MAHENDRA COLLEGE OF ENGINEERING

Approved by AICTE and Affiliated by Anna University, Chennai NAAC Accredited–
Recognized U/S 2(F) &12(B) of UGC Act 1956

Salem-Chennai Highway NH79, Minnampalli, Salem-636106

DEPARTMENT OF INFORMATION TECHNOLOGY

CCS334 - BIG DATA ANALYTICS
RECORD NOTE BOOK
CLASS:III YEAR/VI SEMESTER

MAHENDRA

COLLEGE OF ENGINEERING

NAAC Accredited

Approved by AICTE and Affiliated by AnnaUniversity, Chennai Salem-Chennai Highway NH 79, Minnampalli, Salem-636106

Department of

_____

LABORATORY RECORD

Certified to be the Bonafide of work done by

Name:_____Register No:_____

Class: _____Branch:_____

Laboratory Name:_____

HEAD OF THE DEPARTMENT STAFF IN-CHARGE

DATE:

Submitted for the University Practical Examination on

_____

INTERNAL EXAMINER EXTERNAL EXAMINER

MAHENDRA COLLEGE OF ENGINEERING

SALEM-CAMPUS, ATTUR MAIN ROAD, MINNAMPALLI, SALEM -636 106.

INSTITUTION VISION AND MISSION

VISION

Mahendra College of Engineering is committed to be a leader in Higher Education achieving excellence through world class learning environment for Science and Technology with a blend of advanced research to create ethical and competent professionals.

MISSION

- To provide a conductive atmosphere to impart innovative knowledge and commendable skills through quality education by continuous improvement and customization of teaching.
- To nurture research attitude and bring about tangible developments with dynamic Industry - Institute Interaction.
- To create society oriented citizens with professional ethics.

MAHENDRA COLLEGE OF ENGINEERING

SALEM-CAMPUS, ATTUR MAIN ROAD, MINNAMPALLI, SALEM -636 106.

DEPARTMENT OF INFORMATION TECHNOLOGY

DEPARTMENT VISION AND MISSION

VISION

To become a department, producing graduates with good technical skills in emerging areas of Information Technology, through value based education and research.

MISSION

- To provide exposure to students to the emerging technologies in Hardware and Software.
- To inculcate students with sound application knowledge.
- To establish strong Industry- Institute Interaction.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

To ensure graduates

- Have proficiency in programming skills to design, develop and apply appropriate techniques, to solve complex engineering problems.
- Have knowledge to build, automate and manage business solutions using cutting edge technologies.
- Have excitement towards research in applied computer technologies.

PROGRAM OUTCOMES (POs)

1. Engineering knowledge:

Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis:

Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions:

Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems:

Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usages:

Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society:

Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability:

Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics:

Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work:

Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communications:

Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance:

Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and imultidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological chang

CCS334 - BIGDATA ANALYTICS LABORATORY COURSE OBJECTIVES:

- To understand big data.
- To learn and use NoSQL big data management.
- To learn map reduce analytics using Hadoop and related tools
- To work with map reduce applications
- To understand the usage of Hadoop related tools for Big Data Analytics

LIST OF EXPERIMENTS:

1. Downloading and installing Hadoop; Understanding different Hadoop modes. Startup scripts, Configuration files.
2. Hadoop Implementation of file management tasks, such as Adding files and directories, retrieving files and Deletingfiles
3. Implement of Matrix Multiplication with Hadoop Map Reduce
4. Run a basicWord CountMap Reduce programto understandMap ReduceParadigm.
5. Installation of Hive along with practice examples.
7. Installationof HBase, Installing thrift along with Practice examples
8. Practice importing and exporting data from various databases.

TOTAL: 30 PERIODS

SOFTWAREAND HARDWARE REQUIREMENTS:

SoftwareRequirements	Cassandra, Hadoop,Java, Pig,Hiveand HBase.
Hardware Requirements	DesktopComputer

COURSEOUTCOMES:

CO1: Describe big data and use cases from selected business domains.

CO2: Explain NoSQL big data management.

CO3: Install, configure, and run Hadoop and HDFS.

CO4: Perform map-reduce analytics using Hadoop.

CO5: Use Hadoop-related tools such as HBase, Cassandra, Pig, and Hive for big data analytics.

CO's-PO's&PSO's MAPPING

CO's	O's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	3	3	3	3	3	-	-	-2	2	2	3	1	1	3	3
	3	3	2	3	2	-	-	-3	2	2	3	3	2	3	2
	3	3	3	2	3	-	-	-4	2	2	1	2	2	3	3
	2	3	3	3	3	-	-	-	2	2	3	2	3	3	2
5	3	3	3	3	3	-	-	-	3	1	3	2	3	2	3
AVg.	2.8	3	2.8	2.8	2.8	-	-	-	2.2	1.8	2.6	2	2.2	.8	2.6

1 -low,2 - medium, 3-high, '-' -no correlation

CCS334-BIG DATA ANALYTICS

LABORATORYLISTOFEXPERIMENTS:

1. Downloading and installing Hadoop; Understanding different Hadoop modes.Startup scripts, Configuration files.
2. Hadoop Implementation of file management tasks, such as Adding files and directories, retrieving files and Deleting files
3. Implement of Matrix Multiplication with Hadoop Map Reduce
4. Run a basic Word Count Map Reduce program to understand Map Reduce

Paradigm. 5.Installationof Hive along with practice examples. 6.Installationof HBase, Installing thrift along with Practice examples

7. Practice importing and exporting data from various databases. Software Requirements: Cassandra, Hadoop, Java, Pig, Hive andHBase.

INDEX

S.NO	DATE	TITLE	PAGE NO	MARK	SIGN

INDEX

S.NO	DATE	TITLE	PAGE NO	MARK	SIGN

EXPT.NO.1 (a)	Downloading and installing Hadoop; Understanding different Hadoop modes. Start-up scripts, Configuration files.
DATE:	

AIM:

To Downloading and installing Hadoop; Understanding different Hadoop modes. Startup scripts, Configuration files.

PROCEDURE:

Prerequisites to Install Hadoop on Ubuntu

Hardware requirement-The machine must have 4GB RAM and minimum 60 GB hard disk for better performance.

Check java version- Itis recommended toinstall Oracle Java 8.

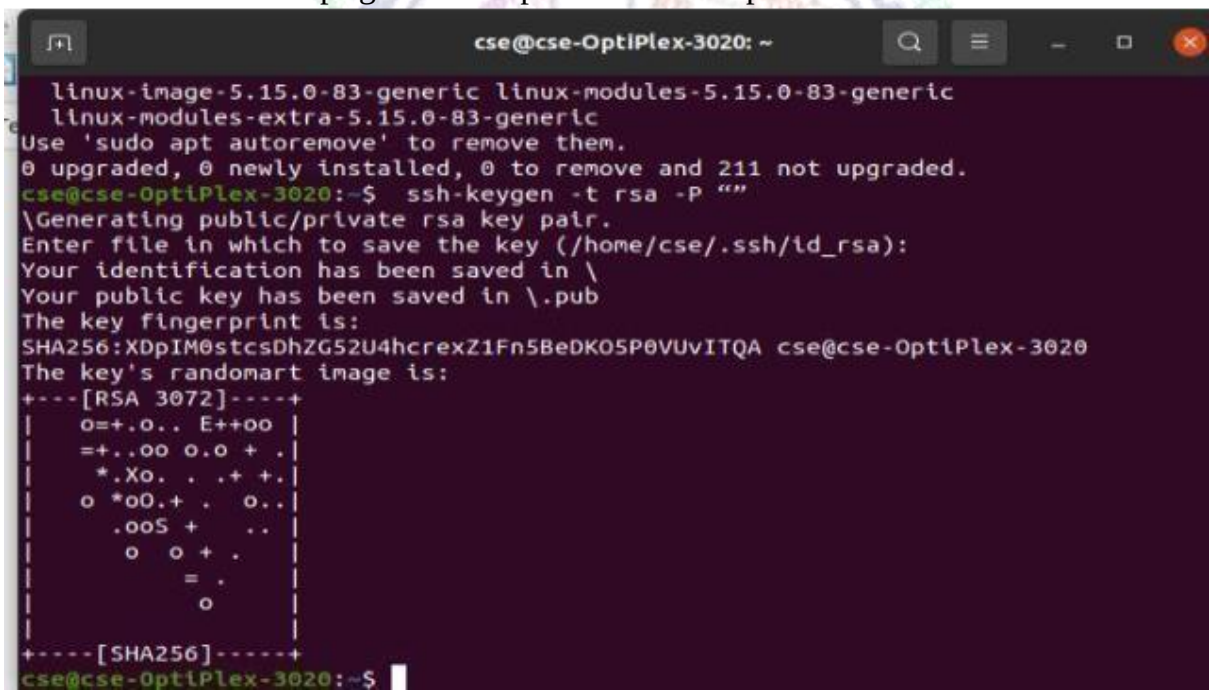
The user cancheck theversion of java with below command.

$ java –version

STEP1:Setuppasswordless ssh

a)Install Open SSH Server and Open SSH Client

We will now setup the password less ssh client with the following command. 1.$sudo apt-get install open ssh-server open ssh-client

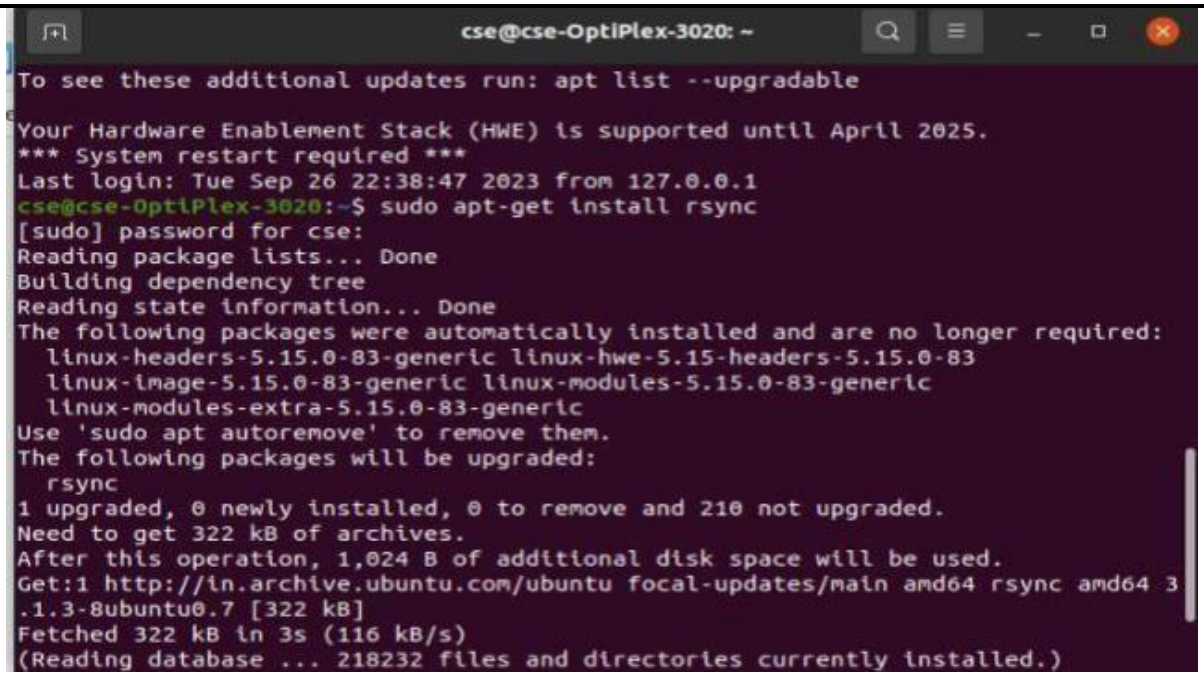


b) Generate Public & Private Key Pairs

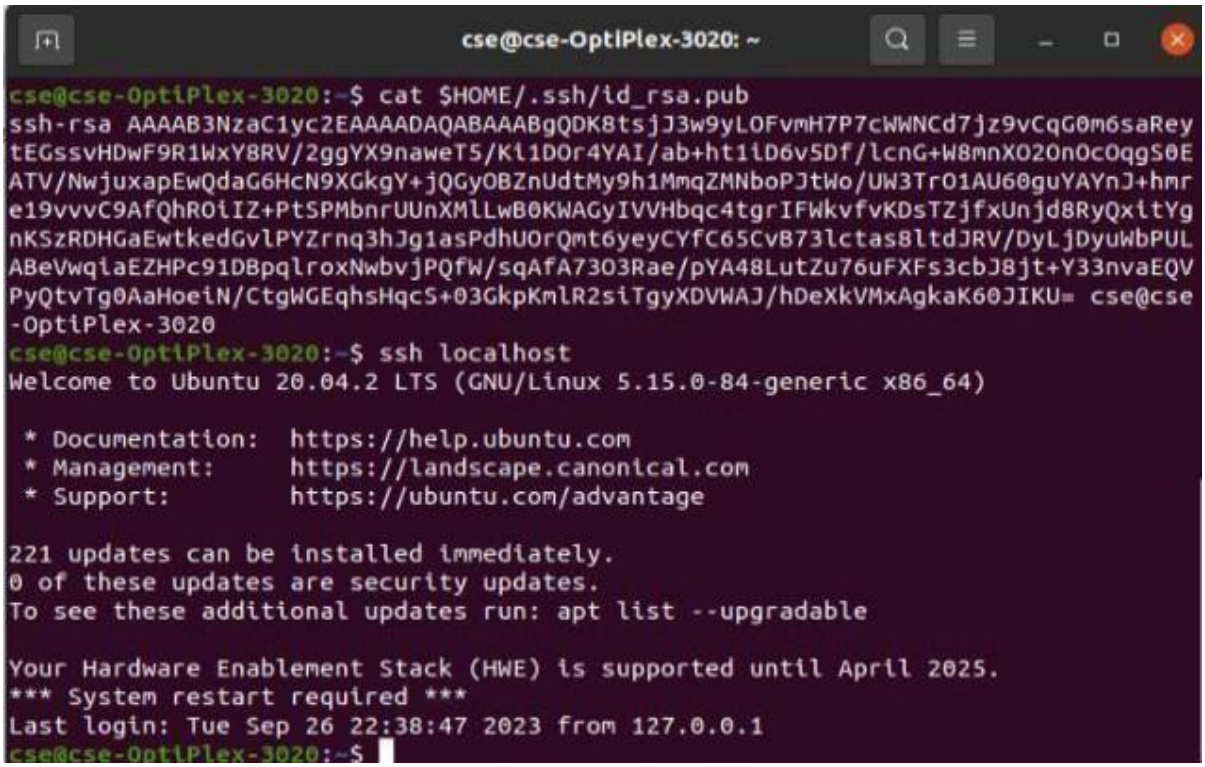
2.ssh-keygen -t rsa-P“”

c)Configurepassword-less SSH

3.cat $HOME/.ssh/id_rsa.pub>>$HOME/.ssh/authorized_keys



d)Now verify the working of password-
lessssh $ ssh localhost



e)Now install rsync with command $ sudo apt-get install rsync

STEP2:Configure and SetupHadoop Downloading Hadoop

$wgethttps://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6tar.gz $tar xzf hadoop-3.3.6.tar.gz

Result:

Downloaded and installed Hadoop and also understand different Hadoop modes are successfully implemented

EXP.NO:1(b) DATE:	Downloading and installing Hadoop; Startup scripts, Configuration

AIM:

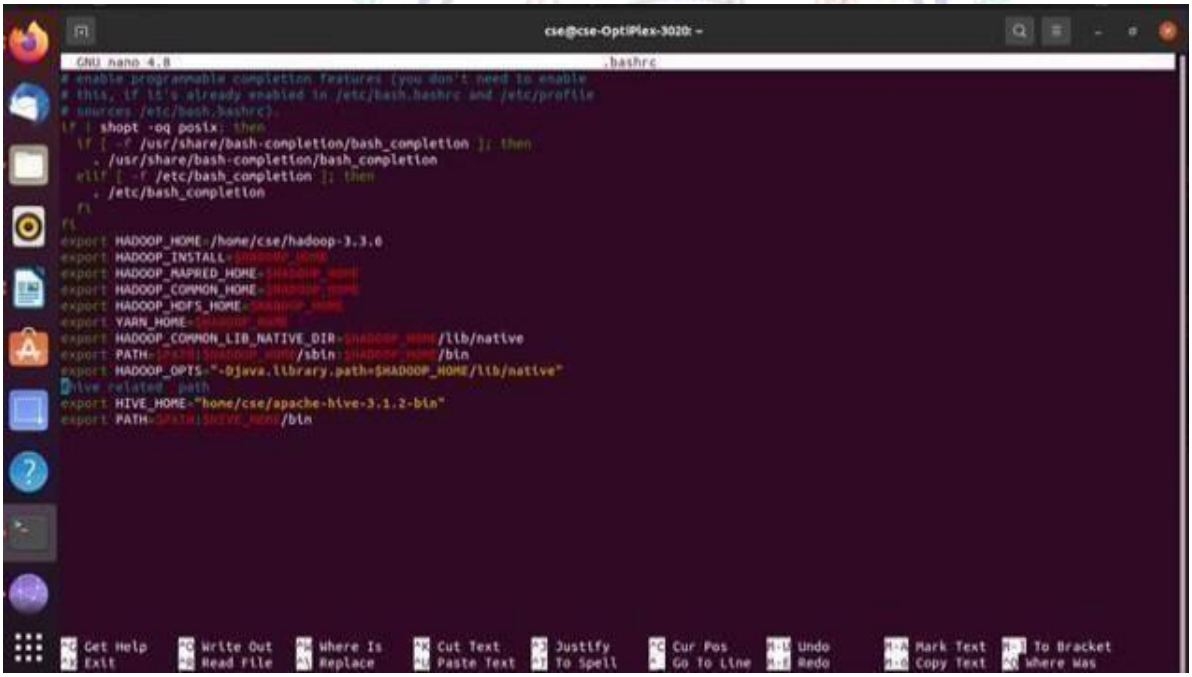
To Downloading and installing Hadoop;Understanding different Hadoop modes. Startup scripts, Configuration files.

STEP1:SetupConfiguration

a)Setting Up the environment variables

Edit .bashrc- Edit the bas hrc and therefore add hadoop in a path: $nano bash.bashrc

export HADOOP_HOME=/home/cse/hadoop-3.3.6
export HADOOP_INSTALL=$HADOOP_HOMEexport HADOOP_MAPRED_HOME=$HADOOP_HOMEexport HADOOP_COMMON_HOME=$HADOOP_HOMEexport HADOOP_HDFS_HOME=$HADOOP_HOME
exportYARN_HOME=$HADOOP_HOME
exportHADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
exportPATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin



Source .bashrcin current loginsession in terminal $source~/.bashrc

b)Hadoop configuration file

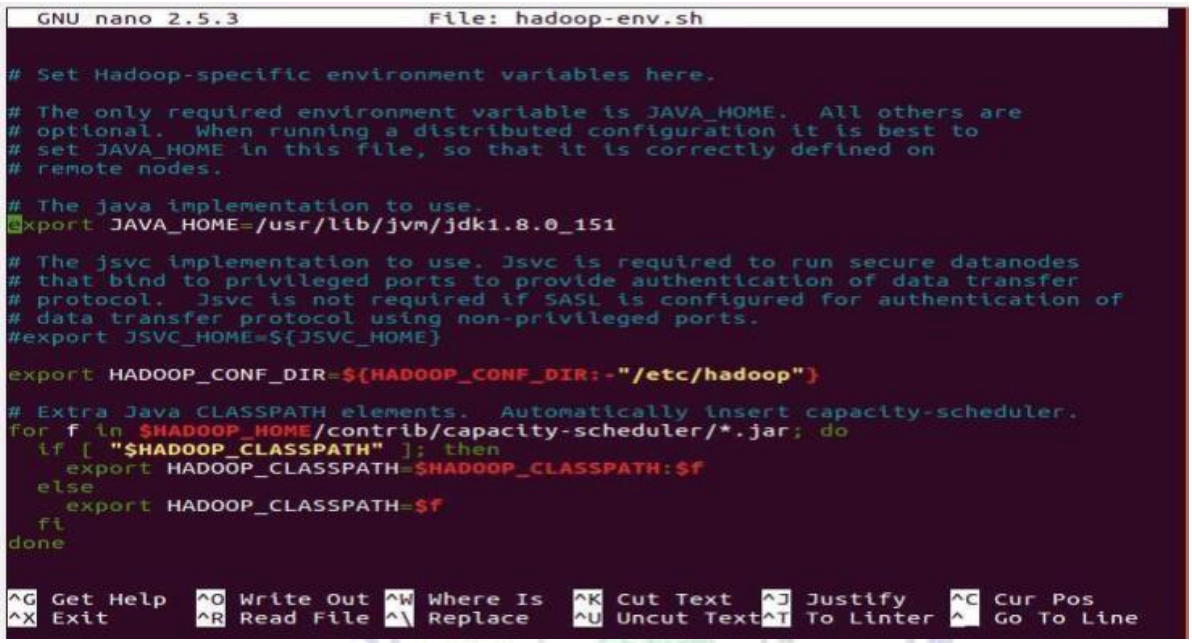
changes **Edit hadoop-env.sh**

Edit hadoop-env.sh file which is in etc/hadoop inside the Hadoop installation directory. $sudo nano $HADOOP_HOME/etc/hadoop/Hadoop-env.sh The user can set JAVA_HOME:

export JAVA_HOME=<root directory of Java-installation>

(eg:/usr/lib/jvm/jdk1.8.0_151/)

