```

```
scala> val df2 = data2.toDF("id2")
df2: org.apache.spark.sql.DataFrame = [id2: int]
```

```
scala> val dfJoined = df1.join(df2, $"id1" === $"id2") dfJoined: org.apache.spark.sql.DataFrame
= [id1: int, id2: int]
```

```
scala> dfJoined.queryExecution.executedPlan
scala> dfJoined.show
```



The screenshot shows a terminal window with the following content:

```
scala>
scala>
scala> spark.conf.set("spark.sql.autoBroadcastJoinThreshold", 2)
scala> spark.conf.set("spark.sql.join.preferSortMergeJoin", "false")
scala> spark.conf.get("spark.sql.join.preferSortMergeJoin")
res20: String = false
scala> spark.conf.get("spark.sql.autoBroadcastJoinThreshold")
res21: String = 2
scala> val data1 = Seq(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
data1: Seq[Int] = List(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
scala> val df1 = data1.toDF("id1")
df1: org.apache.spark.sql.DataFrame = [id1: int]
scala> val data2 = Seq(30, 20, 40, 50)
data2: Seq[Int] = List(30, 20, 40, 50)
scala> val df2 = data2.toDF("id2")
df2: org.apache.spark.sql.DataFrame = [id2: int]
scala> val dfJoined = df1.join(df2, $"id1" === $"id2")
dfJoined: org.apache.spark.sql.DataFrame = [id1: int, id2: int]
scala> dfJoined.queryExecution.executedPlan
res22: org.apache.spark.sql.execution.SparkPlan =
AdaptiveSparkPlan isFinalPlan=false
+- ShuffledHashJoin [id1#38], [id2#44], Inner, BuildRight
  :- Exchange hashpartitioning(id1#38, 200), ENSURE_REQUIREMENTS, [id=#119]
  : + LocalTableScan [id1#38]
  +- Exchange hashpartitioning(id2#44, 200), ENSURE_REQUIREMENTS, [id=#120]
    + LocalTableScan [id2#44]
scala> dfJoined.show
<console>:23: error: not found: value dfjoined
      dfjoined.show
```

```

(t) Microsoft Teams | Subscription Details | Nuvepro | New tab | + | Mon 12 Dec, 19:00 labuser
https://wa.nuvepro.com/guacamole/#/client/aS0wZDdhODUyjk2MDE2MDYwMABjAG51dmVsaW5r... | Applications: (1) Chaarvika A, +2 | Mi... | hadoop@ip-172-31-22-7... | hadoop - File Manager | hadoop@ip-172-31-22-77: ~/spark-3.2.0-bin-hadoop3.2/bin | + | Mon 12 Dec, 19:00 labuser

File Edit View Search Terminal Help

scala> val data2 = Seq(30, 20, 40, 50)
data2: Seq[Int] = List(30, 20, 40, 50)

scala> val df2 = data2.toDF("id2")
df2: org.apache.spark.sql.DataFrame = [id2: int]

scala> val dfJoined = df1.join(df2, $"id1" === $"id2")
dfJoined: org.apache.spark.sql.DataFrame = [id1: int, id2: int]

scala> dfJoined.explain()
res22: org.apache.spark.sql.execution.SparkPlan =
AdaptiveSparkPlan$.isFinalPlan=false
+- ShuffledHashJoin [id1#38], [id2#44], Inner, BuildRight
  : Exchange hashpartitioning(id1#38, 200), ENSURE_REQUIREMENTS, [id#119]
  : + LocalTableScan [id1#38]
  + Exchange hashpartitioning(id2#44, 200), ENSURE_REQUIREMENTS, [id#120]
  + LocalTableScan [id2#44]

scala> dfJoined.show
<console>:23: error: not found: value dfjoined
           dfjoined.show
               ^
scala> dfJoined.show
+---+---+
|id1|id2|
+---+---+
| 20| 20|
| 20| 20|
| 40| 40|
| 30| 30|
| 40| 40|
| 20| 20|
| 20| 20|
| 20| 20|
| 50| 50|
+---+---+
scala>

```

3. CARTESIAN PRODUCT JOIN

COMMANDS

