

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 NH 79, Minimally, Salem-636106



DEPARTMENT OF INFORMATION TECHNOLOGY

REGISTER NUMBER	:	_____
STUDENT NAME	:	_____
SEMESTER	:	_____
YEAR	:	_____
SUBJECT CODE	:	_____
SUBJECT NAME	:	_____



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 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 in multidisciplinary 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 change.

CCS334 BIG DATA ANALYTICS LABORATORY

COURSE OBJECTIVES:

- To understand big data.
- To learn and use NoSQL big data management.
- To learn mapreduce 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 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. Installation of Hive along with practice examples.
7. Installation of HBase, Installing thrift along with Practice examples
8. Practice importing and exporting data from various databases.

TOTAL: 30 PERIODS

SOFTWARE AND HARDWARE REQUIREMENTS:

Software Requirements	Cassandra, Hadoop, Java, Pig, Hive and HBase.
Hardware Requirements	Desktop Computer

COURSE OUTCOMES:

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	PO'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	3	1	1	3	3	
2	3	3	2	3	2	-	-	-	2	2	3	3	2	3	2	
3	3	3	3	2	3	-	-	-	2	2	1	2	2	3	3	
4	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	2.8	2.6	

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

INDEX

S.NO	DATE	LIST OF EXPERIMENTS	PAGE NO	MARK	SIGN
CONTEND BEYOND SYLLABUS					

INDEX

S.NO	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 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	Installation of Hive along with practice examples.
6	Installation of HBase, Installing thrift along with Practice example.
7	Practice importing and exporting data from various databases. Software Requirements: Cassandra, Hadoop, Java, Pig, Hive and HBase.
CONTEND BEYOND SYLLABUS	
8	Installation and Creation of database and Collection CRUD Document: Insert, Query, Update and Delete Document.
9	To Create -file, data in memory, other RDD. Lazy Execution, Persisti RDD , RDD, Actions and Transformation on RDD
10	To Create Stories using Tableau ,Connect to data, Build Charts and Analyze Data, Create Dashboard,

CCS334-BIG DATA ANALYTICS LABORATORY

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 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. Installation of Hive along with practice examples.
6. Installation of HBase, Installing thrift along with Practice examples
7. Practice importing and exporting data from various databases. Software Requirements: Cassandra, Hadoop, Java, Pig, Hive and HBase.

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

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- It is recommended to install Oracle Java 8.

The user can check the version of java with below command.

\$ java –version

STEP 1: Setup passwordless ssh

- a) Install Open SSH Server and Open SSH Client

We will now setup the passwordless ssh client with the following command.

1.\$sudo apt-get install openssh-server openssh-client

```
cse@cse-OptiPlex-3020: ~
linux-image-5.15.0-83-generic linux-modules-5.15.0-83-generic
linux-modules-extra-5.15.0-83-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 211 not upgraded.
cse@cse-OptiPlex-3020:~$ ssh-keygen -t rsa -P ""
\Generating public/private rsa key pair.
Enter file in which to save the key (/home/cse/.ssh/id_rsa):
Your identification has been saved in \
Your public key has been saved in \.pub
The key fingerprint is:
SHA256:XDpIM0stcsDhZG52U4hcrexZ1Fn5BeDK05P0VUvITQA cse@cse-OptiPlex-3020
The key's randomart image is:
+---[RSA 3072]----+
| o=+.o.. E++oo |
| =+.oo o.o + . |
| * .Xo. . .+ +. |
| o *o0.+ . o.. |
| .ooS + .. |
| o o + . |
| = . |
| o |
+---[SHA256]---+
cse@cse-OptiPlex-3020:~$
```

- b) Generate Public & Private Key Pairs

2. ssh-keygen -t rsa -P “”

- c) Configure password-less SSH

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

```

cse@cse-OptiPlex-3020: ~
To see these additional updates run: apt list --upgradable
Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
Last login: Tue Sep 26 22:38:47 2023 from 127.0.0.1
cse@cse-OptiPlex-3020:~$ sudo apt-get install rsync
[sudo] password for cse:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.15.0-83-generic linux-hwe-5.15-headers-5.15.0-83
  linux-image-5.15.0-83-generic linux-modules-5.15.0-83-generic
  linux-modules-extra-5.15.0-83-generic
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  rsync
1 upgraded, 0 newly installed, 0 to remove and 210 not upgraded.
Need to get 322 kB of archives.
After this operation, 1,024 B of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 rsync amd64 3
.1.3-8ubuntu0.7 [322 kB]
Fetched 322 kB in 3s (116 kB/s)
(Reading database ... 218232 files and directories currently installed.)