Editcore-site.xm

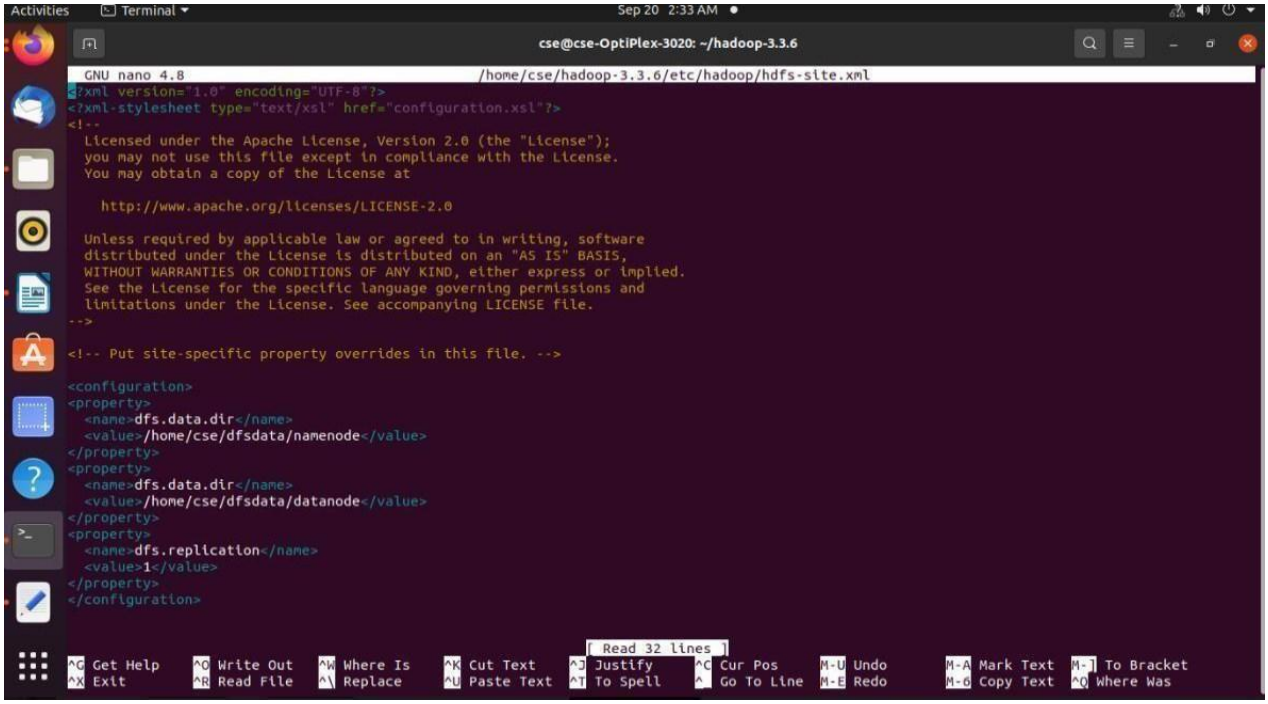


$sudonano $HADOOP_HOME/etc/hadoop/core-site.xml <configuration>
<property><name>fs.defaultFS</nam
e><value>hdfs://localhost:9000</valu
e></property>
<property><name>hadoop.tmp.di
r</name><value>/home/cse/hdat
a</value></property>
</configuration>

Edit hdfs-site.xml

$sudonano $HADOOP_HOME/etc/hadoop/hdfs-site.xml

#Addbelowlinesinthisfile(between"<configuration>"and"<"/configuration>") <property>
<name>dfs.data.dir</name><value>/home/cs
e/dfsdata/namenode</value>
</property>
<property>
<name>dfs.data.dir</name><value>/home/c
se/dfsdata/datanode</value>
</property>
<property>
<name>dfs.replication</name>
<value>1</value>
</propert

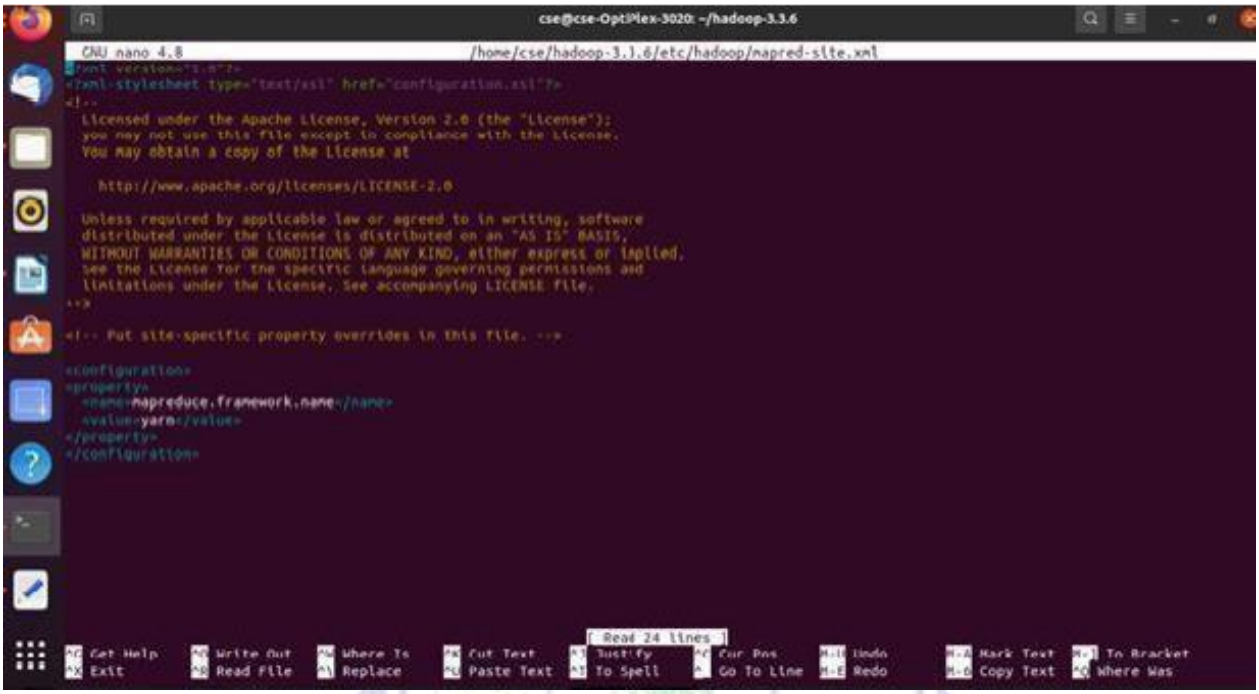


Edit mapred-site.xml

$sudonano $HADOOP_HOME/etc/hadoop/mapred-site.xml

#Add below linesin this file(between "<configuration>"and "<"/configuration>")

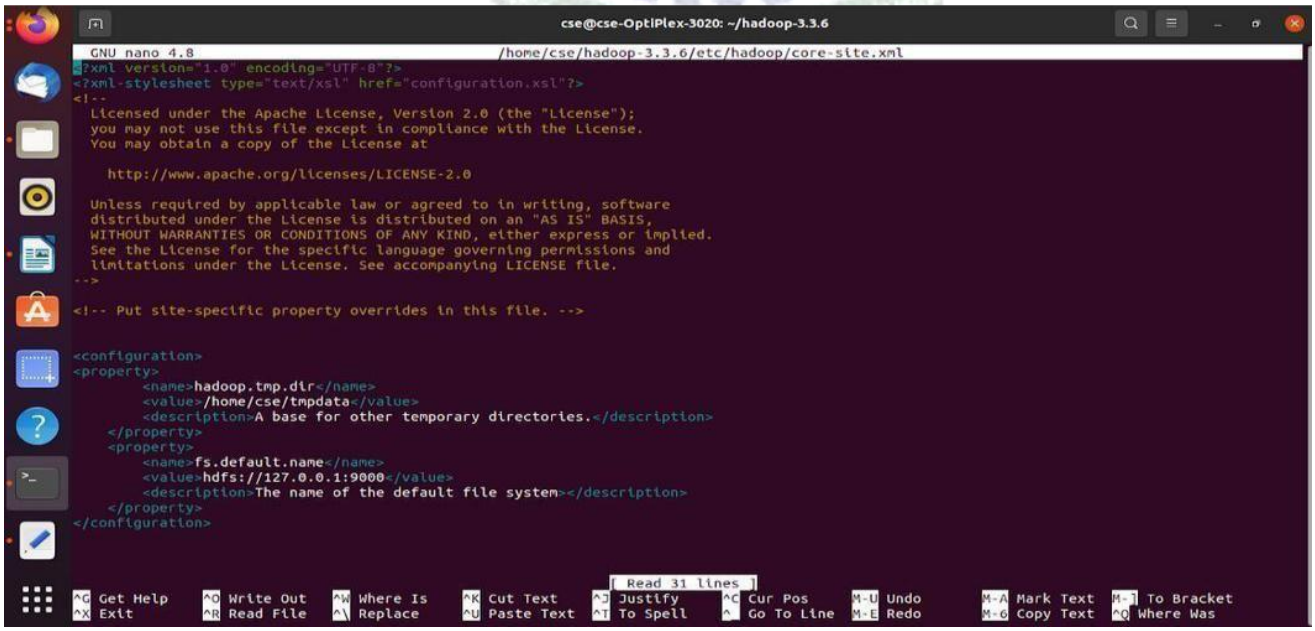
<property><name>mapreduce.framework.nam
e</name><value>yarn</value></property>



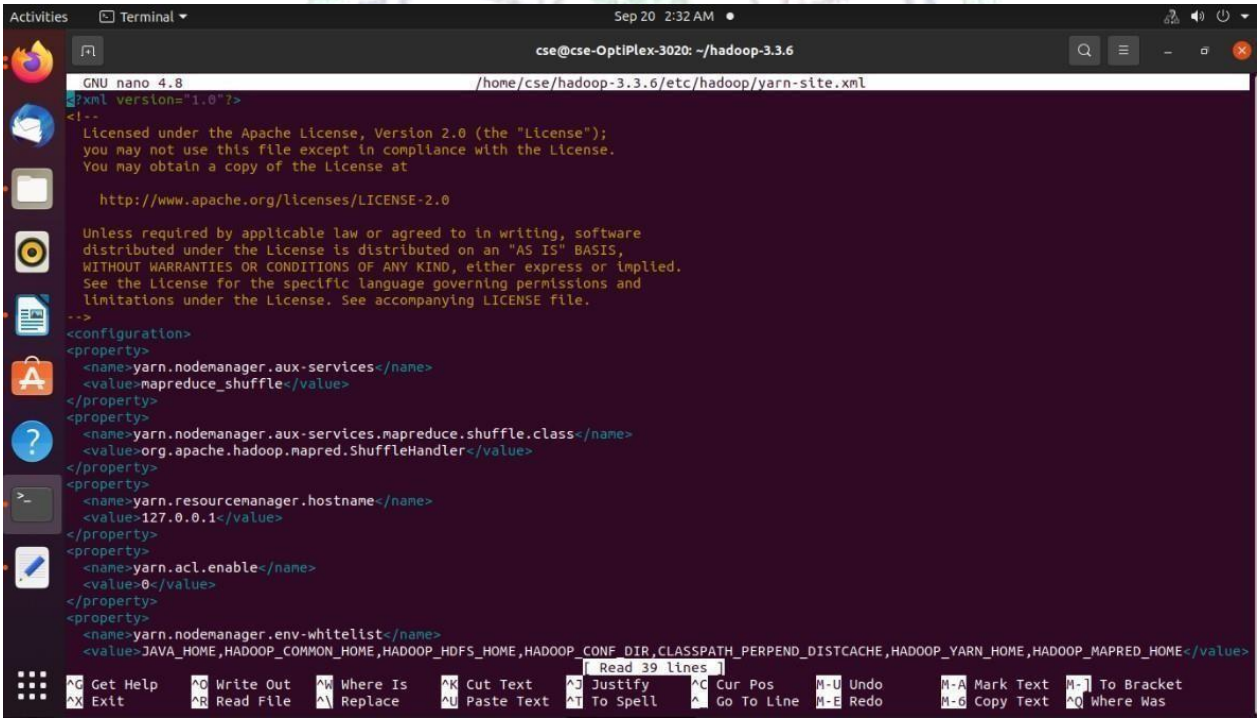
Edit yarn-site.xml

$sudonano $HADOOP_HOME/etc/hadoop/yarn-site.xml

#Add below linesin this file(between "<configuration>"and "<"/configuration>")



```
<property>
  <name>yarn.node manager.aux-
  services</name><value>mapreduce_shuffle</va
  lue>
</property>
<property>
  <name>yarn.node manager.aux-
  services.mapreduce.shuffle.class</name><value>org.apache.hadoop.map
  red.ShuffleHandler</value>
</property>
<property>
  <name>yarn.resource manager.hostname</name><value>127.0.0.1</value>
</property>
<property>
  <name>yarn.acl.enable</name>
  <value>0</value>
</property>
<property>
  <name>yarn.node manager.env-
whitelist</name><value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDF
S_HOME,HADO
OP_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HA
DOOP_MAPRED_HOME</value>
</property>
```



Step 2: Start the cluster

We willnow start the single node cluster withthe following commands.

a) Format the name node

$hdfs name node –format

```
17/11/06 01:55:11 INFO blockmanagement.BlockManager: encryptDataTransfer        = false
17/11/06 01:55:11 INFO blockmanagement.BlockManager: maxNumBlocksToLog          = 1000
17/11/06 01:55:12 INFO namenode.FSNamesystem: fsOwner             = hduser (auth:SIMPLE)
17/11/06 01:55:12 INFO namenode.FSNamesystem: supergroup          = supergroup
17/11/06 01:55:12 INFO namenode.FSNamesystem: isPermissionEnabled = true
17/11/06 01:55:12 INFO namenode.FSNamesystem: HA Enabled: false
17/11/06 01:55:12 INFO namenode.FSNamesystem: Append Enabled: true
17/11/06 01:55:12 INFO util.GSet: Computing capacity for map INodeMap
17/11/06 01:55:12 INFO util.GSet: VM type       = 32-bit
17/11/06 01:55:12 INFO util.GSet: 1.0% max memory 966.7 MB = 9.7 MB
17/11/06 01:55:12 INFO util.GSet: capacity      = 2^21 = 2097152 entries
17/11/06 01:55:12 INFO namenode.FSDirectory: ACLs enabled? false
17/11/06 01:55:12 INFO namenode.FSDirectory: XAttrs enabled? true
17/11/06 01:55:12 INFO namenode.NameNode: Caching file names occurring more than 10 times
17/11/06 01:55:12 INFO util.GSet: Computing capacity for map cachedBlocks
17/11/06 01:55:12 INFO util.GSet: VM type       = 32-bit
17/11/06 01:55:12 INFO util.GSet: 0.25% max memory 966.7 MB = 2.4 MB
17/11/06 01:55:12 INFO util.GSet: capacity      = 2^19 = 524288 entries
17/11/06 01:55:12 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
17/11/06 01:55:12 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
17/11/06 01:55:12 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
17/11/06 01:55:12 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
17/11/06 01:55:12 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
17/11/06 01:55:12 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
17/11/06 01:55:12 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
17/11/06 01:55:12 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
17/11/06 01:55:12 INFO util.GSet: Computing capacity for map NameNodeRetryCache
17/11/06 01:55:12 INFO util.GSet: VM type       = 32-bit
17/11/06 01:55:12 INFO util.GSet: 0.029999999329447746% max memory 966.7 MB = 297.0 KB
17/11/06 01:55:12 INFO util.GSet: capacity      = 2^16 = 65536 entries
17/11/06 01:55:12 INFO namenode.FSImage: Allocated new BlockPoolId: BP-2001603931-127.0.1.1-1509962112981
17/11/06 01:55:13 INFO common.Storage: Storage directory /home/hduser/hdata/dfs/name has been successfully formatted.
17/11/06 01:55:13 INFO namenode.FSImageFormatProtobuf: Saving image file /home/hduser/hdata/dfs/name/current/fsimage.ckpt_0000000000000000000 u
sing no compression
17/11/06 01:55:13 INFO namenode.FSImageFormatProtobuf: Image file /home/hduser/hdata/dfs/name/current/fsimage.ckpt_0000000000000000000 of size
323 bytes saved in 0 seconds.
17/11/06 01:55:13 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
17/11/06 01:55:13 INFO util.ExitUtil: Exiting with status 0
17/11/06 01:55:13 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
```

b)Start the HDFS

$start-all.sh

c)Verify if all process started $ jps

6775 Data Node

7209 Resource Manager

7017 Secondary Name Node

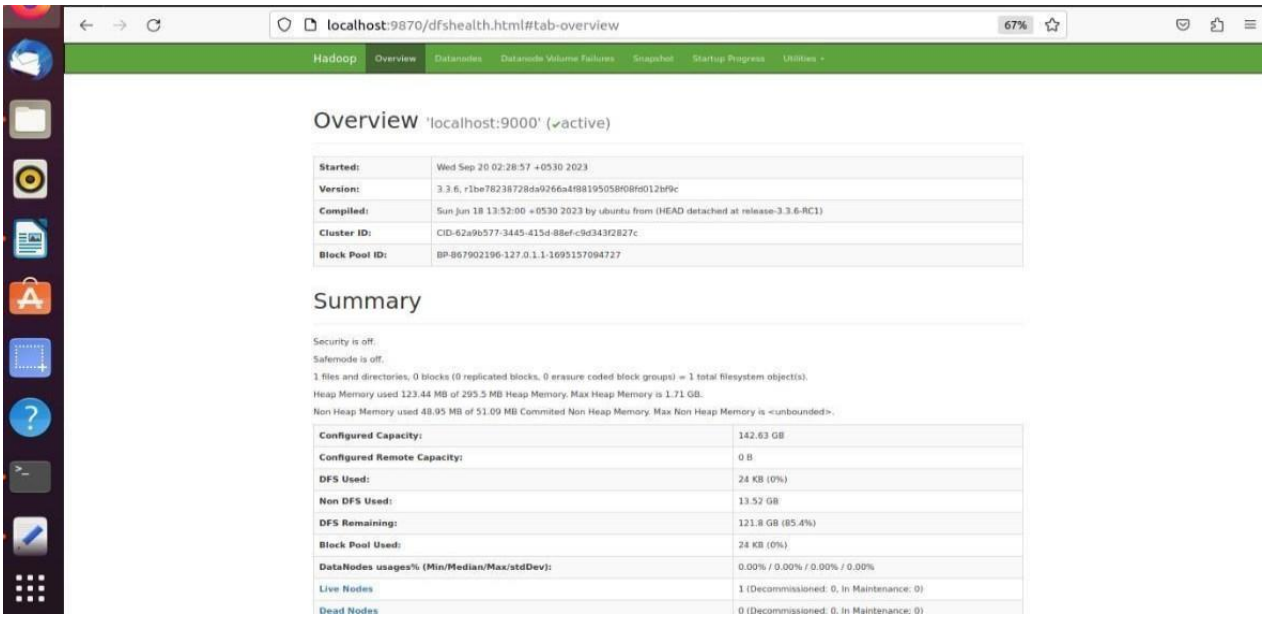
6651 Name Node

7339 Node Manager

7663 Jps

```
cse@cse-OptiPlex-3020:~/hadoop-3.3.6$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as cse in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [cse-OptiPlex-3020]
cse-OptiPlex-3020: Warning: Permanently added 'cse-optiplex-3020' (ECDSA) to the list of known hosts.
Starting resourcemanager
Starting nodemanagers
cse@cse-OptiPlex-3020:~/hadoop-3.3.6$ jps
3969 SecondaryNameNode
3587 NameNode
4166 ResourceManager
4297 NodeManager
3723 DataNode
4637 Jps
cse@cse-OptiPlex-3020:~/hadoop-3.3.6$
```

d)Web interface-For viewing Web UI of Name Node visit :(http://localhost:9870)



Result:

Downloaded and installed Hadoop and also understand different Hadoop modes. Startup scripts, Configuration files are successfully implemented

EXP NO:2 DATE:	Hadoop Implementation of file management tasks, such as Adding files and directions, retrieving files and Deleting file

AIM:

To implement the following file management tasks in Hadoop:

1. Adding files and directories
2.Retrieving
files
3.Deleting
Files

DESCRIPTION:-HDFS is a scalabled is tribute file system designed to scale to peta bytes of data while running on top of the underlying file system of the operating system. HDFS keeps track of where the data resides in a network by associating the name of its rack (or network switch)with the data set. This allows Hadoop to efficiently schedule tasks to those nodes that contain data, or which are nearest to it, optimizing band width utilization. Hadoop provides a set of command line utilities that work similarly to the Linux file commands, and serve as your primary interface with HDFS. We‘re going to have a look into HDFS by interacting with it from the command line.We will take a look at the most common file management tasks in Hadoop,
which include:

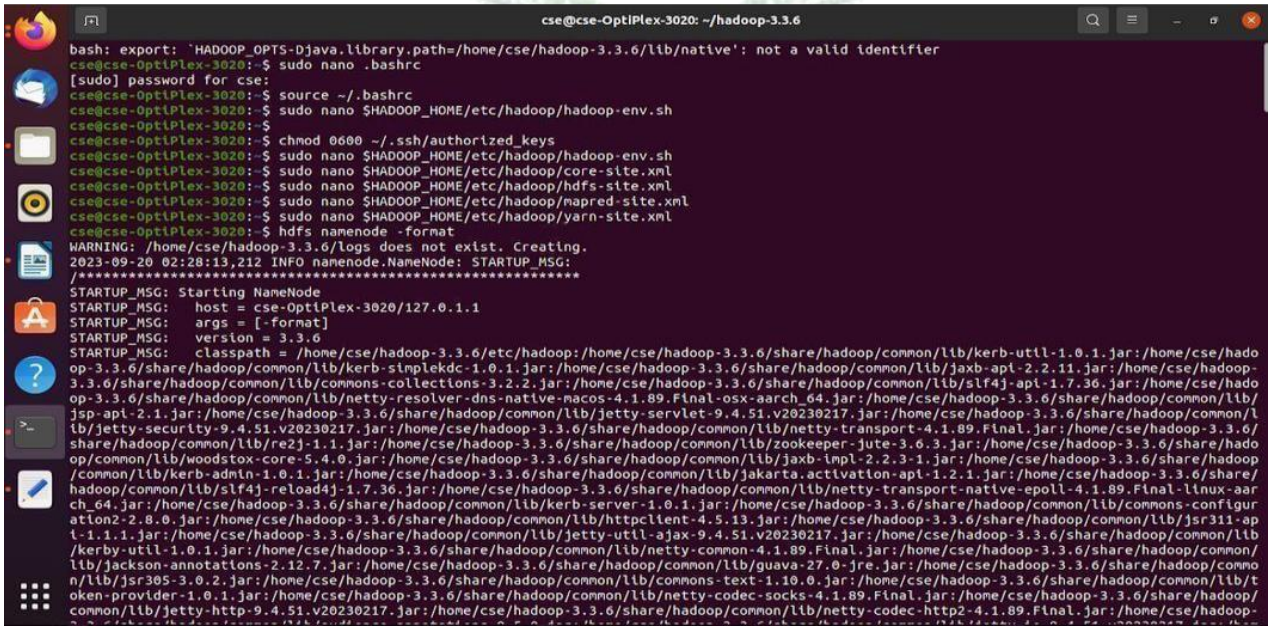
1.Adding files and directories to HDFS
2.Retrieving files from HDFS to local file system
3. Deleting Fileseleting files fromHDFS

SYNTAXANDCOMMANDSTOADD,RETRIEVEANDDELETEDATA FROMHDFS

Step 1:StartingHDFS

InitiallyyouhavetoformattheconfiguredHDFSfilesystem,opennamenode(HDFSserver), and execute the following command.