```

scala> spark.conf.get("spark.sql.join.preferSortMergeJoin") res1: String = true
scala> spark.conf.get("spark.sql.autoBroadcastJoinThreshold") res2: String = -1
scala> val data1 = Seq(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
data1: Seq[Int] = List(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
scala> val df1 = data1.toDF("id1")
df1: org.apache.spark.sql.DataFrame = [id1: int] 5.scala> val data2 = Seq(30, 20, 40, 50)
data2: Seq[Int] = List(30, 20, 40, 50) 6.scala> val df2 = data2.toDF("id2")
df2: org.apache.spark.sql.DataFrame = [id2: int]

```

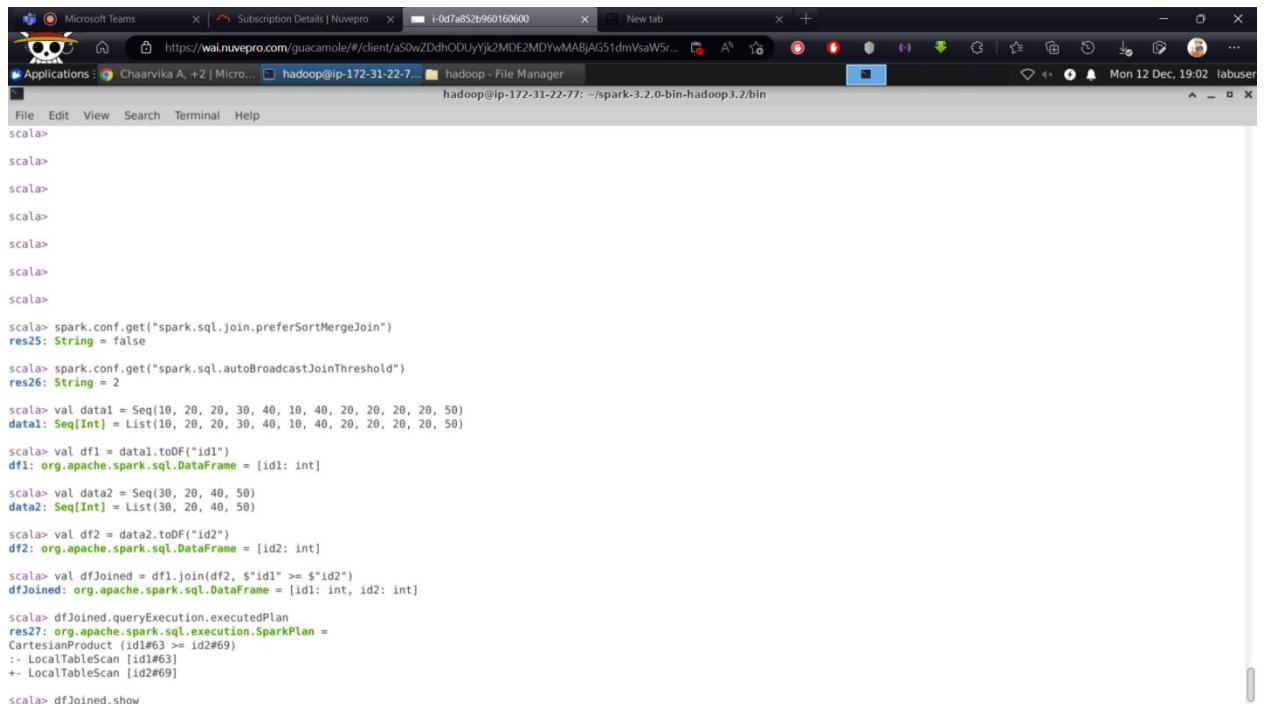
```