```

d) Now verify the working of password-less ssh

```
$ ssh localhost
```

```

cse@cse-OptiPlex-3020:~$ cat $HOME/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQDK8tsjJ3w9yLOFvmH7P7cWWNCd7jz9vCqG0m6saReytEGssvHDwF9R1WxY8RV/2ggYX9naweT5/Ki1DOr4YAI/ab+ht1iD6v5Df/lcnG+W8mnX020n0cOqgS0EATV/NwjuxapEwQdaG6HcN9XGkgY+jQGyOBZnUdtMy9h1MmqZMNboPJtWo/UW3Tr01AU60guYAYnJ+hmr
e19vvvC9AfQhR0iIZ+PtSPMbnrUUUnXMrLwB0KWAGyIVVHbqc4tgrIFWkvfvKDSTZjfxUnjd8RyQxitYgnKSzRDHGaEwtkedGvlPYZrnq3hJg1asPdhU0rQmt6eyeyCYfc65CVB73lctas8ltdJRV/DyLjDyuWbPUL
ABeVwqiaEZHPc91DBpqlroxNwbvjPQfW/sqAfa7303Rae/pYA48LutZu76uFXFs3cbJ8jt+Y33nvaEQV
PyQtvTg0AaHoeIN/CtgWGEqhsHqcS+03GkpKmlR2siTgyXDVAJ/hDeXkVMxAgkaK60JIKU= cse@cse
-OptiPlex-3020
cse@cse-OptiPlex-3020:~$ ssh localhost
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.15.0-84-generic x86_64)

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

221 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
*** System restart required ***
Last login: Tue Sep 26 22:38:47 2023 from 127.0.0.1
cse@cse-OptiPlex-3020:~$ 

```

e) Now install rsync with command

```
$ sudo apt-get install rsync
```

STEP 2: Configure and Setup Hadoop

Downloading Hadoop

```
$ wget https://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

EXPT.NO.1(b)	Downloading and installing Hadoop; Startup scripts, Configuration files.
---------------------	--

DATE:	
--------------	--

AIM:

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

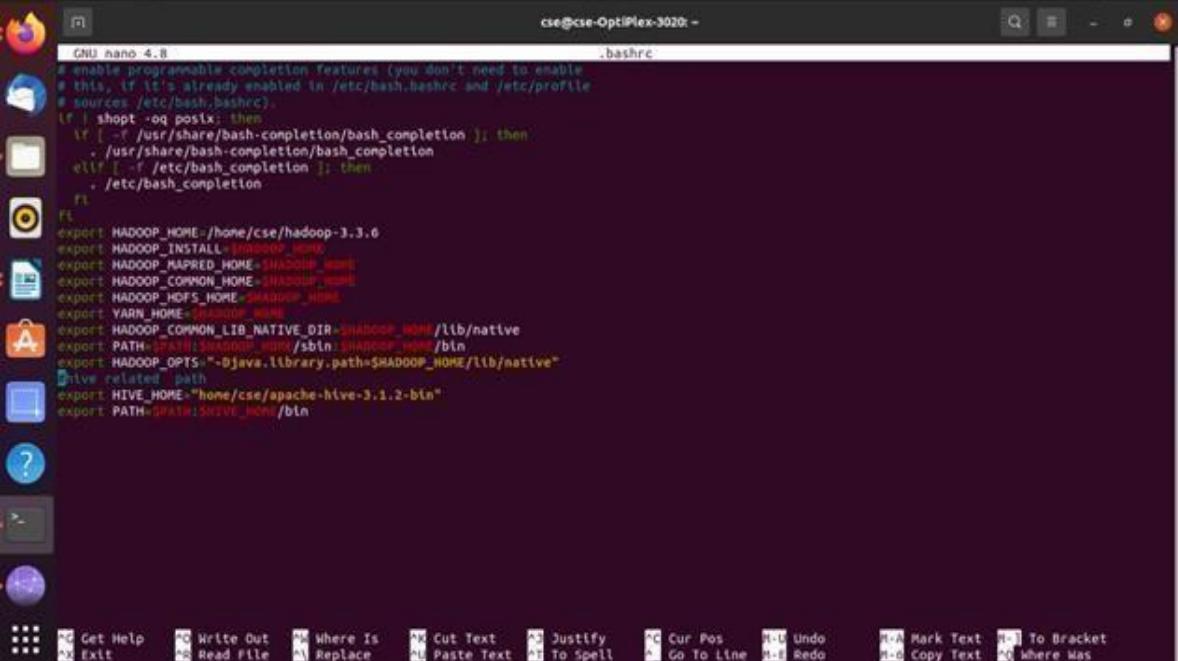
STEP 1: Setup Configuration

- Setting Up the environment variables

Edit .bashrc- Edit the bashrc and therefore add hadoop in a path:

```
$nano bash.bashrc
```

```
export HADOOP_HOME=/home/cse/hadoop-3.3.6
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
```



```
GNU nano 4.8
.cse@cse-OptiPlex-3020: ~
.bashrc
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile)
# sources /etc/bash.bashrc
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
export HADOOP_HOME=/home/cse/hadoop-3.3.6
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"
#hive related path
export HIVE_HOME="/home/cse/apache-hive-3.1.2-bin"
export PATH=$PATH:$HIVE_HOME/bin
```

Source .bashrc in current login session in terminal

```
$source ~/.bashrc
```

- 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/)
```

```

GNU nano 2.5.3          File: hadoop-env.sh

# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
#export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_151

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol. Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

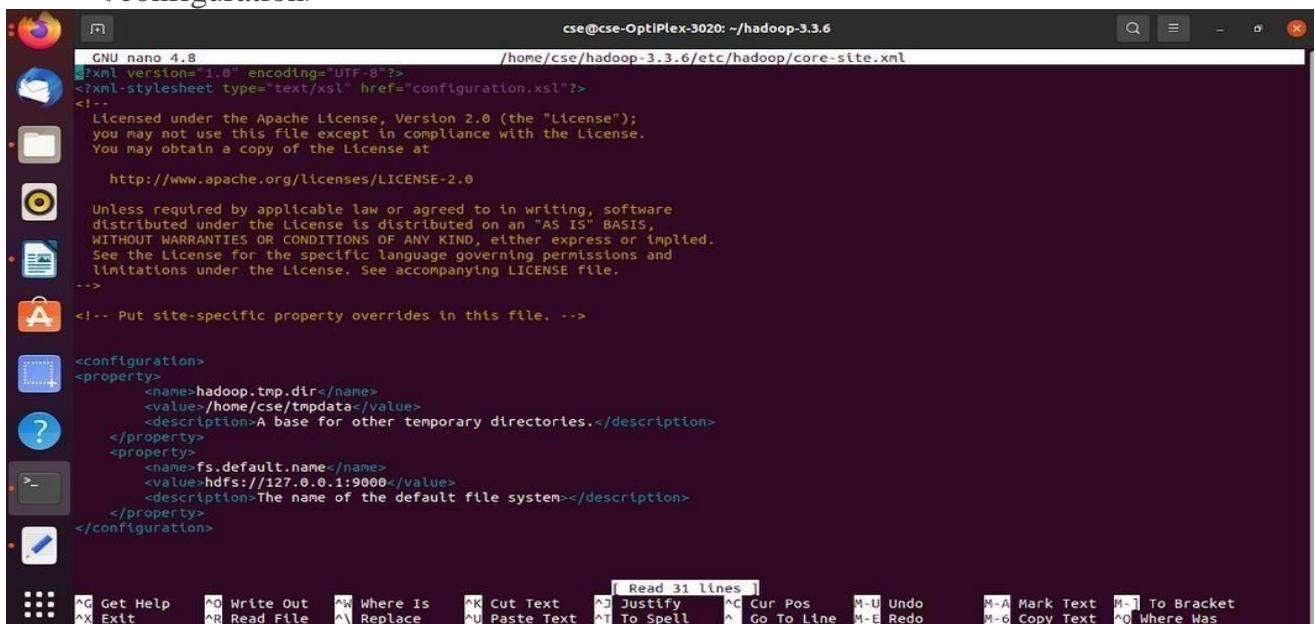
# Extra Java CLASSPATH elements. Automatically insert capacity-scheduler.
for f in $HADOOP_HOME/contrib/capacity-scheduler/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
    export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
  else
    export HADOOP_CLASSPATH=$f
  fi
done

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Linter  ^L Go To Line