$ hadoopnamenode -format



AfterformattingtheHDFS,startthedistributedfilesystem.Thefollowingcommandwillstart the namenodeas well as the data nodes as cluster.$start-dfs.sh

Listing Files inHDFS

Afterloading the information in theserver, we can findthe listoffiles in a directory, statusof a file, using ls Givenbelow isthe syntax of ls thatyou can pass toa directory or a filename as an argument.

```
ambal2@Ubuntu:~$ hadoop fs -ls
2023-10-12 11:37:33,233 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
Found 4 items
drwxr-xr-x   - ambal2 supergroup          0 2023-10-11 13:04 bigdata
drwxr-xr-x   - ambal2 supergroup          0 2023-10-12 11:35 new
drwxr-xr-x   - ambal2 supergroup          0 2023-10-09 11:39 sqoop
drwxr-xr-x   - ambal2 supergroup          0 2023-10-09 12:41 sqoop1
ambal2@Ubuntu:~$
```

$ $HADOOP_HOME/bin/hadoopfs-ls <args>

Inserting Data into HDFS

Assume we have data in the file called file.txt in the local system which is ought to be saved in the hdfs file system.Follow the steps given below to insert the required file in the Hadoop Filesystem

Step-2:Adding Files and Directories toHDFS

$ $HADOOP_HOME/bin/hadoopfs-mkdir /user/input

```
ambal2@Ubuntu:~$ hadoop fs -mkdir new
2023-10-12 11:33:52,575 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where


ambal2@Ubuntu:~$
ambal2@Ubuntu:~$
ambal2@Ubuntu:~$ hadoop fs -ls
2023-10-12 11:37:33,233 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where
Found 4 items
drwxr-xr-x   - ambal2 supergroup          0 2023-10-11 13:04 bigdata
drwxr-xr-x   - ambal2 supergroup          0 2023-10-12 11:35 new
drwxr-xr-x   - ambal2 supergroup          0 2023-10-09 11:39 sqoop
drwxr-xr-x   - ambal2 supergroup          0 2023-10-09 12:41 sqoop1
```

Transfer and store a data file from local systems to the Hadoop file system using the put command.

$$HADOOP_HOME/bin/hadoop fs-put /home/file.txt /user/input **Step 3 :You can verifythe file usingls command.**

$ $HADOOP_HOME/bin/hadoopfs-ls /user/input

Step 4RetrievingData from HDFS

AssumewehaveafileinHDFScalledoutfile.Givenbelowisasimpledemonstrationfor retrieving the requiredfile from theHadoop file system.

Initially, view the datafrom HDFS using cat command.

$ $HADOOP_HOME/bin/hadoopfs-cat /user/output/outfile

Get thefile from HDFS to the local file system using get command.

$ $HADOOP_HOME/bin/hadoopfs-get/user/output/ /home/hadoop_tp/

Step-5: Deleting Filesfrom HDFS

$ hadoopfs -rmfile.txt

Step 6:Shutting Downthe HDFS

Youcan shut down theHDFS byusing the following command.

$ stop-dfs

Result:

Thus the Installing of Hadoop in three operating modes has been successfully completed

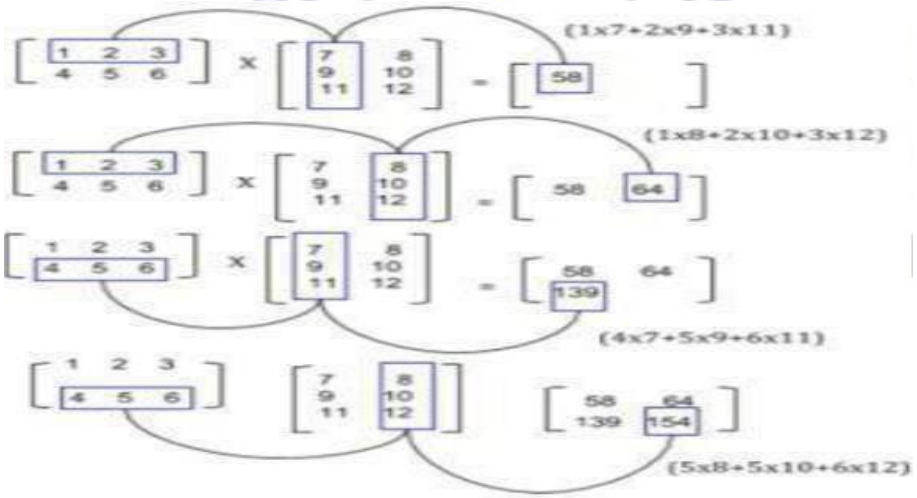
EXPT.NO.3(a)	Implement of Matrix Multiplication with Hadoop Map Reduce
DATE:	

AIM:

To Develop a Map Reduce program to implement Matrix Multiplication.

Description

In **mathematics**, **matrix multiplication** or the **matrix product** is a binary operation that produces a matrix from two matrices. The definition is motivated by linear equations and linear transformations on vectors, which have numerous applications in applied mathematics, physics, and engineering. In more detail, if **A**isan $n \times m$ matrix and **B** is an $m \times p$ matrix, their matrix product **AB** is an $n \times p$ matrix, in which the men tries across a row of **A** are multiplied with the men tries down a column of **B** and summed to produce an entry of **AB**. When two linear transformations are represented by matrices, then the matrix product represents the composition of the two transformations



Algorithm for Map Function.

a. for each element m_{ij} of M do produce (key, value) pairs as ((i,k),(M,j,m_{ij}),for k=1,2,3,..upto the number of columns of N

b. for each element njk of N do
produce(key, value)pairs as((i,k),(N,j,N_{jk}),for i=1,2,3,..Up to the number of rows of M.

c.return Set of (key,value)pairs that each key(i,k),has list with values (M,j,m_{ij})and(N,j,n_{jk}) for all possible values of j.

Algorithm for Reduce Function.

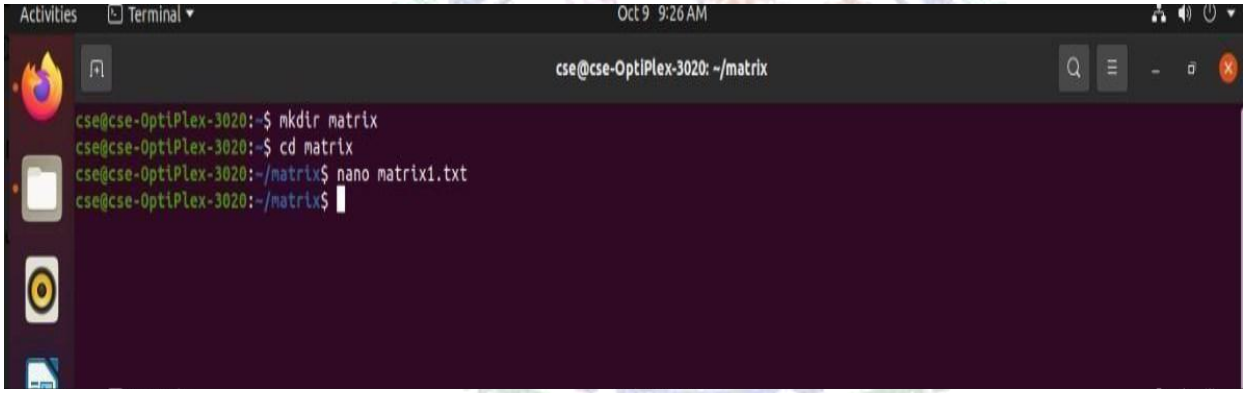
d.for each key(i,k) do

e.sort valuesb egin with M by j in list M sort values begin with N by jin list N multiply mij and njk for jth value of each list

f.sum up mij x njk return(i,k), Σj=1mij x njk

Step 1. Creating directory for matrix

Then open matrix1.txt and matrix2.txt put the values in that text files



Step 2. CreatingMapper file for Matrix Multiplication.

```
#!/usr/bin/envpythonimport
sys
cache_info=open("cache.txt").readlines()[0].split(",")
row_a,col_b=map(int,cache_info)for
line in sys.stdin:
        matrix_index,row,col,value=line.rstrip().split(",")if
        matrix_index== "A":
                for i in xrange(0,col_b):
                        key=row+","+str(i)print
                        "%s\t%s\t%s"%(key,col,value)
        else:
                for j in x range(0,row_a):
                        key=str(j)+","+colprint
                        "%s\t%s\t%s"%(key,row,value)
```

Step 3. Creating reducer file for Matrix Multiplication.

```
#!/usr/bin/envpythonimport
sys
from operator import item
getter prev_index= None
value_list= []
For line insys.stdin:
curr_index,index,value=line.rstrip().split("\t")index,
value=map(int,[index,value])if curr_index== prev_index:
        value_list.append((index,value))
    else:
        if prev_index:value_list=sorted(value_list,key=itemgetter(0))
        i=0
        result = 0
        whilei<len(value_list) - 1:
                ifvalue_list[i][0]==value_list[i+1][0]:
                result+=value_list[i][1]*value_list[i+1][1]i+=2
                else:
                    i += 1
```

```
            print "%s,%s"%(prev_index,str(result))
        prev_index= curr_index
        value_list =[(index,value)]

  if curr_index==prev_index:
  value_list=sorted(value_list,key=itemgetter(0))i=0
     result = 0 whilei<len(value_list) - 1:
        if  value_list[i][0]==  value_list[i+1][0]:  result
           +=value_list[i][1]*value_list[i+1][1]i+= 2
        else:
           i += 1
     print "%s,%s"%(prev_index,str(result))
```

Step 4.To view this file using cat command

$cat *.txt |python mapper.py

$ chmod +x~/Desktop/mr/matrix/Mapper.py$

chmod +x~/Desktop/mr/matrixl/Reducer.py

$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \

>-input /user/cse/matrices/\

>-output /user/cse/mat_output \

>-mapper~/Desktop/mr/matrix/Mapper.py \

>-reducer~/Desktop/mr/matrix/Reducer.py

Step5:Toviewthisfulloutput

Result:

Thus the Map Reduce program to implement Matrix Multiplication was successfully executed

EXP NO:4 **DATE:**	Run a basicWord Count MapReduce program to understandMap Reduce Paradigm

AIM:

To Develop a Map Reduce program to calculate the frequency of a given word in

A given file **Map Function** – It takes a setof data and converts it into another setof data, where individual elements are broken down into tuples (Key-Value pair).

Example– (Map function inWordCount)

Input

Set of data

Bus, Car,bus, car, train, car, bus,car, train, bus,TRAIN,BUS,buS, caR, CAR,car, BUS, TRAIN

Output

Convertintoanotherset

of data(Key,Value)

(Bus,1), (Car,1), (bus,1), (car,1), (train,1), (car,1), (bus,1), (car,1), (train,1), (bus,1), (TRAIN,1),(BUS,1), (buS,1), (caR,1), (CAR,1), (car,1), (BUS,1), (TRAIN,1)

Reduce Function– Takes the output from Map as an input and combines those data tuples into a smaller set of tuples.

Example – (Reducefunction inWordCount)

Input Set of Tuples(output of Map function)

(Bus,1), (Car,1), (bus,1), (car,1), (train,1), (car,1), (bus,1), (car,1), (train,1), (bus,1),(TRAIN,1),(BUS,1), (buS,1),(caR,1),(CAR,1),(car,1),(BUS,1),(TRAIN,1)

1. **Splitting –** The splitting parameter can be anything, e.g. splitting by space,comma, semicolon, or even by a new line (‘\n’).
2. **Mapping –** as explained above
3. Intermediate splitting – the entire process in parallel on different clusters. In orderto group them in “Reduce Phase” the similar KEY data should be on same cluster.
4. **Reduce –** it is nothing but mostly group by phase
5. **Combining –** The last phase where all the data (individual result set from eachcluster) is combine together to form a Result

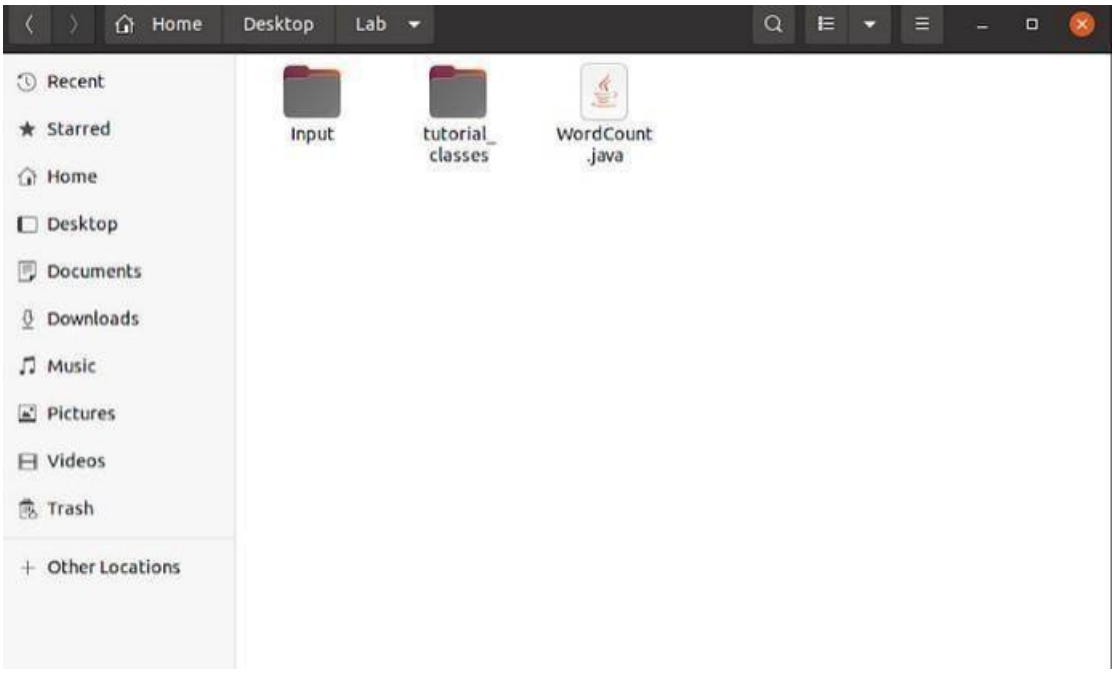
Now Let’s See the Word Count Program in Java

Step1 :Make sure Hadoop and Java are installed properly

hadoop version

javac –version

Step 2. Create a directory on the Desktop named Lab and inside it create two folders;
one called “Input” and the other called “tutorial_classes”.
[You can do this step using GUI normally or through terminal commands]
cd Desktop
mkdir Lab
mkdir Lab/Input
mkdir Lab/tutorial_classes
Step 3. Add the file attached with this document “WordCount.java” in the directory Lab
Step 4. Add the file attached with this document “input.txt” in the directory Lab/Input.



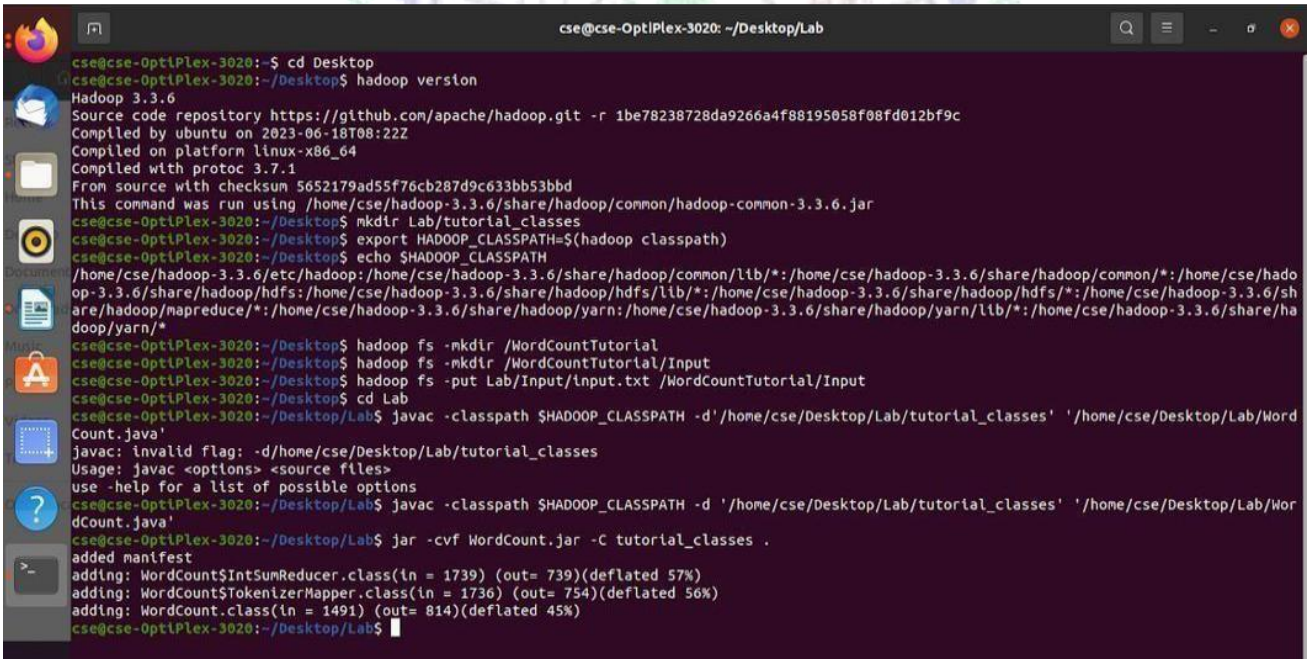
Step5. Type the following command to export the hadoop class path into bash.

export HADOOP_CLASSPATH=$(hadoopclasspath)

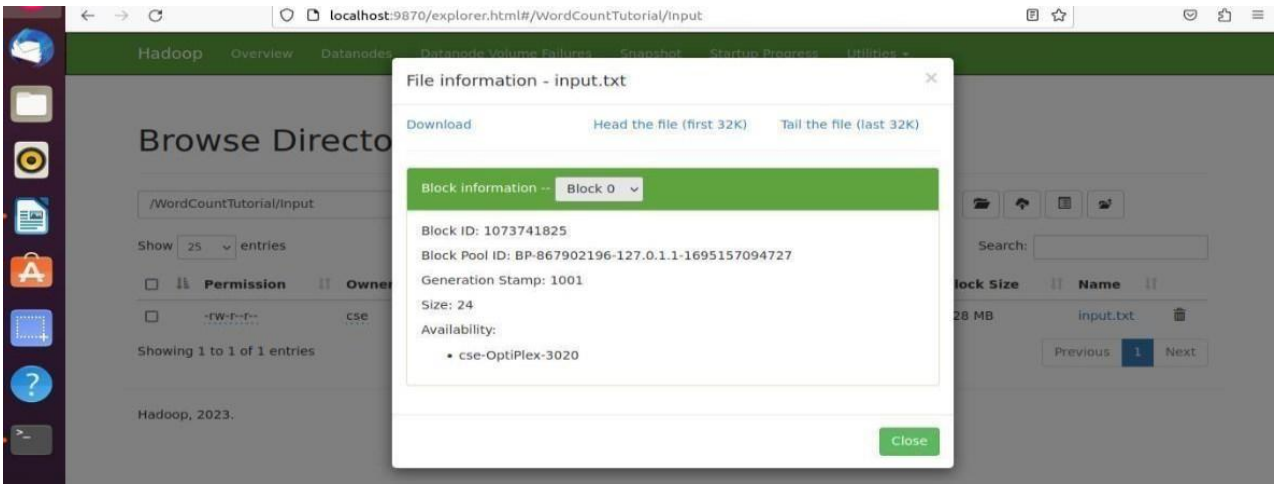
Make sureit is now exported.

echo$HADOOP_CLASSPATH

Step6.It is time to create these directories on HDFS rather than locally. Type the following commands. Hadoop fs-mk dir /Word Count Tutorial hadoop fs-mkdir /Word Count Tutorial/Input hadoop fs -put Lab/Input/input.txt /Word Count Tutorial/Input



Step7.Go to local host: 9870 from the browser, Open “Utilities → Browse File System ”and you should see the directories and files we placed in the file system.



Step8.Then, back to local machine where we will compile the Word Count.java file.