scala> val dfJoined = df1.join(df2, $"id1" >= $"id2")
dfJoined: org.apache.spark.sql.DataFrame = [id1: int, id2: int]

```

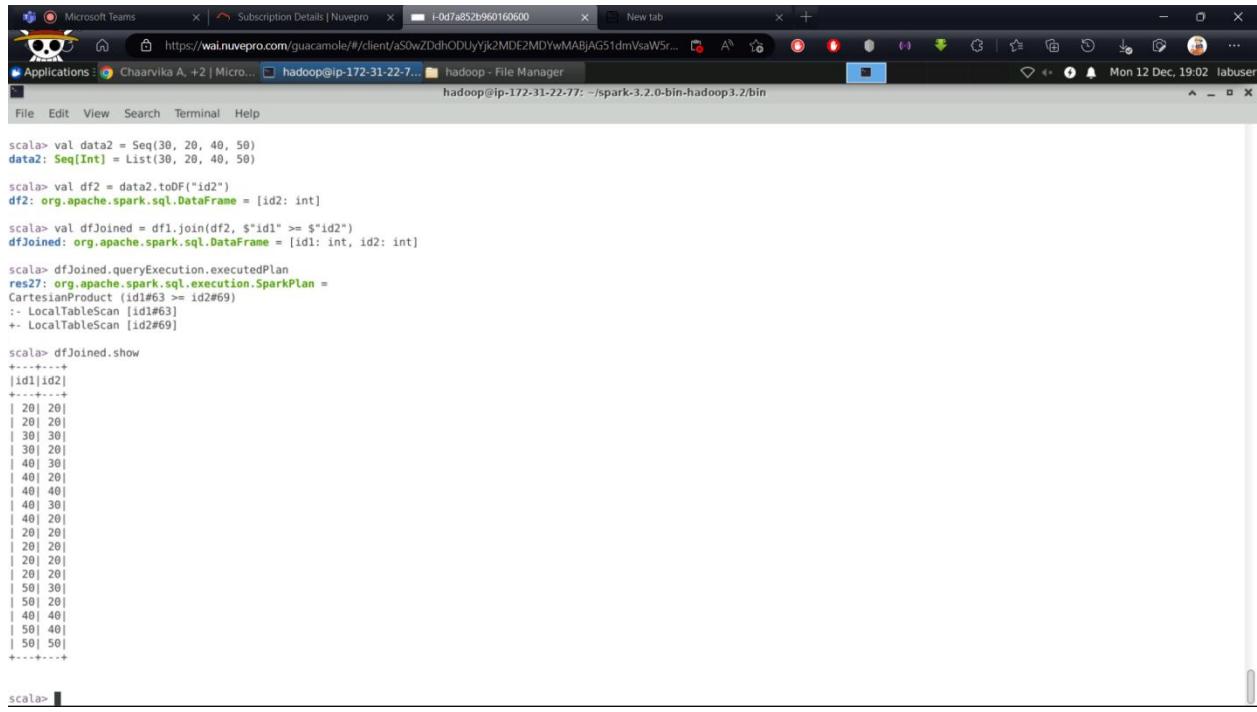
```
scala> dfJoined.queryExecution.executedPlan
```

```
scala> dfJoined.show
```



The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is a terminal window titled 'hadoop@ip-172-31-22-7: ~/spark-3.2.0-bin-hadoop3.2/bin'. The terminal is running a Scala REPL session. The user has entered several commands to demonstrate DataFrame joins:

```
scala>
scala>
scala>
scala>
scala>
scala> spark.conf.get("spark.sql.join.preferSortMergeJoin")
res25: String = false
scala> spark.conf.get("spark.sql.autoBroadcastJoinThreshold")
res26: String = 2
scala> val data1 = Seq(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
data1: Seq[Int] = List(10, 20, 20, 30, 40, 10, 40, 20, 20, 20, 20, 50)
scala> val df1 = data1.toDF("id1")
df1: org.apache.spark.sql.DataFrame = [id1: int]
scala> val data2 = Seq(30, 20, 40, 50)
data2: Seq[Int] = List(30, 20, 40, 50)
scala> val df2 = data2.toDF("id2")
df2: org.apache.spark.sql.DataFrame = [id2: int]
scala> val dfJoined = df1.join(df2, $"id1" >= $"id2")
dfJoined: org.apache.spark.sql.DataFrame = [id1: int, id2: int]
scala> dfJoined.queryExecution.executedPlan
res27: org.apache.spark.sql.execution.SparkPlan =
CartesianProduct (id1#63 >= id2#69)
:- LocalTableScan [id1#63]
+- LocalTableScan [id2#69]
scala> dfJoined.show
```



A screenshot of a terminal window titled "hadoop - File Manager" running on a Hadoop cluster. The window shows Scala code being executed. The code defines a sequence of integers, converts it to a DataFrame, joins it with itself, and then prints the resulting DataFrame.

```
scala> val data2 = Seq(30, 20, 40, 50)
data2: Seq[Int] = List(30, 20, 40, 50)

scala> val df2 = data2.toDF("id2")
df2: org.apache.spark.sql.DataFrame = [id2: int]

scala> val dfJoined = df1.join(df2, $"id1" >= $"id2")
dfJoined: org.apache.spark.sql.DataFrame = [id1: int, id2: int]

scala> dfJoined.queryExecution.executedPlan
res27: org.apache.spark.sql.execution.SparkPlan =
CartesianProduct (id1#63 >= id2#69)
:- LocalTableScan [id1#63]
+- LocalTableScan [id2#69]

scala> dfJoined.show
+---+---+
|id1|id2|
+---+---+
| 20| 20|
| 20| 20|
| 30| 30|
| 30| 20|
| 40| 30|
| 40| 20|
| 40| 40|
| 40| 30|
| 40| 20|
| 20| 20|
| 20| 20|
| 20| 20|
| 50| 30|
| 50| 20|
| 40| 40|
| 50| 40|
| 50| 50|
+---+---+
scala>
```