```

Edit core-site.xml

```
$sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/cse/hdata</value>
</property>
</configuration>
```



```

cse@cse-OptiPlex-3020: ~/hadoop-3.3.6
GNU nano 4.8          /home/cse/hadoop-3.3.6/etc/hadoop/core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet type="text/xsl" href="configuration.xsl"?>
<!-- Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
--&gt;
<!-- Put site-specific property overrides in this file. --&gt;

&lt;configuration&gt;
&lt;property&gt;
&lt;name&gt;hadoop.tmp.dir&lt;/name&gt;
&lt;value&gt;/home/cse/tmpdata&lt;/value&gt;
&lt;description&gt;A base for other temporary directories.&lt;/description&gt;
&lt;/property&gt;
&lt;property&gt;
&lt;name&gt;fs.default.name&lt;/name&gt;
&lt;value&gt;hdfs://127.0.0.1:9000&lt;/value&gt;
&lt;description&gt;The name of the default file system&lt;/description&gt;
&lt;/property&gt;
&lt;/configuration&gt;

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text  ^T To Spell  ^L Go To Line  M-U Undo  M-A Mark Text  M-T To Bracket
^M-C Copy Text  ^Q Where Was
</pre>

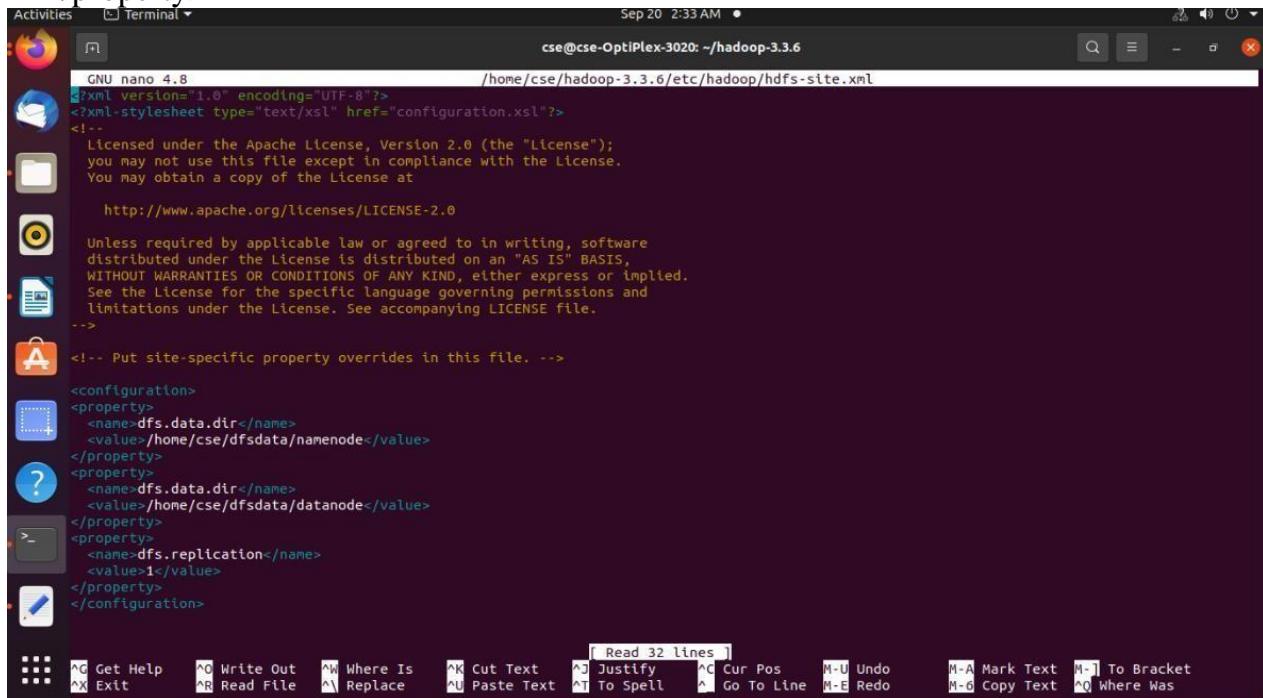
```