Assuming we are currently in the Desktop directory.

cdLab

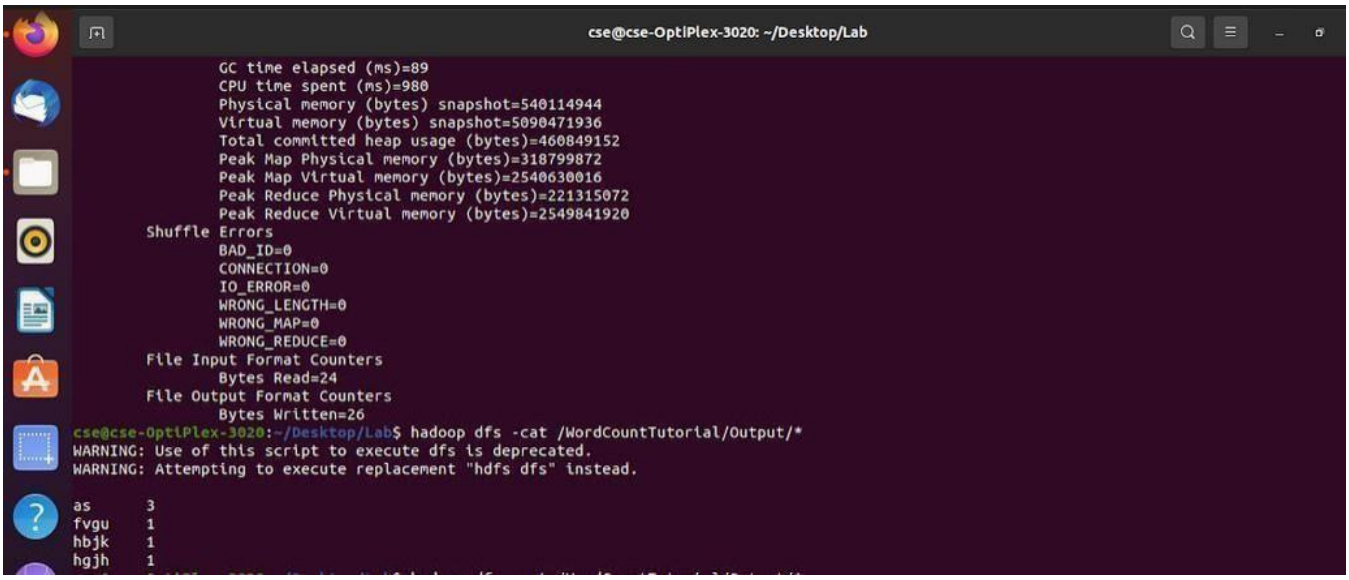
Java c-class path$ HADOOP_CLASS PATH-d tutorial classes Word Count.java

Put the output files in one jar file(There is a dot at the end)

Jar-cvf Word Count.jar-Ctutorial_classes.Step9.Now, we run the jar file on Hadoop.

Hadoop jar Word Count.jar Word Count/Word Count Tutorial/Input /Word Count Tutorial/Output

Step10.Output theresult:



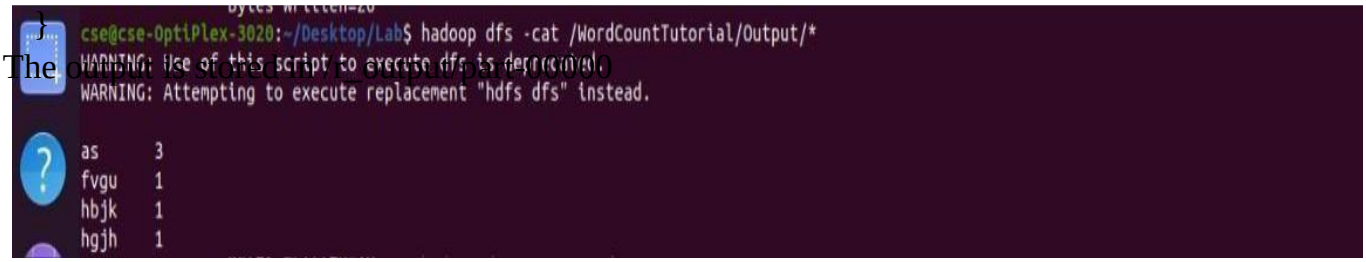
Hadoopd fs-cat/Word Count Tutorial/Output/*

Program:Step5.Type following Program:

```
package Package Demo;
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
```

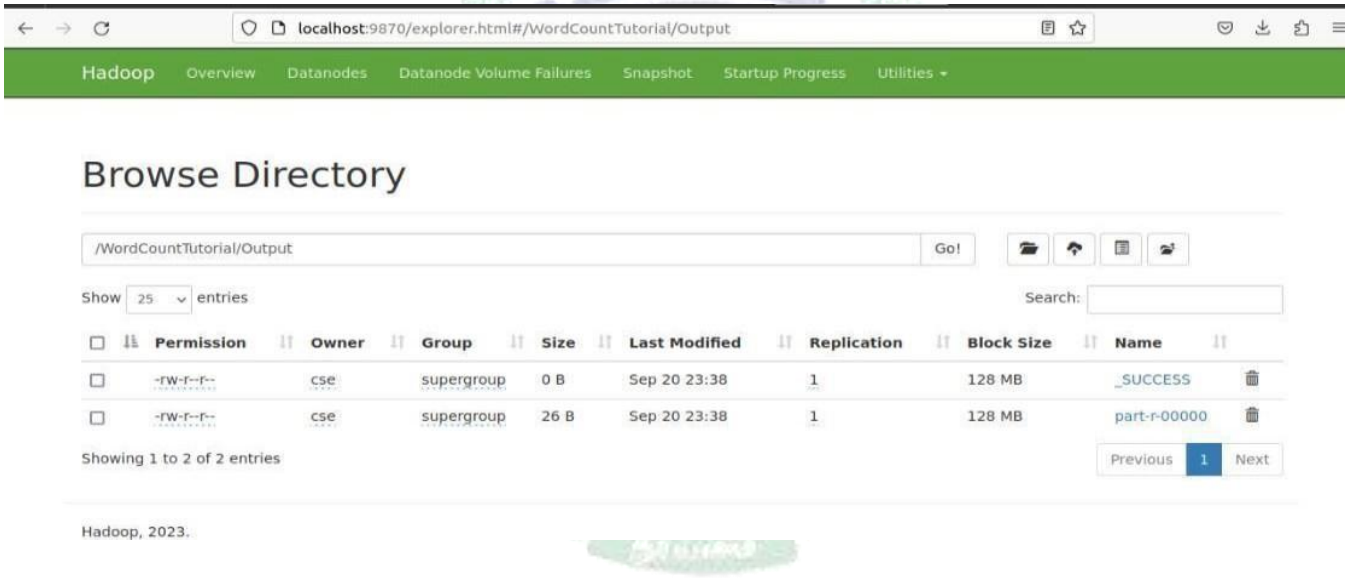
```
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
 import org.apache.hadoop.mapreduce.lib.input.File Input Format;
import org.apache.hadoop.mapreduce.lib.output.File Output Format;
import org.apache.hadoop.util.Generic Options Parser;
public class WordCount {
public static voidmain(String [] args) throws
Exception {
Configuration c=newConfiguration();
String[]files=new
Generic Options Parser(c,args).get Remaining
Args();Path input=new Path(files[0]); Path output=new
Path(files[1]); Jobj=new
Job(c,"word count");
j.set Jar B Class (Word Count.c lass);
j.set Mapper Class(Map For Word Count.class);
 j.set Reducer Class(Reduce For Word Count.class);
j.set Output Key Class (Text.class);
j.set Output Value Class(Int Writable.class);
File Input Format.add Input Path(j, input);
File Output Format.setOutputPath(j, output);
System.exit(j.wait For
Completion(true)?0:1); }
Public static class Map For Word Countext ends Mapper
<LongWritable,Text,Text,IntWritable>
{ public void map
(LongWritablekey,Textvalue,Contextcon) throws IO Exception, Interrupted Exception
{
Stringline=value.toString();
String[]words=line.split(",");
for(Stringword: words
)
{
   Text output Key= new Text (word.to Upper
   Case().trim());Int Writable output Value=new Int
   Writable(1);con.write(output Key,output Value);
}
}
}
public static class Reduce For Word Count extends Reducer<Text, Int Writable,Text,Int
Writable> {
public void reduce(Text word, Iterable<Int Writable> values, Context con) throws
IOException, Interrupted Exception
{int sum= 0;
```

```
for(Int Writable value
:values) {
sum +=
value.get(); }
con.write(word, new Int
Writable(sum)); }
}
```



The output is stored in /Output/part-00000

OUTPUT:



Result:

Thus the Word Count Map Reduce program to understand Map Reduce Paradigm was successfully executed

EXPT.NO.5(a) DATE:	Installation hive

AIM:

To installing Hive With example

Steps for hive installation

- Downloadand UnzipHive
- Edit .bashrcfile
- Edit hive-config.shfile
- Create Hivedirectoriesin HDFS
- Initiate Derby database•Configurehive-site.xml file **Step 1:**

download and unzip Hive

==========

=

wget

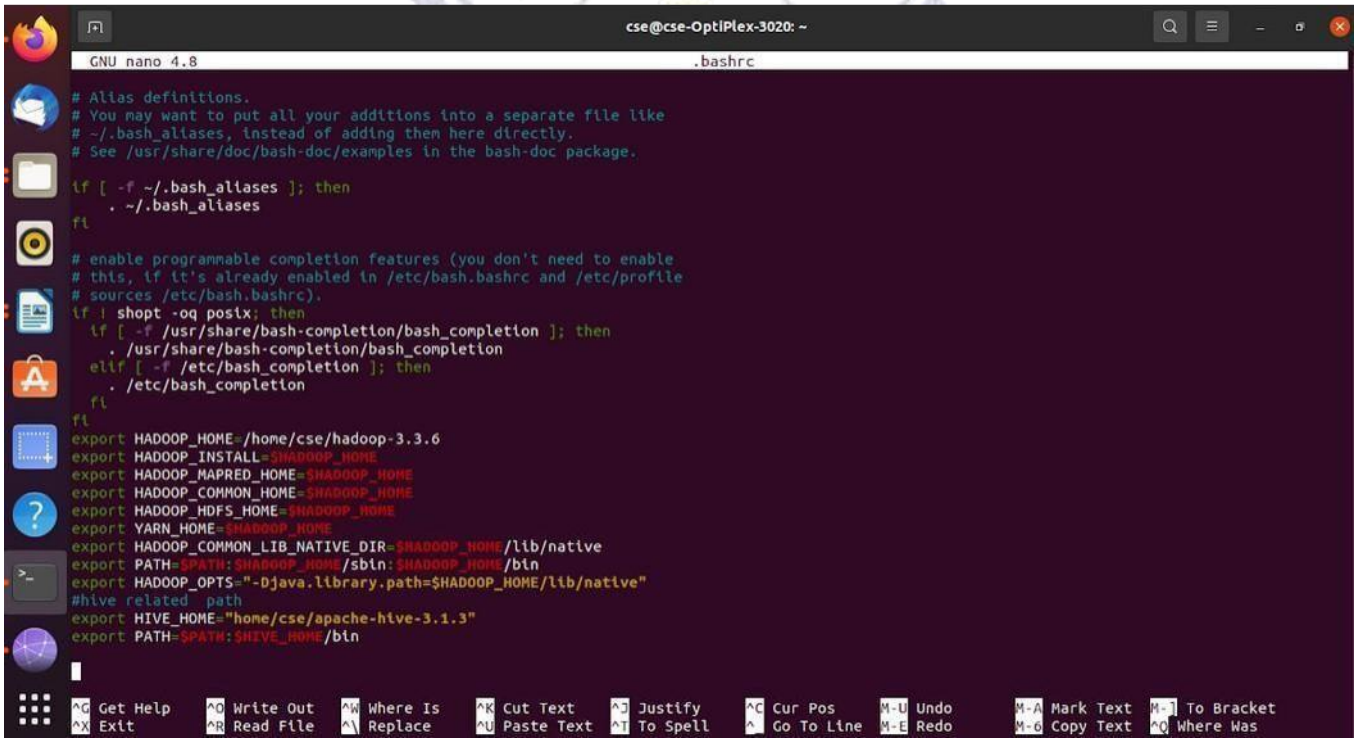
https://downloads.apache.org/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz tar xzf apache-hive-3.1.2-bin.tar.gz**step 2:**

Edit .bashrcfile

==========

= sudo nano.bashrc

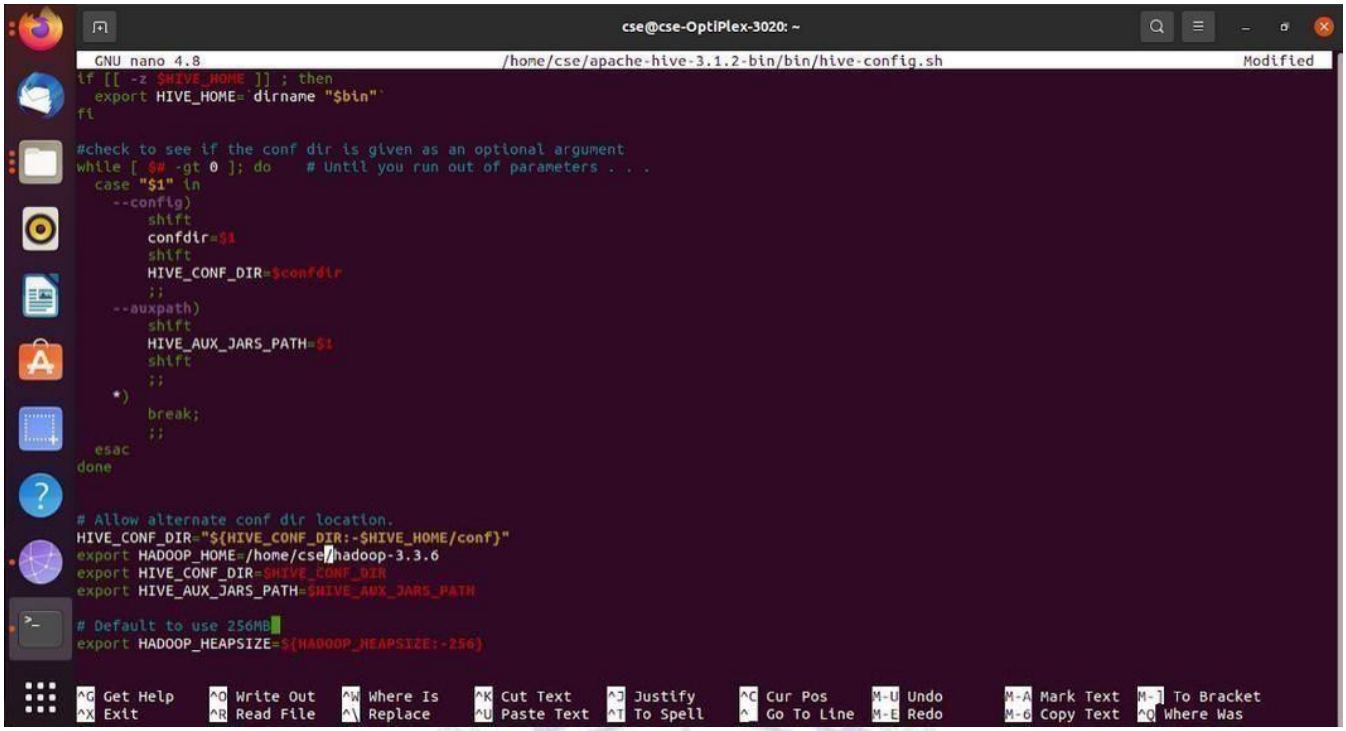
export HIVE_HOME=/home/hdoop/apache-hive-3.1.2-binexport PATH=$PATH:$HIVE_HOME/bin



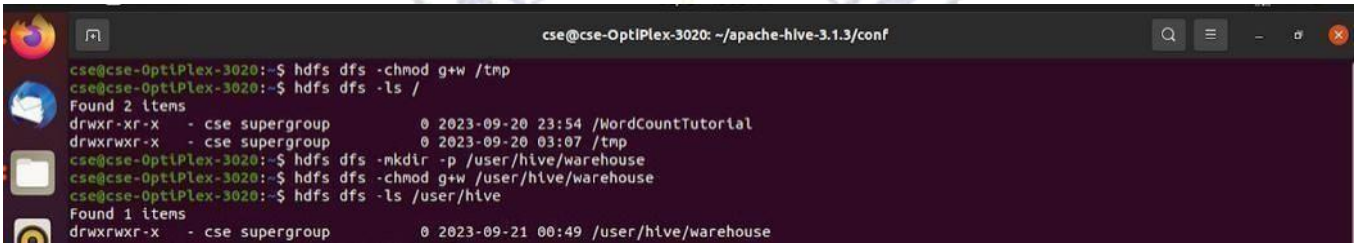
step

3: source~/.bashrc**step**

4:Edit hive-config.shfile ==========
sudonano$HIVE_HOME/bin/hive-config.shexport HADOOP_HOME=/home/cse/hadoop-3.3.



step 5:
Create Hivedirectoriesin HDFS
==========
hdfsdfs-mkdir/tmphdfsdfs-chmodg+w
/tmp hdfs dfs -mkdir -p
/user/hive/warehousehdfsdfs-chmodg+w
/user/hive/warehouse

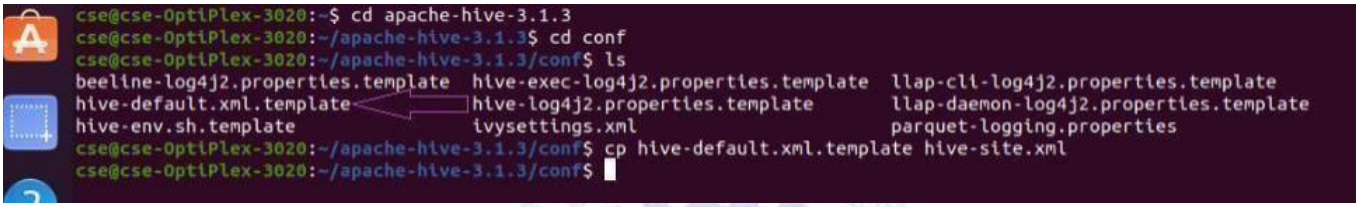


step 6:

Fixing guavaproblem–Additional step
==========

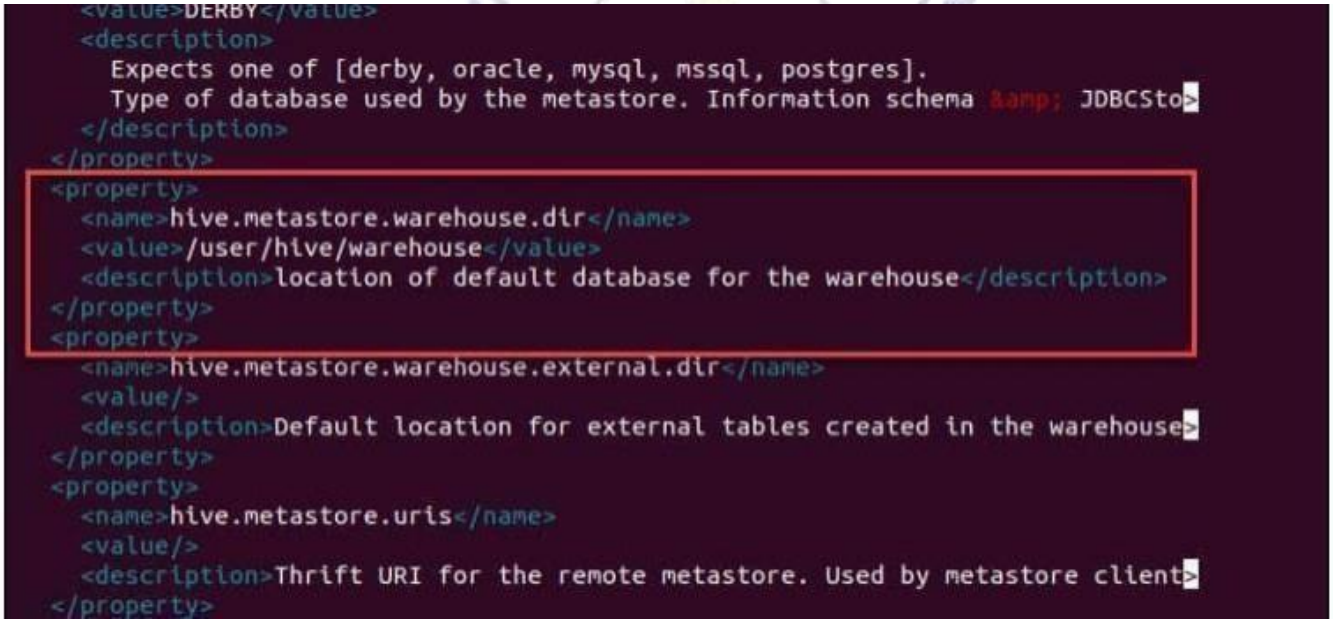
rm $HIVE_HOME/lib/guava-19.0.jar
cp $HADOOP_HOME/share/hadoop/hdfs/lib/guava-27.0-jre.jar $HIVE_HOME/lib/

step7:Configurehive-site.xmlFile(Optional) Use thefollowingcommandtolocatethecorrectfile:cd $HIVE_HOME/conf
List thefiles containedin the folderusingthe ls command



Use the hive-default.xml.templateto createthe hive-site.xml file:

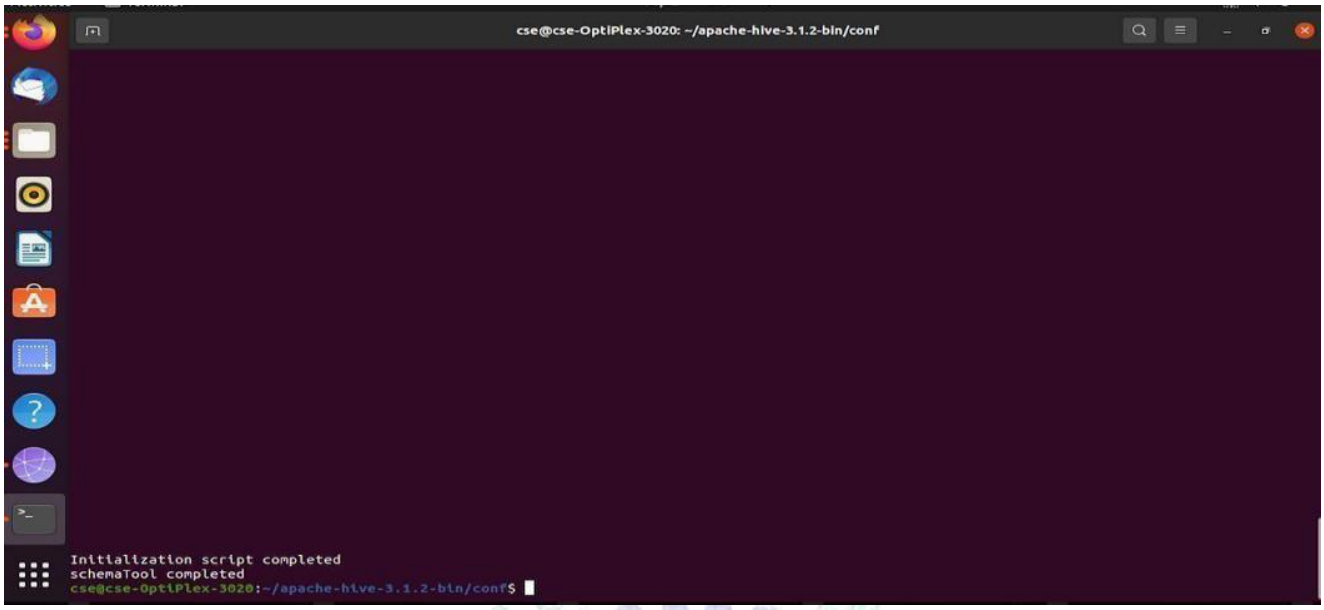
cp hive-default.xml.template hive-site.xml
Access the hive-site.xml fileusing thenanotext editor: sudo nano hive-site.xm

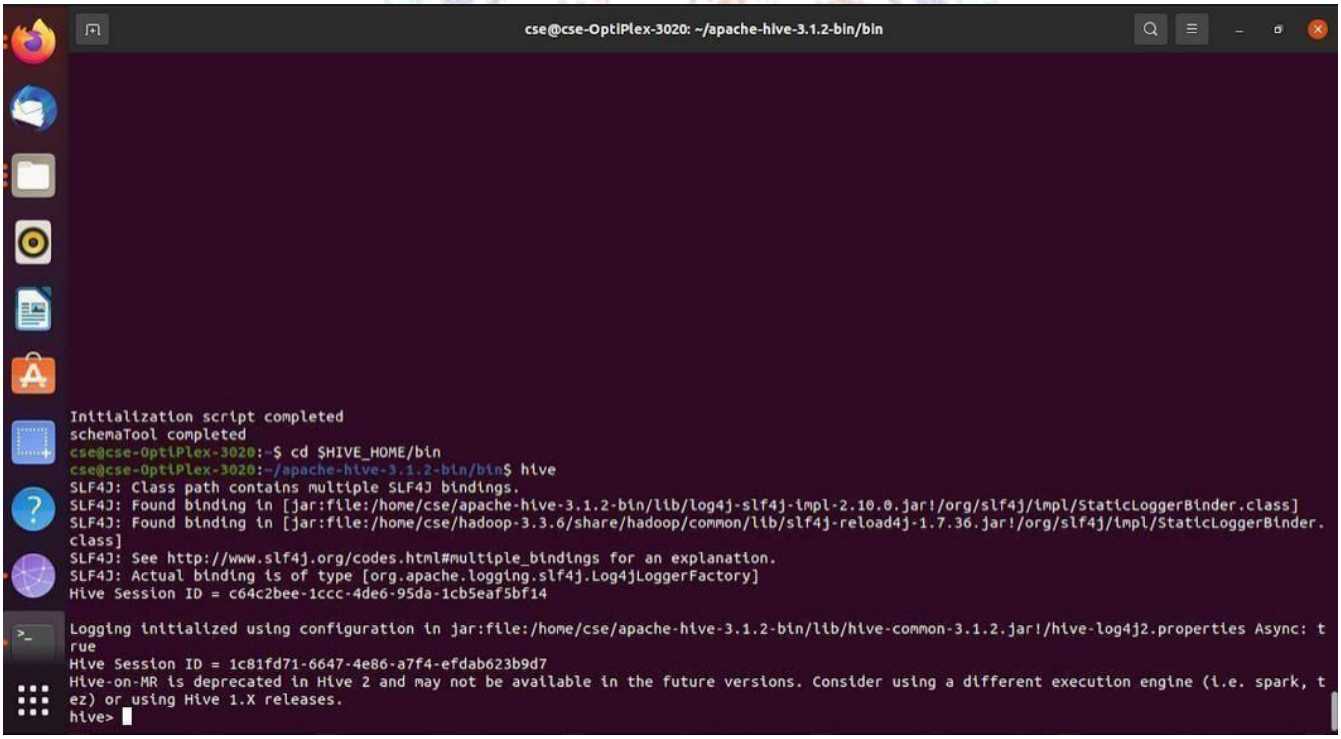


Step8: Initiate Derby Database

==========

$HIVE_HOME/bin/schematool-dbType derby –initSchem





Result:

Thus Installation hivewas successfully installed andexecuted

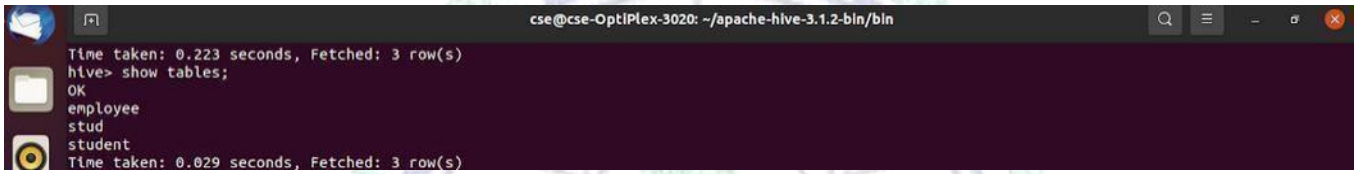
EXPT.NO.5(b)	Hive with examples
DATE:	

AIM:

To installing hive withexample

Create Database from Hive Bee line shell
1.Create database database_name; Ex:
>Create database Emp;
>use Emp;
>create table emp.employee(sno int,user String,city String)Row format delimited fields terminate dby /nstored as textfile;

Show Database



Result:
Thus Installation hive was successfully installed and executed with example

EXPT.NO.6(a)	Installationof HBase, Installing thrift along with Practice examples
DATE:	

AIM:

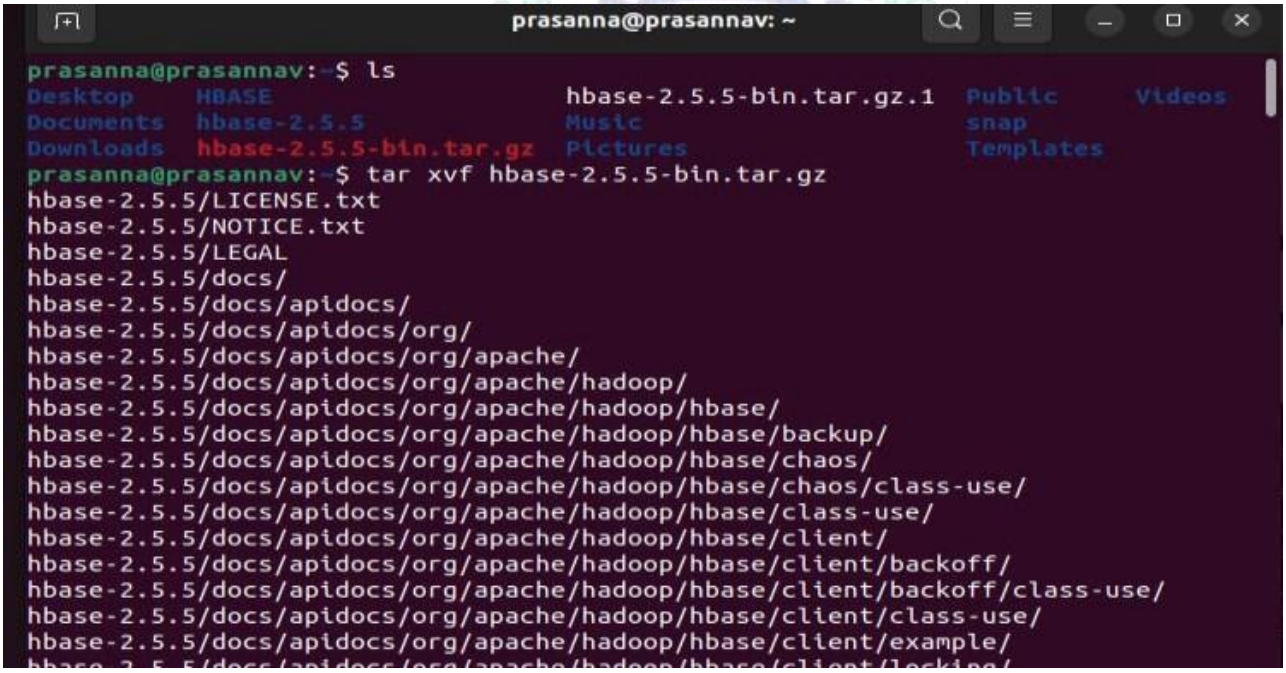
To Install HBase on Ubuntu 18.04HBasein Standalone Mode

PROCEDURE: Pre-requisite:

Ubuntu 16.04or higher installed on a virtual machine**.**

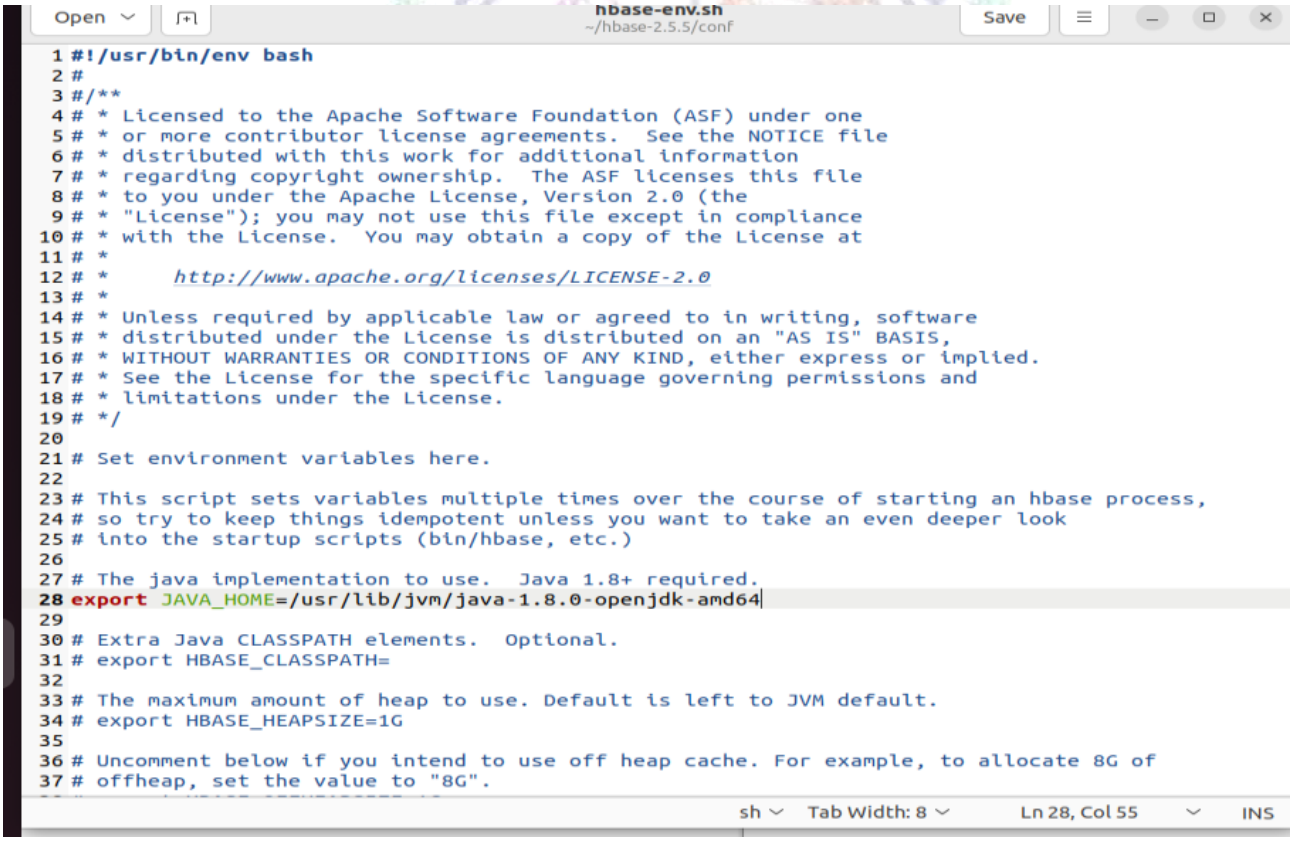
step-1:Make sure that java has installed in your machine to verify that run java–version
If any Error Occurred While Execute this command , then java is not installed in your system

To Install Java sudo apt install openjdk-8-jdk-y**Step-2:Download Hbase**
wgethttps://dlcdn.apache.org/hbase/2.5.5/hbase-2.5.5-bin.tar.gz
Step-3:Extract The hbase-2.5.5-bin.tar.gz file by using the command tar xvf hbase-2.5.5-bin.tar.gz



step-4: goto hbase2.5.5/conf folder and open hbase-env.shfile

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd6



step-5 :Edit .bashrc file

and then open.bashrc file and mention HBASE_HOME path as shown in below

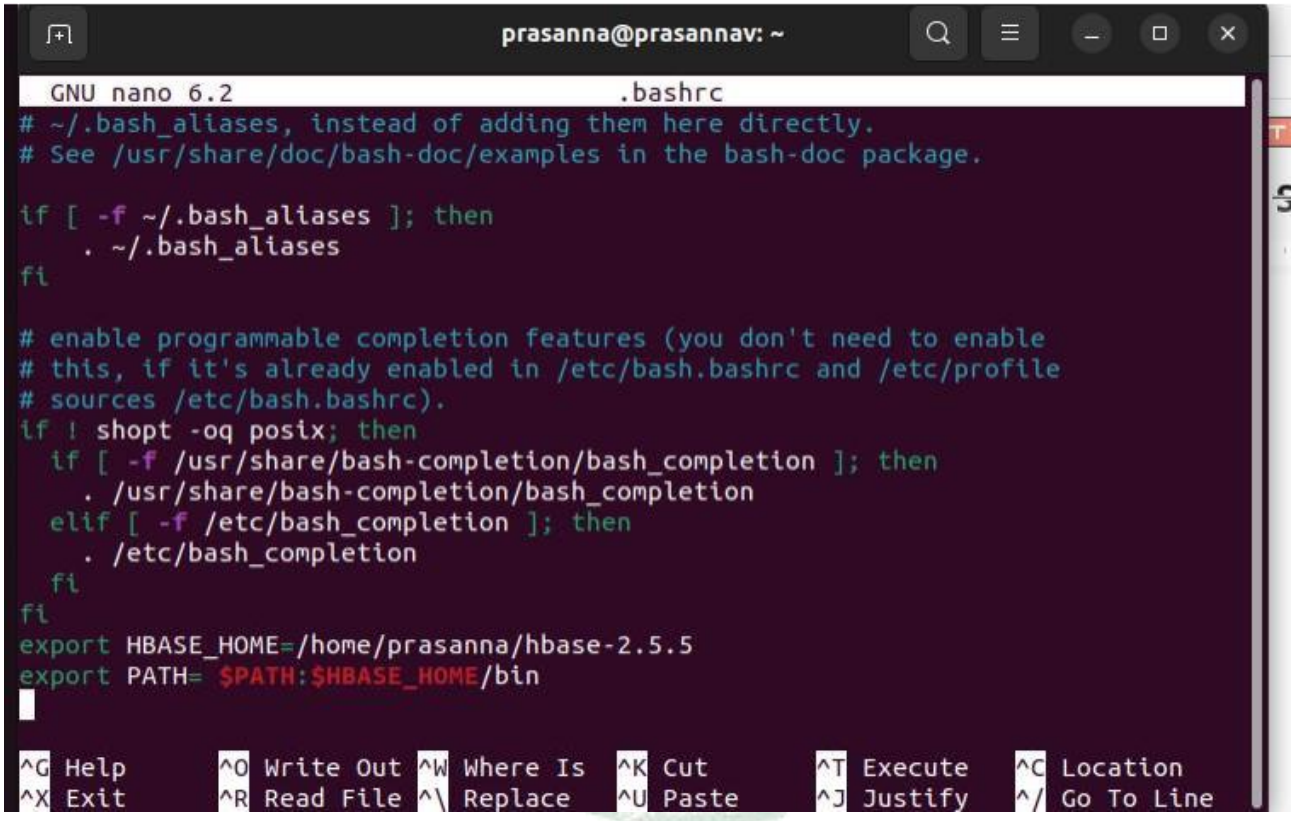
export HBASE_HOME=/home/prasanna/hbase-2.5.5hereyoucanchangename

according to your local machine name eg : export

HBASE_HOME=/home/<your_machine_name>/hbase-2.5.5

export PATH= $PATH:$HBASE_HOME/bin

Note:*make sure that the hbase-2.5.5 folder in home directory before setting HBASE_HOME path , if not then movethe hbase-2.5.5 file to home directory



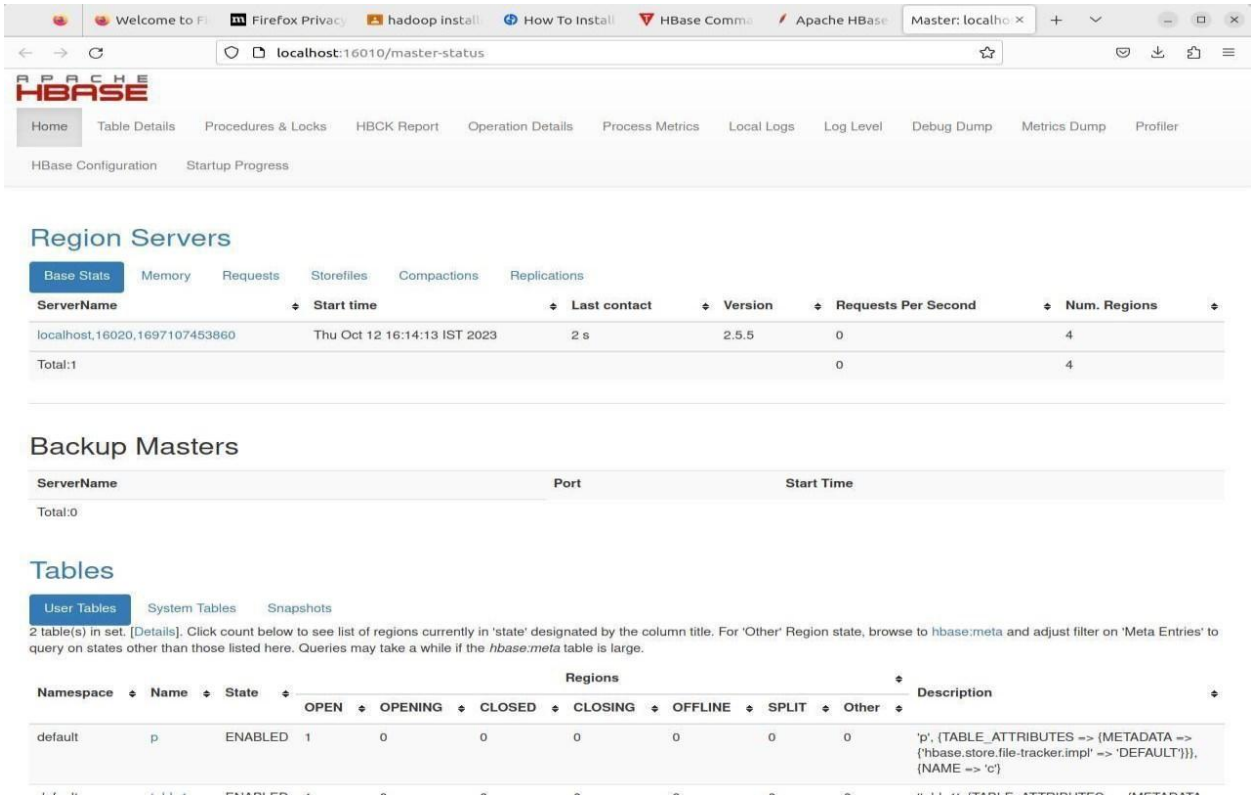
step-6 :Add properties in the hbase-site.xml

put the below property between the <configuratio></configuration> tag

<property>
<name>hbase.rootdir</name><value>file:///home/pra
sanna/HBASE/hbase</value></property>
<property><name>hbase.zookeeper.property.dataDir
</name><value>/home/prasanna/HBASE/zookeeper
</value>
</property>**step-7:GotoTo/etc/folderandrun the followingcommand and configure**

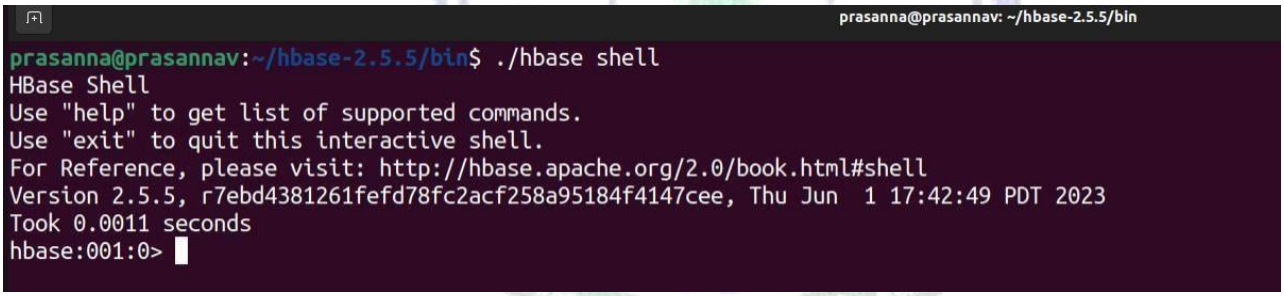
change inline no-2 by default the ipis 127.0.1.1
change it to 127.0.0.1 in second line only step-8:starting hbase goto hbase-2.5.5/bin folder

After this run jps command to ensure that hbase is running



run http://localhost:16010***to see hbase web UI***

step-9: accessing hbase shell by running ./hbase shell command



Result:

HBase was successfully installed on Ubuntu 18.0

EXPT.NO.6(b)	HBase, Installing thriftalong with Practiceexamples
DATE:	

Aim:

To Install HBase on Ubuntu 18.04 HBase in Standalone Mod

EXAMPLE

1) To create Table syntax: create
'Table_Name','col_fam_1','col_fam_1', ... 'col_fam-n'

Code :

create 'aamec','dept','year'

```
hbase:007:0> create 'aamec','dept','year'
2023-10-13 12:04:54,143 INFO  [main] client.HBaseAdmin (HBaseAdmin.java:postOper
ationResult(3591)) - Operation: CREATE, Table Name: default:aamec, procId: 100 c
ompleted
Created table aamec
Took 0.6840 seconds
=> Hbase::Table - aamec
hbase:008:0>
```

2) List All Tables code :list

```
hbase:010:0> list
TABLE
aamec
amazon
college
prasanna
table_name
5 row(s)
Took 0.0133 seconds
=> ["aamec", "amazon", "college", "prasanna", "table_name"]
hbase:011:0>
```

3) insert data

```
hbase:008:0> put 'aamec','cse','dept:studentname','prasanna'
Took 0.0240 seconds
hbase:009:0> put 'aamec','cse','dept:year','third'
Took 0.0072 seconds
hbase:010:0> put 'aamec','cse','dept:section','A'
Took 0.0342 seconds
hbase:011:0>
```

syntax:

put 'table_name','row_key','column_family:attribute','value'

here row_key is a unique key to retrieve data

code :

this data will enter data into the dept column family

put 'aamec','cse','dept:student name','prasanna'
put'aamec','cse','dept:year','third'put
'aamec','cse','dept:section','A'

This data will enterdata intotheyearcolumn family

put 'aamec','cse','year:joinedyear','2021' put 'aame

```
hbase:026:0> get 'aamec','cse'
COLUMN                  CELL
 dept:section           timestamp=2023-10-13T12:30:57.010, value=B
 dept:studentname       timestamp=2023-10-13T12:13:11.914, value=prasanna
 dept:year              timestamp=2023-10-13T12:13:41.018, value=third
 year:finishingyear     timestamp=2023-10-13T12:16:57.291, value=2025
 year:joinedyear        timestamp=2023-10-13T12:16:41.876, value=2021
1 row(s)
Took 0.0506 seconds
hbase:027:0>
```

4.ScanTable

same as descin RDBMS

syntax: scan
'table_name'

code:
scan 'aamec'

5)To get specific data

syntax:
get'table_name','row_key',[optionalcolumnfamily:attribute]

code :
get'aamec','cse'

6.update table value

```
hbase:025:0> put 'aamec','cse','dept:section','B'
Took 0.0134 seconds
hbase:026:0>
```

The same put command is used to update the table value, if the row key is already present in the database then it will update data according to the value, if not present the it will create new row with the given row key
previously the value for the section in cse is A, But after running this command the value will be changed into B

7)To Delete Data

syntax:delete'table_name','row_key','column_family:attribute'
code:delete 'aamec','cse','year:joinedyear'

```
hbase:011:0> put 'aamec','cse','year:joinedyear','2021'
Took 0.0739 seconds
hbase:012:0> put 'aamec','cse','year:finishingyear','2025'
Took 0.0411 seconds
hbase:013:0>
```

8.DeleteTable first we need to disable the table before dropping it To Disable:
syntax: disable 'table_name' **code:**

disable 'aamec'

```
hbase:029:0> disable 'aamec'
2023-10-13 12:42:08,027 INFO  [main] client.HBaseAdmin (HBaseAdmin.java:rpcCall(926)) - Started
disable of aamec
2023-10-13 12:42:08,699 INFO  [main] client.HBaseAdmin (HBaseAdmin.java:postOperationResult(3591
)) - Operation: DISABLE, Table Name: default:aamec, procId: 106 completed
Took 0.7578 seconds
hbase:030:0>
```

```
hbase:027:0> delete 'aamec','cse','year:joinedyear'
Took 0.0138 seconds
hbase:028:0> get 'aamec','cse'
COLUMN                  CELL
 dept:section           timestamp=2023-10-13T12:30:57.010, value=B
 dept:studentname       timestamp=2023-10-13T12:13:11.914, value=prasanna
 dept:year              timestamp=2023-10-13T12:13:41.018, value=third
 year:finishingyear     timestamp=2023-10-13T12:16:57.291, value=2025
1 row(s)
Took 0.0686 seconds
hbase:029:0>
```

Result:

HBase was successfully installed with an example on Ubuntu 18.04.

EXPT.NO.7	Practice importing andexporting data fromvarious databases.
DATE:	

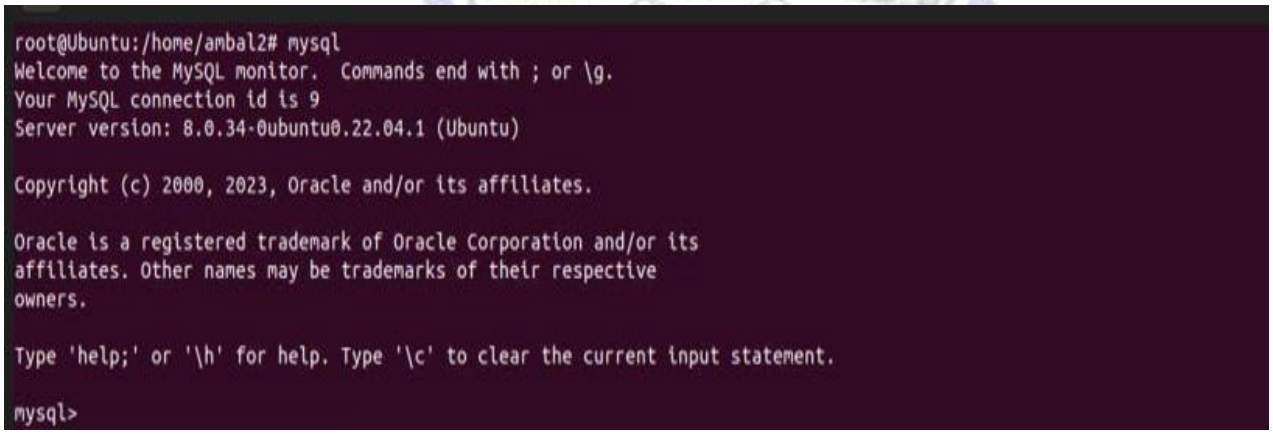
Aim:

To import or export, the order of columnsin My SQL and Hive **Pre-requisite**
Hadoopand Java
MySQL
Hive
SQOOP

Step 1:To start hdfs

Step 2: MySQLInstallation

sudo apt install my sql-server(use this command to install MySQLserver



COMMANDS:

~$sudosu

After this enter your linux user password,then the root mode will be open here we don’t need any authentication for mysql.

~root$mysql

Creating user profiles and grant them permissions:

Mysql> CREATE USER ‘bigd ata'@'localhost' IDENTIFIED

BY‘bigdata’;Mysql>grant all privileges on*.*to big data@localhost;

Note: This step is not required if you just use theroot userto makeCRUD operations in the MySQL

Mysql>CREATEUSER‘bigdata’@’127.0.0.1'IDENTIFIEDBY ‘bigdata’;Mysql>grantallprivilegeson*.*to bigdata@127.0.0.1;

Note: Here, *.* means that the user we create has all the privileges on all the tables of all the databases.

Now, we have created user profiles which will be used to make CRUD operations in the mysql

Step3: Create a database and tableand insertdata.

Example:

Create database Employe;

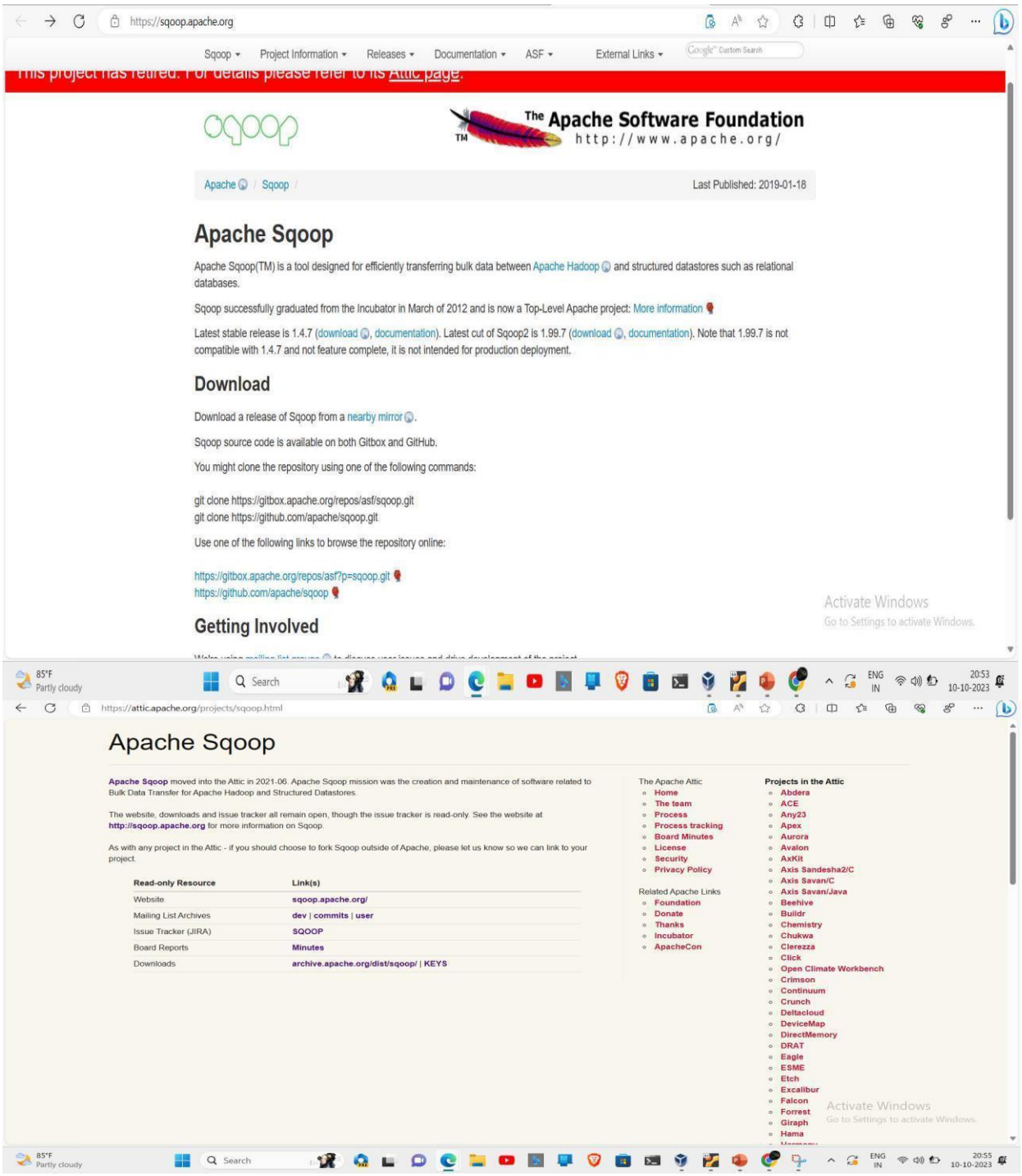
Create table Employe.Emp(author_namevarchar(65),total_no_of_articlesint,phone_noint, address var char(65));

Insert into Emp values(“Rohan”,10,123456789,”Lucknow”);

Step3:Create a database and table in the hive where data should be imported.create table geeks_hive_table(name string, total_articlesint, phone_noint, address string) row format delimited fields terminated by’,’;

```
mysql> insert into dell values('inspiron',3505);
Query OK, 1 row affected (0.12 sec)

mysql> insert into dell values('alienware',5005);
Query OK, 1 row affected (0.03 sec)

mysql> insert into dell values('inspiron',3550);
Query OK, 1 row affected (0.02 sec)
```

Step 4: SQOOPINSTALLATION :



After downloading the sqoop, go to the directory where we downloaded the sqoop and then extract it using the following command:

$tar-xvf sqoop-1.4.4.bin___hadoop-2.0.4-alpha.tar.gz

Thenenter intothe superuser: $su

Nexttomovethattotheusr/libwhich requiresa superuser privilege

$mvsqoop-1.4.4.binhadoop-2.0.4-alpha/usr/lib/sqoop

Then exit :$ exit Goto .bashrc: $sudo nano .bashrc ,

and then add the following export SQOOP_HOME=/usr/lib/sqoop export

PATH=$PATH:$SQOOP_HOME/bin

$ source ~/.bashrc

Thenconfigurethesqoop,gotothedirectoryoftheconfigfolderofsqoop_homeandthenmovethe contents of template file to the environment file.

$ cd $SQOOP_HOME/conf

$ mv sqoop-env-template.sh sqoop-env.sh

Thenopenthesqoop-environmentfileandthenaddthefollowing,

exportHADOOP_COMMON_HOME=/usr/local/Hadoopex

port HADOOP_MAPRED_HOME=/usr/local/hadoop

Note : Here we add the path of the Hadoop libraries and files and it may different from the path which we mentioned here. So, add the Hadoop path based onyour installation.

Step 5:Download and Configure mysql-connector-java:

We can download mysql-connector-java-5.1.30.tar.gz file from thefollowinglink.

Next, to extract thefile and place it to the lib folder
ofsqoop $ tar -zxf mysql-connector-java-5.1.30.tar.gz
$ su
$ cd mysql-connector-java-5.1.30
$ mv mysql-connector-java-5.1.30-bin.jar /usr/lib/sqoop/lib
Note:This is library file is very important don't skip this step because it contains the libraries to connect the mysql databases to jdbc.
Verify sqoop:sqoop-version
Step3:
Hive database Creation
hive>create database sqoop_example;
hive>use sqoop_example;
hive>create table sqoop(usr_namestring,no_opsint,ops_names string);
Hive commands much more a like mysql commands. Here, we just create the structure to store the data which we want to import in hive.

```
ambal2@Ubuntu:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/ambal2/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/ambal2/hadoop-3.2.3/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 1fb24ab2-af10-4d03-948f-73de05944193

Logging initialized using configuration in jar:file:/home/ambal2/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = 63f5f215-bf1c-4eb8-a6b5-01338cc55110
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or us
hive>
```

```
hive> show databases;
OK
default
sqoop
Time taken: 0.683 seconds, Fetched: 2 row(s)
hive> use sqoop;
OK
Time taken: 0.08 seconds
hive> show tables;
OK
bigdata
sqoop
Time taken: 0.148 seconds, Fetched: 2 row(s)
hive> create table dell(mdl_name string,mdl_num int);
OK
Time taken: 2.564 seconds
hive>
```

Step 6:Importing data from MySQL to hive :

Sqoop import connect\jdbc:mysql://127.0.0.1:3306/database_name_in_mysql\
--usernameroot--password cloudera\ --
tabletable_name_in_mysql\
--hive-import--hive-tabledatabase_name_in_hive.table_name_in_hive\--m 1

```
Activities  Terminal                                   Oct 12 12:06
                                                      ambal2@Ubuntu: ~
ambal2@Ubuntu: $ sqoop import --connect jdbc:mysql://127.0.0.1:3306/sqoop_example --username dell --password dell --table dell --hive-import --hive-table sqoop.dell --m 1
```

```
        Job Counters
                Launched map tasks=1
                Other local map tasks=1
                Total time spent by all maps in occupied slots (ms)=8912
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=8912
                Total vcore-milliseconds taken by all map tasks=8912
                Total megabyte-milliseconds taken by all map tasks=9125888
        Map-Reduce Framework
                Map input records=3
                Map output records=3
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=141
                CPU time spent (ms)=2800
                Physical memory (bytes) snapshot=223498240
                Virtual memory (bytes) snapshot=2542714880
                Total committed heap usage (bytes)=136839168
                Peak Map Physical memory (bytes)=223498240
                Peak Map Virtual memory (bytes)=2542714880
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=43
2023-10-12 12:15:07,009 INFO mapreduce.ImportJobBase: Transferred 43 bytes in 31.8907 seconds (1.3484 bytes/sec)
2023-10-12 12:15:07,047 INFO mapreduce.ImportJobBase: Retrieved 3 records.
Thu Oct 12 12:15:07 IST 2023 WARN: Establishing SSL connection without server's identity verification is not recommended. According
ished by default if explicit option isn't set. For compliance with existing applications not using SSL the verifyServerCertificate
 useSSL=false, or set useSSL=true and provide truststore for server certificate verification.
2023-10-12 12:15:07,188 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `dell` AS t LIMIT 1
2023-10-12 12:15:07,282 INFO hive.HiveImport: Loading uploaded data into Hive
2023-10-12 12:15:09,556 INFO hive.HiveImport: SLF4J: Class path contains multiple SLF4J bindings.
2023-10-12 12:15:09,557 INFO hive.HiveImport: SLF4J: Found binding in [jar:file:/home/ambal2/apache-hive-3.1.2-bin/lib/log4j-slf4j-
2023-10-12 12:15:09,557 INFO hive.HiveImport: SLF4J: Found binding in [jar:file:/home/ambal2/hadoop-3.2.3/share/hadoop/common/lib/s
2023-10-12 12:15:09,557 INFO hive.HiveImport: SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2023-10-12 12:15:09,562 INFO hive.HiveImport: SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
```

OUTPUT:

```
ambal2@Ubuntu:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/ambal2/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/ambal2/hadoop-3.2.3/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = ca95a42a-a85e-4d00-948a-c435099df78f

Logging initialized using configuration in jar:file:/home/ambal2/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = a1776f23-c763-4313-a2c9-e3bc02cb423e
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive
hive> show databases;
OK
default
sqoop
Time taken: 0.573 seconds, Fetched: 2 row(s)
hive> use sqoop;
OK
Time taken: 0.073 seconds
hive> select * from dell;
OK
inspiron        3505
alienware       5005
inspiron        3550
Time taken: 3.087 seconds, Fetched: 3 row(s)
hive>
```

Result:

Thus the import and export, the order of columns in MySQL queries are exported to hive successfully

CONTENT BEYOND SYLLABUS

EXPT.NO.8	**Installation and creation of database and collection CRUD document** : Insert,Query,Update and Delete Document
DATE:	

AIM:

ToInstallandCreatedatabaseandCollectionCRUDDocument:Insert,Query,Updateand Delete Document.

PROCEDURE:

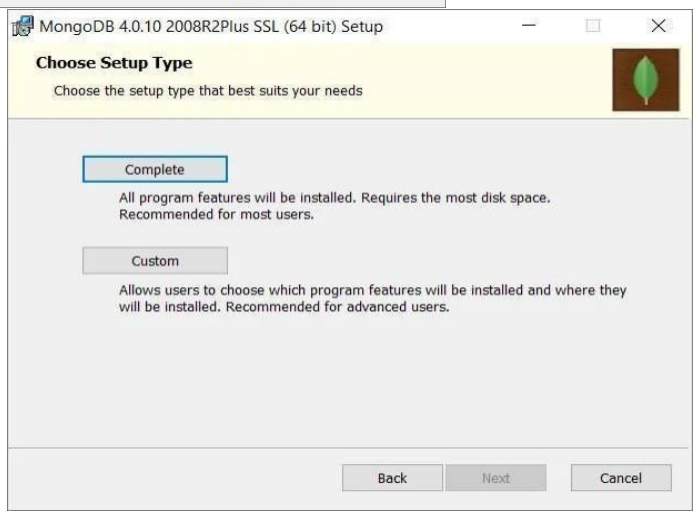
InstallMongo DBinWindows

The websiteof MongoDBprovides all the installation instructions and MongoDB is supported byWindows, Linux as well as Mac OS.

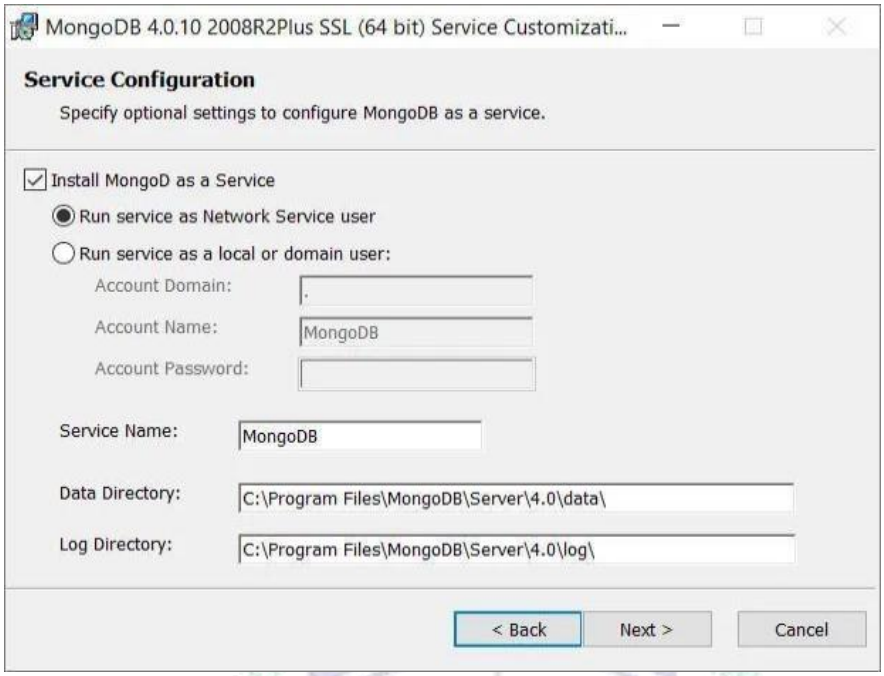
Itistobenotedthat,MongoDBwillnotrunin WindowsXP;soyouneed to installhigher versionsofwindows touse this database.

Onceyou visit thelink(http://www.mongodb.org/downloads), click the downloadbutton.

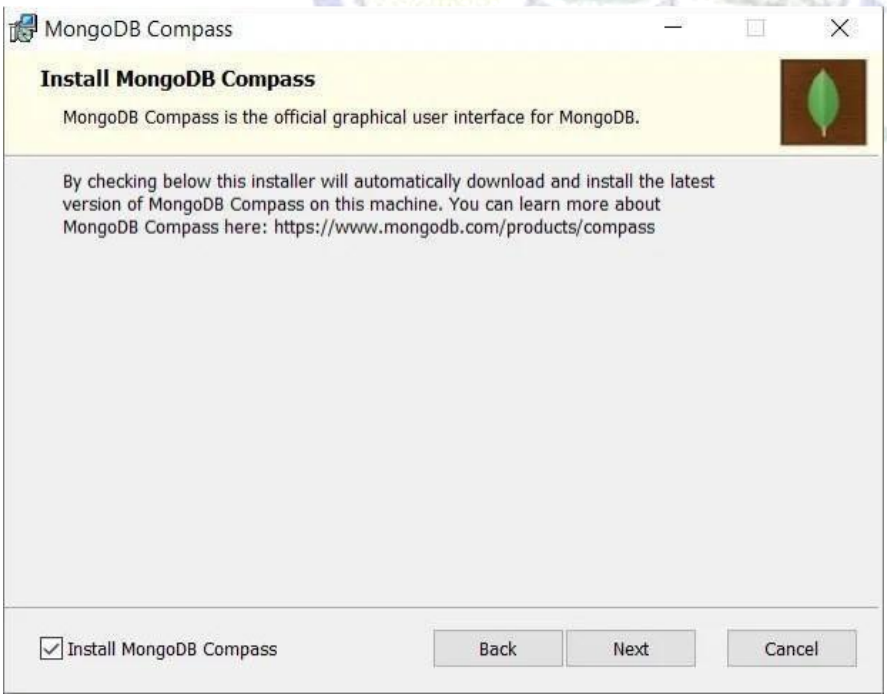
1. Once the download is complete, double click this setup file to install it.
2. Follow the step:



4.Then, select the radio button "Run services as Network service user."



5. The setup system will also prompt you to install Mongo DB Compass, which is Mongo DB official graphical user interface (GUI).You can tick the checkbox to install that as well.



Once the installation is done completely, you need to start Mongo DB and to do so follow the process:

1. Open Command Prompt.
2. Type: C:\Program Files\MongoDB\Server\4.0\bin
3. Now type the command simply: **mongodb** to run the server.

In this way, you can start your MongoDB database.Now, for running Mongo DB primary client system, you haveto use the command:

C:\Program Files\MongoDB\Server\4.0\bin>**mongo.exe**

Result

Downloading and Installation and Creationofdatabase and Collection CRUD Document:Insert, Query,Update andDelete Document has been successfully completed.

EXPT.NO.9	Create-file, data in memory, other RDD. Lazy Execution, Persisti RDD , RDD, Actions and Transformationon RDD
DATE:	

AIM:

To Create-file, data in memory, other RDD.Lazy Execution, Persisti RDD, RDD, Actions and Transformation on RDD

Procedure:

1. Spark create RDD fromSeq or List (using Parallelize)
2. Creating an RDD from atext file

Creating from 3 another RDD

4. Creating from existing DataFrames andDataSet

Spark Create RDDfrom Seqor List (using Parallelize)

RDD'saregenerallycreatedbyparallelizedcollectioni.e.bytaking anexistingcollectionfromdriverprogram(scala,pythone.t.c)and passingittoSparkContext'sparallelize()method.Thismethodisused onlyfortestingbutnotinrealtimeastheentire datawillresideonone node whichis notidealfor production.

```
valrdd=spark.sparkContext.parallelize(Seq(("Java", 20000),
 ("Python", 100000), ("Scala", 3000)))
rdd.foreach(println)Outputs:
(Python,100000)(Scala,3000)
(Java,20000)
```

.

Mostlyforproductionsystems,wecreateRDD’sfromfiles.herewillseehow to create anRDDbyreading data froma file.

valrdd=spark.sparkContext.textFile("/path/textFile.txt")

This createsan RDD for which eachrecord represents aline in a file.

IfyouwanttoreadtheentirecontentofafileasasinglerecordusewholeTextFiles() methodonsparkContext.

val rdd2 = spark.sparkContext.wholeTextFiles("/path/textFile.txt")
rdd2.foreach(record=>println("FileName:"+record._1+",FileContents :"+record._2))

Inthiscase,eachtext fileisasinglerecord. Inthis,thenameofthefileisthe first column and the value of the text file isthe secondcolumn.

CreatingfromanotherRDD

Youcanusetransformations likemap,flatmap,filtertocreateanew RDDfrom an existing one. val rdd3 = rdd.map(row=>{(row._1,row._2+100)})

Above, creates a newRDD “rdd3” by adding 100 to eachrecord onRDD. thisexample outputs below.

(Python,100100)

(Scala,3100)

(Java,20100)

From existing DataFramesandDataSet

ToconvertDataSetorDataFrametoRDDjustuserdd()methodon anyofthese data types.

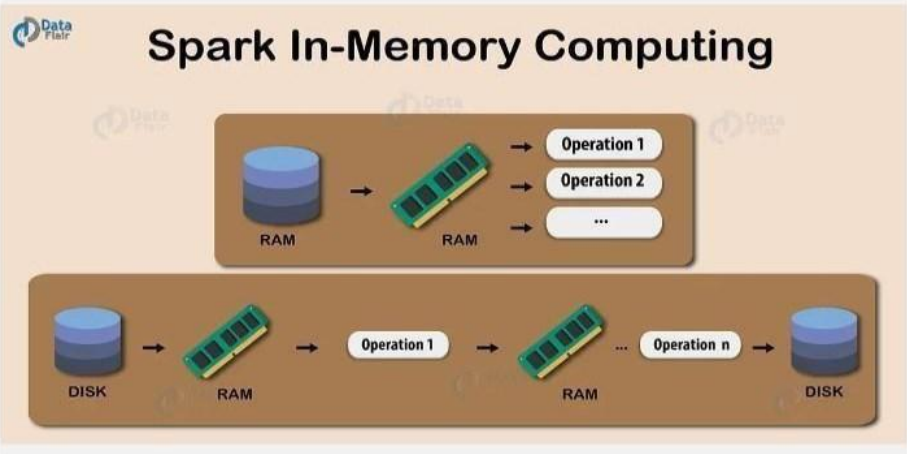
val myRdd2 = spark.range(20).toDF().rdd

toDF() creates a DataFrameand bycallingrdd on DataFrame returns backRDD.

SPARK IN-MEMORYCOMPUTING

Thedataiskeptinrandomaccessmemory(RAM)insteadofsomeslow diskdrivesandisprocessedinparallel.Usingthiswecandetectapattern, analyzelargedata.Thishasbecomepopularbecauseitreducesthecostof memory.So,in-memoryprocessingiseconomicforapplications.Thetwo maincolumnsofin-memorycomputationare-Paralleldistributed processing.

.



SparkIn-Memory Computing

Keepingthedatain-memoryimprovestheperformancebyanorderof magnitudes.ThemainabstractionofSparkisitsRDDs.AndtheRDDs arecached using the cache() or persist() method.

Whenweusecache()method,alltheRDDstoresin-memory.WhenRDD storesthevalueinmemory,thedatathatdoesnotfitinmemoryiseither recalculatedortheexcessdataissenttodisk.WheneverwewantRDD,it canbeextractedwithoutgoingtodisk.Thisreducesthespace-time complexityand overhead of diskstorage.

Thein-memorycapabilityofSparkisgoodformachinelearningand micro-batchprocessing. It provides faster execution for iterative jobs.

Whenweusepersist()methodtheRDDscanalsobestoredin-memory, wecanuseitacrossparalleloperations.Thedifferencebetweencache() andpersist()isthatusingcache()thedefaultstoragelevelis MEMORY_ONLY whileusingpersist()wecanusevariousstoragelevels.

Storage levels of RDDPersist() inSpark

The variousstoragelevel of persist() method inApache Spark RDD are:

❖MEMORY_ONLY

❖MEMORY_AND_DISK

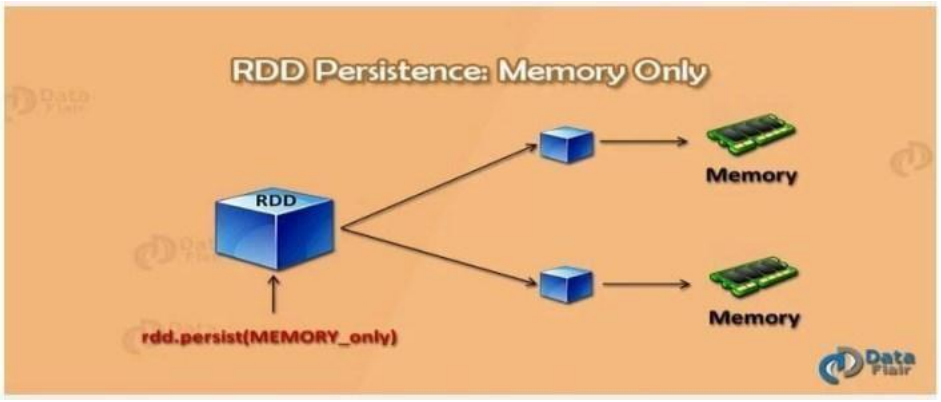
❖MEMORY_ONLY_SER

❖MEMORY_AND_DISK_SER

❖DISK_ONLY

❖MEMORY_ONLY_2and MEMORY_AND_DISK_2

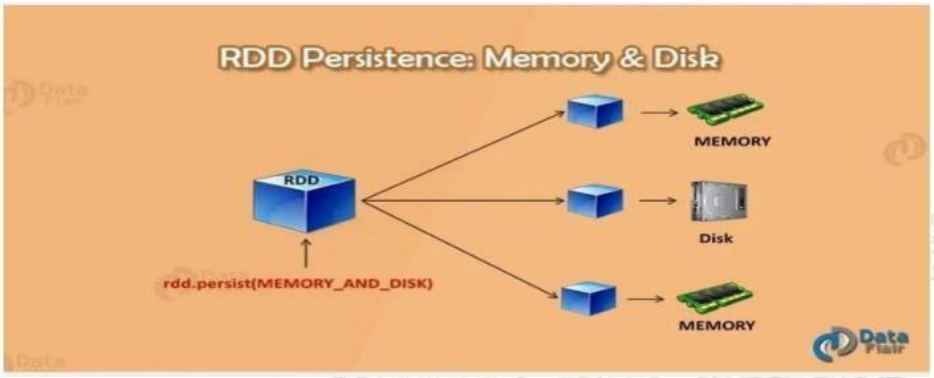
.



Sparkstoragelevel –memory only

InthisstoragelevelSpark,RDDstoreasdeserializedJAVAobjectin JVM.IfRDDdoesnotfitinmemory,thentheremainingwillre-compute each timethey are needed.

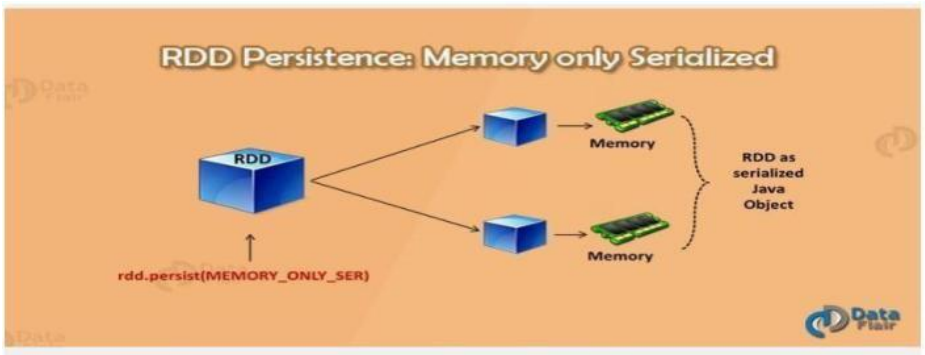
MEMORY AND DISK



Sparkstoragelevel-memory and disk

Inthislevel,RDDisstoredasdeserializedJAVAobjectinJVM.Ifthe fullRDDdoesnotfitinmemorythentheremainingpartitionisstoredon disk,instead of recomputing it every timewhen it is needed.

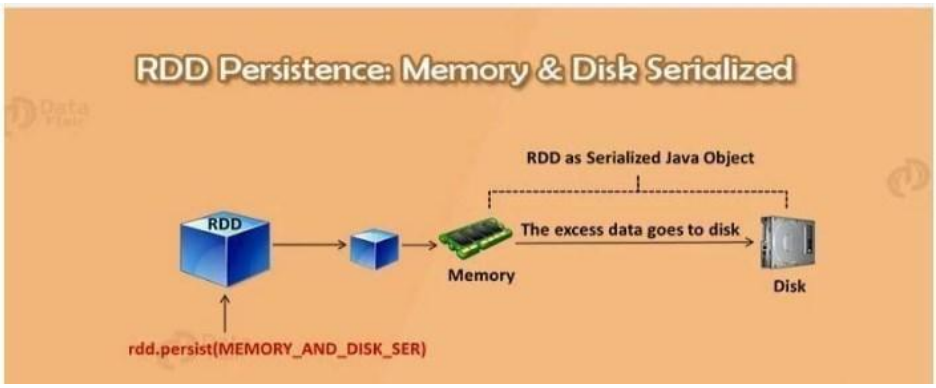
MEMORYONLYSER



.

especially when we use fast serializer.

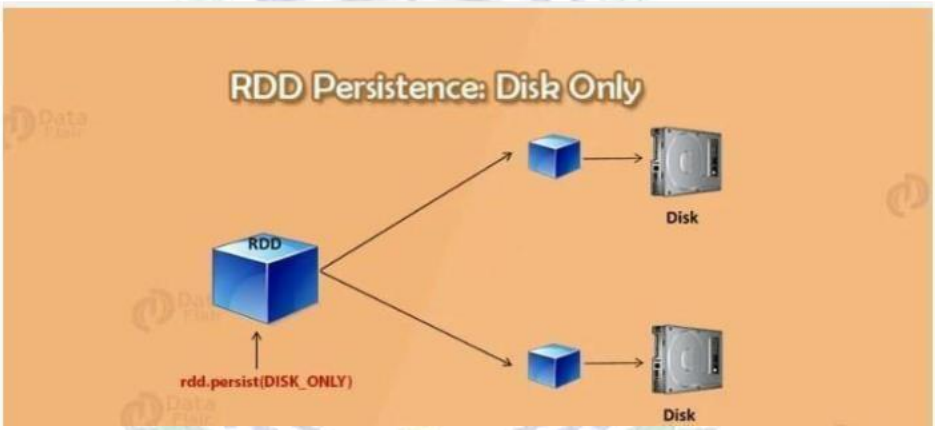
MEMORY AND DISK SER



Sparkstoragelevel –memory and disk serialized

ThislevelstoresRDDasserializedJAVA object.IfthefullRDDdoesnot fitinthememorythenitstorestheremainingpartitiononthedisk,instead of recomputing it everytime whenwe need.

DISK_ONLY



Spark storage level-disk-only

This soragelevel stores the RDD partitions only on disk.

MEMORYONLY2 and MEMORY AND DISK 2

ItislikeMEMORY_ONLYandMEMORY_AND_DISK.Theonlydifference is that each partition gets replicate on two nodes in the cluster.

Advantages of In-memory Processing

Afte.rstudyingSparkin-memorycomputingintroductionandvariousstoragelevelsin detail, let'sdiscusstheadvantagesof in-memory computation

1. an retrieveiteasily.
2. It is good for real-time risk management andfraud detection.
3. The data becomes highlyaccessible.
4. The computation speed of the system increases.
5. Improves complex eventprocessing.
6. Cached alarge amountof data.
7. It is economic, as the cost of RAM has fallen overa period of time

Result:

ThustoCreate-file, datainmemory, other RDD. LazyExecution,PersistiRDD ,RDD,ActionsandTransformation on RDD installed Hadoop has been verifiedsucessfully

EXPT.NO.10	Create Stories usingTableau,Connect to data,Build Charts andAnalyze Data, Create Dashboard.
DATE:	

AIM:

ToCreateStoriesusingTableau,Connecttodata,BuildChartsandAnalyzeData,Create Dashboard.

PROCEDURE:

WhatisCustomSQLQuery?

TableauqueryeachdatasourceusingSQLthat'sspecifictothedatatype.Tableauallows theSQL usedtoqueryadatasourcetobecustomizedinordertomanipulatethejoins,filters, and fieldlengths and types producea more accurate output.

Sample SQLquery:

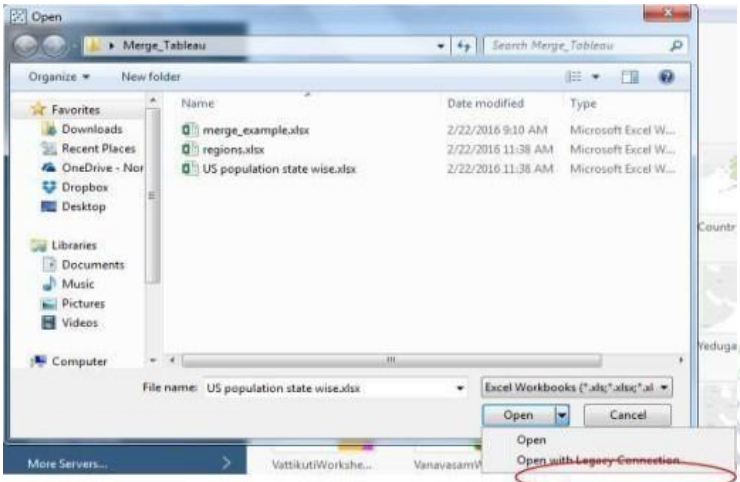
```
SELECT['usstates data 1$'].[State]AS [State], ['us states data
1$'].[Population]AS [Population],['usstates data 1$'].[Region]AS
[Region]
FROM ['us states data1$']UNION
SELECT['usstates data 2$'].[State]AS [State], ['us states data
2$'].[Population]AS [Population],['usstates data 2$'].[Region]AS
[Region]
FROM ['us states data2$']UNION
SELECT['usstates data 3$'].[State]AS [State], ['us states data
3$'].[Population]AS [Population],

    UNION
    SELECT['usstates data 4$'].[State]AS
    [State], ['us states data4$'].[Population]AS
    [Population],['us state
    [Region]
    FROM ['us states data
```

Steps to visualizeusin

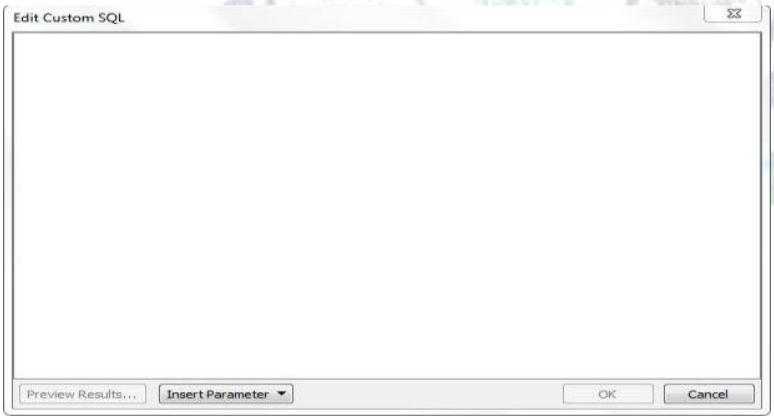
1. Thefollowingexcelworkbookconsistsof4sheetsconsistingofUSpopulationstatewise.

USpopulation state wise.xlsx

2. OpenTableau andconnect the Excel data as Openwith Legacy Connection.



3. NowyourTableauwindowlookslikethiswith anadditionalsheetnamedas New Custom SQL.



4. Now weneed to write SQLquery to perform union operation on the excel sheets and generatea combined sheet.

Ex:

SELECT[Sheet1$].[ID]AS [ID],

[Sheet1$].[Type]AS

[Type], [Sheet1$].[Value]

AS [Value]FROM

[Sheet1$]

UNIONALL

SELECT[Sheet2$].[ID]AS [ID],

[Sheet2$].[Type]AS

[Type], [Sheet2$].[Value]

AS [Value]FROM

[Sheet2$]

5. TheaboveSQLqueryperformsunionoperationontwo sheetsnamelySheet1andSheet2withID,Type,andValueas theircolumns.

6. AfterwritingthequeryclickonPreviewResults ontheEditCustom SQL window pane to check whether the query workedfine or not.

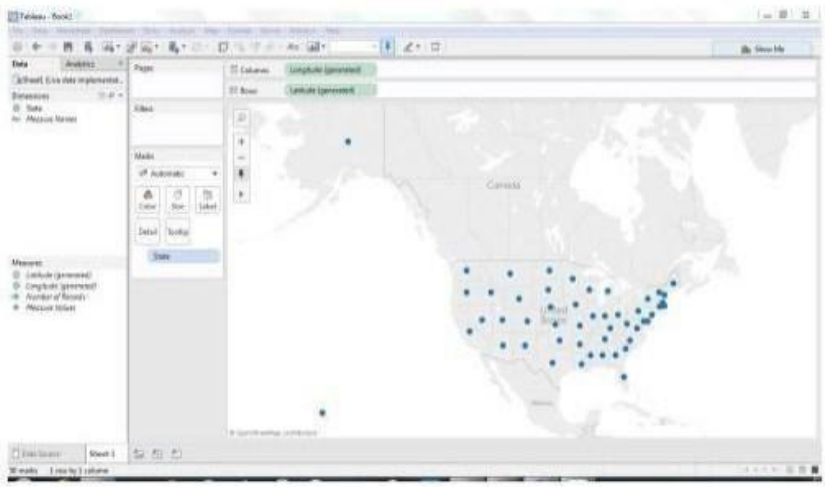
7. The resultingdatashouldcontainall thestates,populationandregionname to which it belong to.

8. Click on theTableau sheet to visualize the data.

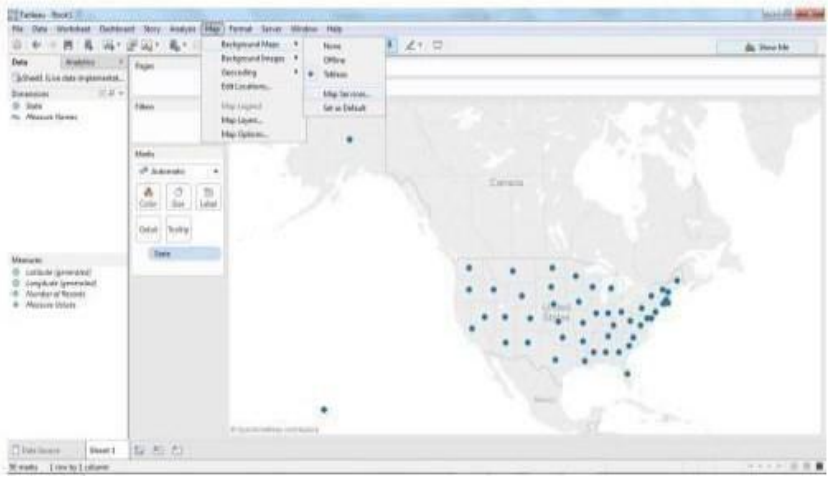
9. NowyouwillseeState,RegionasDimensionsandPopulationas Measures.

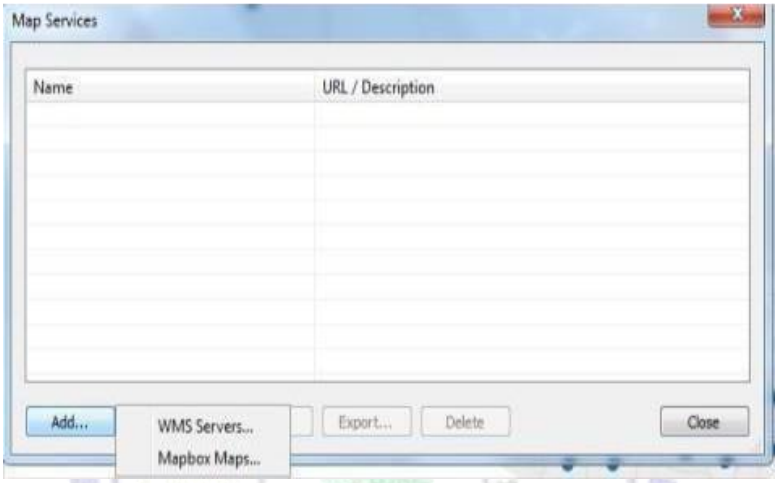
LiveDataorReal-timedata(RTD)denotesinformationthatis deliveredimmediatelyaftercollection.Thereisnodelayinthe timelinessoftheinformationprovided.Real-timedataisoftenused fornavigationortracking.AWebMapService(WMS)isastandard protocolforservinggeoreferencedmapimageswhichamapserver generatesusingdatafromaGISdatabase.A WebMapService(WMS) producesanimage(e.g.GIF,JPG)ofgeospatialdata.Stepsto implementlive map usingTableau:

1. OpenTableau andconnect the Excel data.
2. DraganddropstateDimensioninDetailmenuprovidedinMarkswindow pane.
3. NowyoucanseeSymbolmaprepresentingthestatesofUnitedStates.It looks as shown below

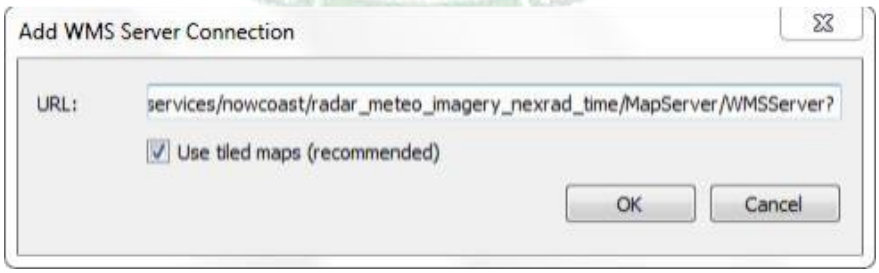


4. In this we willbe using OGCWMSserver to get the live geospatial data.
5. ClickonMapiteminthemenubar,thenselectBackgroundMapsandselect WMS Server as shown

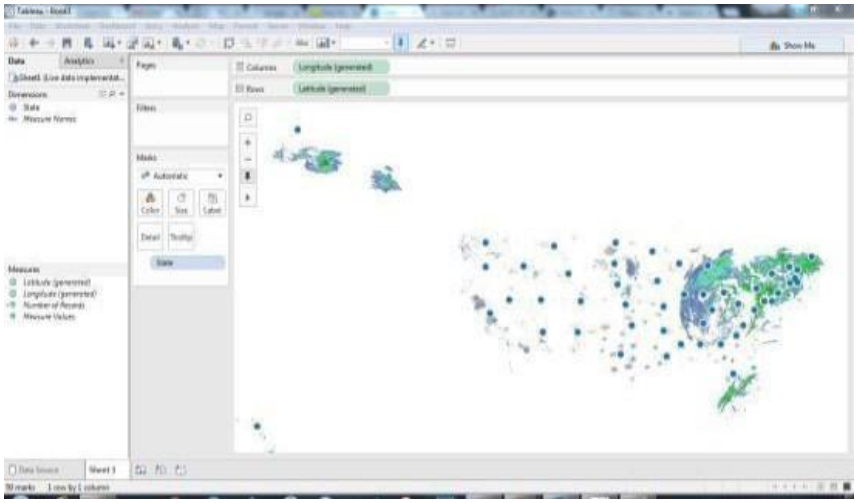




7. ClickonAddandselectWMSServers.AfterclickingonWMSServersa window pane as shown below willbe prompted to enter the URL of the server. http://nowcoast.noaa.gov/arcgis/services/nowcoast/radar_meteo_imagery_nexrad_time/MapServer/WMSServer?

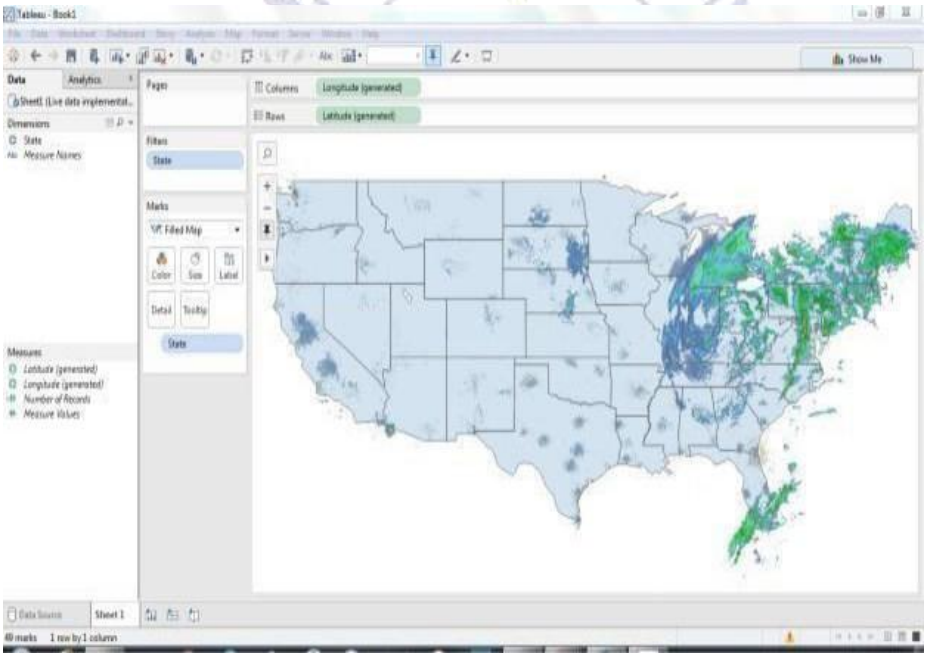


8. NowclickonOKandthencloseWMSserverwindowpaneandamap can beseen as shown



10.Changethecolortransparencytolessthan20%sothattherequiredmap is visible.

11.Thenrequiredmaplookslikethis



Result:

ToCreateStoriesusingTableau,Connecttodata,BuildChartsandAnalyzeData,Create Dashboardhasbeenverifiedsuccessfully.

VIVA QUESTIONS

VIVAQUESTIONS ANDANSWERS

1.What is the port number for Name Node ,TaskTracker and JobTracker?

- NameNode 50070
- JobTracker 50030
- TaskTracker 50060

2.Explainaboutthe processof interclusterdata copying.

HDFSprovides a distributed data copying facility through theDistCPfrom sourceto destination. Ifthisdata copyingis withinthe hadoop cluster then itis referred to asintercluster data copying. DistCPrequires both sourceanddestination tohavea compatible or sameversion ofhadoop.

3.How can you overwritethereplicationfactors inHDFS?

Thereplication factorin HDFS can bemodifiedor overwrittenin 2 ways-

1)Using the Hadoop FS Shell,replication factorcanbechangedperfile basisusing thebelowcommand-$hadoop fs –setrep–w 2/my/test_file (test_fileis thefilename whose replicationfactor will besetto 2)

2)Using theHadoop FS Shell, replication factor of all files under a given directory can be modifiedusing the below command-

3)$hadoopfs–setrep–w5/my/test_dir(test_diristhe name ofthedirectoryandallthefilesinthisdirectorywill have a replication factor setto 5)

4.Explain thedifferencebetween NAS andHDFS.

NAS runs on asingle machineand thus thereis no probability of data redundancy whereas HDFS runs on a cluster of different machines thus there is data redundancy because ofthereplication protocol.NAS stores dataona dedicated hardwarewhereas in HDFS allthe datablocks aredistributedacross localdrives of the machines.

InNASdataisstoredindependentofthecomputationandhenceHadoopMapReducecannotbeusedfor processingwhereasHDFSworkswithHadoopMapReduceasthecomputationsinHDFSaremovedtodata.

5.Explain what happens if during the PUToperation, HDFS block is assigneda replication factor1 instead of the default value

ReplicationfactorisapropertyofHDFSthatcanbesetaccordinglyfortheentireclustertoadjustthenumberof timestheblocksaretobereplicatedtoensurehighdataavailability.ForeveryblockthatisstoredinHDFS,the clusterwillhaven-1duplicatedblocks.So,ifthereplicationfactorduringthe PUToperationissetto1insteadofthe defaultvalue3,thenitwillhaveasinglecopyofdata.Underthesecircumstanceswhenthereplicationfactorissetto 1 ,iftheDataNodecrashes underany circumstances, thenonly single copy of thedatawould belost.

6.What is the process to change thefiles at arbitrary locations in HDFS?

HDFSdoesnotsupportmodificationsatarbitraryoffsetsinthefileormultiplewritersbutfilesarewrittenbya singlewriterin append only formati.e. writes toafile in HDFSare always made attheend ofthefile.

7.Explain about theindexing process in HDFS.

Indexingprocessin HDFSdependsontheblock size.HDFS storesthelastpartofthedatathatfurtherpointstothe address where thenextpart of data chunk is stored.

8.What is a rack awareness and onwhatbasis is datastored in a rack?

Allthedata nodes puttogetherforma storage area i.e. thephysicallocationofthedata nodes is referredto asRack inHDFS.Therackinformationi.e.therackidofeachdatanodeisacquiredbytheNameNode.Theprocessof selectingcloser datanodesdepending on the rack informationis knownas RackAwareness.

Thecontentspresentinthefileareddividedintodatablockassoonastheclientisreadytoloadthefileintothe hadoopcluster.After consulting with the NameNode, clientallocates 3 data nodes for eachdata block. For eachdatablock,thereexists2copiesinonerackandthethirdcopyispresentinanotherrack.Thisisgenerally referred toas theReplicaPlacementPolicy.

9. What happensto a NameNode that has no data?

There does notexist anyNameNode without data. If itis aNameNodethenitshould have somesort ofdata in it.**10.**

What happenswhen a user submits a Hadoop job when the NameNodeisdown-does the job get in to hold ordoes it fail.

TheHadoop job fails when theNameNodeis down.

11. What happenswhen a usersubmits a Hadoop job when theJobTrackeris down- does the job get in to hold ordoes it fail.

TheHadoop job fails when theJobTrackeris down.

12. Wheneveraclient submits a hadoop job, who receivesit?

NameNodereceivestheHadoopjobwhichthenlooksforthedatarequestedbytheclientandprovidestheblock information. JobTrackertakes care ofresourceallocation ofthe hadoopjob toensuretimely completion.

13. What is a heartbeatin HDFS?

HeartbeatsinHDFSarethesignalsthataresentbyDataNodestotheNameNodetoindicatethatitis functioning properly (alive). By default, theheartbeatintervalis 3 seconds, which can be configured using dfs.heartbeat.intervalinhdfs-site.xml.

14. How would you check whetheryourNameNodeisworking ornot?

Therearemanywaystocheckthestatusoftheb NameNode.

Mostcommonly,oneusesthejpscommandtocheckthestatusofallthedaemonsrunningintheHDFS.Alternatively,oneca nvisittheNameNode'sWebUIforthe same.

15. What is a block?

Youshouldbegintheanswerwithageneraldefinitionofablock.Then,youshouldexplaininbriefabouttheblocks presentin HDFS and alsomention their default size.

Blocksarethesmallestcontinuouslocationonyourharddrivewheredataisstored.HDFSstoreseach fileasblocks, and distributeitacrossthe Hadoop cluster.Thedefault size ofa block in HDFS is128 MB (Hadoop 2.x)and64MB(Hadoop1.x)whichismuchlargerascomparedtotheLinuxsystemwheretheblocksizeis4KB.

Thereasonofhavingthishugeblocksizeistominimizethecostofseekandreducethemetadatainformation generated per block.

16. Suppose thereis file of size514MB storedin HDFS (Hadoop 2.x) using default block size configurationanddefault replication factor.Then, how many blocks will be created in total and what will be the sizeofeach block?

DefaultblocksizeinHadoop2.xis128MB.So,afileofsize514MBwillbedividedinto5blocks(514MB/128 MB)wherethefirstfourblockswillbeof128MBandthelastblockwillbeof2MBonly.Since,we areusingthe

defaultreplication factori.e.3,eachblockwillbereplicatedthrice.Therefore,we willhave15blocksintotalwhere 12 blocks will beof size128 MB each and 3blocksof size 2 MBeach.

17. How to copy a fileinto HDFS with a different block sizetothat ofexisting block size configuration?

♣Tip:You should starttheanswer withthecommand for changingthe block size and then, you should explainthe wholeprocedurewithan example.Thisis how you should answerthis question:

Yes,onecancopyafileintoHDFSwithadifferentblocksizebyusing'-Ddfs.blocksize=block_size'wherethe block_sizeis specifiedinBytes.

Letme explainit with anexample: Suppose, I wantto copya filecalledtest.txt of size, sayof 120 MB,into the HDFSand I wantthe block sizeforthis fileto be32MB (33554432 Bytes) insteadof the default(128

MB). So, I would issuethe following command: hadoop fs -Ddfs.blocksize=33554432 -copyFromLocal /home/edureka/test.txt/sample_hdfs

Now, I cancheck theHDFS block size associatedwith this fileby:hadoop fs -stat %o/sample_hdfs/test.txt

Else, Icanalso usethe NameNodewebUI for seeing the HDFSdirectory.

♣Tip:YoucangothroughtheblogonHadoopShellCommandswhereyouwillfindvariousHadoopcommands, explainedwithan example.

18.Can you change theblock size of HDFS files?

Yes, I can changetheblock sizeof HDFS files by changingthedefault sizeparameterpresentin hdfs-site.xml .But, Iwill have to restartthecluster forthis propertychangetotake effect.

19.What is JobTracker?

JobTrackerisaHadoopserviceusedfortheprocessingofMapReducejobsinthecluster.Itsubmitsandtracks the jobs to specificnodes havingdata.OnlyoneJobTrackerruns on singleHadoopcluster onits ownJVM process. if JobTracker goes down,allthejobs halt.

14. Explainjob scheduling through JobTracker.

JobTrackercommunicateswithNameNodetoidentifydatalocationandsubmitstheworktoTaskTrackernode. TheTaskTrackerplaysamajorroleasitnotifiestheJobTrackerforanyjobfailure.Itactuallyisreferredto theheartbeatreporterreassuring the JobTrackerthatitis stillalive.Later,theJobTrackerisresponsiblefor the actions as initmayeitherresubmitthe job ormarkaspecific recordas unreliable or blacklisti

.

20.What is SequenceFileInputFormat?

AcompressedbinaryoutputfileformattoreadinsequencefilesandextendstheFileInputFormat.Itpassesdata betweenoutput-input(betweenoutputofoneMapReducejobtoinputofanotherMapReducejob)phasesof MapReduce jobs.

21.How to setmappersandreducers forHadoop jobs?

Users can configureJobConf variable to set numberof mappers and reducers.

```
1  job.setNumMaptasks()
2  job.setNumreduceTasks()
```

22. **Explain JobConfin MapReduce.**

Itisaprimaryinterfacetodefineamap-reducejobintheHadoopforjobexecution.JobConfspecifiesmapper,

Combiner,partitioner,Reducer,InputFormat,OutputFormatimplementationsandotheradvancedjobfaetsliek Comparators.

Thesearedes

23. **What is a MapReduce Combiner?**

Alsoknownas semi-reducer,Combinerisanoptionalclass tocombinethemapoutrecordsusingthesamekey. Themainfunctionofacombineristoacceptinputsfrom MapClassandpassthosekey-valuepairstoReducer class

24. **What is RecordReaderina Map Reduce?**

RecordReaderisusedtoreadkey/valuepairsformtheInputSplitbyconvertingthebyte-orientedviewand presenting record-orientedview to Mapper.

25. **DefineWritable data types inMapReduce.**

Hadoopreadsandwritesdatainaserializedforminwritableinterface.TheWritableinterfacehasseveralclasses likeText (storing String data), IntWritable,LongWriatble, FloatWritable, BooleanWritable. users arefreeto define their personalWritableclasses as well.

26. **What arethecomponents used in Hive QueryProcessor?**

Following arethe components of aHiveQuery Processor:

Parseand Semantic Analysis(ql/parse)

MetadataLayer (ql/metadata)

TypeInterfaces (ql/typeinfo)

Sessions (ql/session)

Map/ReduceExecution Engine(ql/exec)

Plan Components (ql/plan)

HiveFunctionFramework(ql/udf)

Tools (ql/tools)

Optimizer(ql/optimizer)

27. **What areBuckets inHive?**

Bucketsin HiveareusedinsegregatingHivetable dataintomultiplefilesordirectories.Theyareusedforefficient querying.

28. **What kind of data warehouseapplication is suitableforHive?What arethetypes of tables inHive?**Hive is not considered afulldatabase.Thedesign rules and regulations of Hadoopand HDFS haveput restrictions on whatHive cando. However, Hive ismost suitable for data warehouseapplications becauseit:Analyzes relatively staticdata

Has lessresponsivetime

Does not makerapidchanges in data

AlthoughHivedoesn’tprovidefundamentalfeaturesrequiredforOnlineTransactionProcessing(OLTP),itis suitable for data warehouseapplicationsin large datasets.There are two typesof tables in Hive:Managed tables Externaltables

29. **What is the definition ofHive?Whatis the present version of Hive? Explain ACIDtransactionsin Hive.**

Hiveis anopen-source data warehouse system.Wecan use Hive for analyzingand querying large datasets. It'ssimilartoSQL.ThepresentversionofHiveis0.13.1.Hivesupports ACID(Atomicity,Consistency, Isolation,andDurability)transactions.ACIDtransactionsareprovidedatrowlevels.Followingarethe options Hiveuses to supportACID transactions:

Insert

Delete

Update

30. **What is the maximum size of a string data typesupported by Hive? Explain how Hivesupports binary formats.**

ThemaximumsizeofastringdatatypesupportedbyHiveis2GB.Hivesupportsthetextfileformatbydefault, and italso supportsthebinary format sequencefiles, ORC files,Avro datafiles, and Parquet files.

Sequencefile:Itisa splittable,compressible,androw-oriented file with a generalbinary format.

ORCfile:Optimizedrowcolumnar(ORC)formatfileisarecord-columnarandcolumn-orientedstoragefile.Itdivides the tablein rowsplit.Eachsplitstores the valueof the first rowinthefirstcolumnandfollows subsequently. Avrodatafile:Itisthesameasasequencefilethatissplittable,compressible,androw-orientedbutwithoutthe support ofschemaevolution and multilingualbinding.

Parquetfile:InParquetformat,alongwithstoringrowsofdataadjacenttooneanother,wecanalsostorecolumn valuesadjacenttoeach other such thatboth horizontally and verticallydatasetsare partitioned.

31. **Give thenameof the keycomponents of HBase**

Thekey componentsof HBase are Zookeeper, RegionServer, Region, CatalogTables and HBase Master.

32. **What is S3?**

S3 stands for simple storageservice and itis aoneof thefilesystemused by hbase.

33. **What is the use of get()method?**

get()method is usedto read thedatafrom thetable.

34. **What is the reason of using HBase?**

HBaseisusedbecauseitprovidesrandomreadandwriteoperationsanditcanperformanumberofoperationper second ona large datasets.

35. **In how many modes HBase can run?**

There are two runmodesof HBase i.e. standaloneand distributed.

36. **Definethe differencebetween hiveand HBase?**

HBase is used to support recordlevel operations buthivedoes not supportrecord leveloperations.

CompareHive and HBase: which one is best for your needs?Learn morein ourHivevs Hbaseblog now!

37. **Definecolumnfamilies?**

Itis acollection ofcolumns whereas row is acollection ofcolumn families.

38. **Definestandalonemodein HBase?**

ItisadefaultmodeofHBase.In standalonemode,HBasedoesnotuseHDFS—itusesthe localfilesysteminstead— and itruns all HBase daemons and a localZooKeeperinthesameJVMprocess.

39. **What is decorating Filters?**

Itis usefulto modify, or extend,the behavior ofa filter togainadditionalcontrol overthe returned data.

40. **What is the full form ofYCSB?**

YCSB stands forYahoo!Cloud Serving
Benchmark.

41. **What is the use ofYCSB?**

Itcanbeusedto runcomparable workloadsagainstdifferent storagesystems.

42. **What is the HiveObjectInspectorfunction?**

TheHiveObjectInspectorfunctionisanimportantfeatureoftheApacheHivedatawarehousesoftwareusedto examineandmodifytheinternaldatastructuresoftables.Itisavitalelementintheprocessofexecuting Hive queries,asittransformstherawdatasavedinHivetablesintoaformatthatHiveoperatorscanhandle.The ObjectInspectorfunctionisresponsibleforcreatingastandardinterfaceforHive operatorstoaccesstabledata,and itsupportsmultipledatatypes,includinguser-defined,complex,andprimitivetypes.Hiveoperatorsextensively use this functionin HiveQLqueries, and itis avaluabletool formanipulatingdatawithinHive.