Edit hdfs-site.xml

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

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

```
<property>
  <name>dfs.data.dir</name>
  <value>/home/cse/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/cse/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
```



```
GNU nano 4.8          /home/cse/hadoop-3.3.6/etc/hadoop/hdfs-site.xml
c?xml version="1.0" encoding="UTF-8"?>
<x?mlstylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

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

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>dfs.data.dir</name>
    <value>/home/cse/dfsdata/namenode</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/home/cse/dfsdata/datanode</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

The terminal status bar at the bottom includes the following keys:

- Get Help
- Write Out
- Where Is
- Cut Text
- Justify
- Cur Pos
- Undo
- Mark Text
- To Bracket
- Exit
- Read File
- Replace
- Paste Text
- To Spell
- Go To Line
- Redo
- Copy Text
- Where Was

Edit mapred-site.xml

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

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

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

```
cse@cse-OptiPlex-3020: ~ /hadoop-3.3.6
GNU nano 4.8
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

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

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

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

[Read 24 lines]
```

Edit yarn-site.xml

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

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

```
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>127.0.0.1</value>
</property>
<property>
  <name>yarn.acl.enable</name>
  <value>0</value>
</property>
<property>
  <name>yarn.nodemanager.env-whitelist</name>
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
```

```

GNU nano 4.8                               /home/cse/hadoop-3.3.6/etc/hadoop/yarn-site.xml
<!--
 Licensed under the Apache License, Version 2.0 (the "License");
 you may not use this file except in compliance with the License.
 You may obtain a copy of the License at

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

 Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 See the License for the specific language governing permissions and
 limitations under the License. See accompanying LICENSE file.
-->
<configuration>
<properties>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
<name>yarn.resourcemanager.hostname</name>
<value>127.0.0.1</value>
</property>
<property>
<name>yarn.acl.enable</name>
<value></value>
</property>
<property>
<name>yarn.nodemanager.env-whitelist</name>
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>

```

File operations: Get Help, Write Out, Where Is, Cut Text, Justify, Cur Pos, Undo, Mark Text, To Bracket, Exit, Read File, Replace, Paste Text, To Spell, Go To Line, Redo, Copy Text, Where Was.

Step 2: Start the cluster

We will now start the single node cluster with the following commands.

- Format the namenode

\$hdfs namenode –format

```

17/11/66 01:55:11 INFO blockmanagement.BlockManager: encryptDataTransfer      = false
17/11/66 01:55:11 INFO blockmanagement.BlockManager: maxNumBlocksToLog        = 1000
17/11/66 01:55:12 INFO namenode.FSNamesystem: fsOwner                  = hduuser (auth:SIMPLE)
17/11/66 01:55:12 INFO namenode.FSNamesystem: supergroup                = supergroup
17/11/66 01:55:12 INFO namenode.FSNamesystem: isPermissionEnabled = true
17/11/66 01:55:12 INFO namenode.FSNamesystem: HA Enabled: false
17/11/66 01:55:12 INFO namenode.FSNamesystem: Append Enabled: true
17/11/66 01:55:12 INFO util.GSet: Computing capacity for map INodeMap
17/11/66 01:55:12 INFO util.GSet: VM type       = 32-bit
17/11/66 01:55:12 INFO util.GSet: 1.0% max memory 966.7 MB = 9.7 MB
17/11/66 01:55:12 INFO util.GSet: capacity      = 2^21 = 2097152 entries
17/11/66 01:55:12 INFO namenode.FSDirectory: ACLs enabled? false
17/11/66 01:55:12 INFO namenode.NameNode: Caching file names occurring more than 10 times
17/11/66 01:55:12 INFO util.GSet: Computing capacity for map cachedBlocks
17/11/66 01:55:12 INFO util.GSet: VM type       = 32-bit
17/11/66 01:55:12 INFO util.GSet: 0.25% max memory 966.7 MB = 2.4 MB
17/11/66 01:55:12 INFO util.GSet: capacity      = 2^19 = 524288 entries
17/11/66 01:55:12 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
17/11/66 01:55:12 INFO namenode.FSNamesystem: dfs.namenode.safemode.mln.datanodes = 0
17/11/66 01:55:12 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension   = 30000
17/11/66 01:55:12 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
17/11/66 01:55:12 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
17/11/66 01:55:12 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
17/11/66 01:55:12 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
17/11/66 01:55:12 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
17/11/66 01:55:12 INFO util.GSet: Computing capacity for map NameNodeRetryCache
17/11/66 01:55:12 INFO util.GSet: VM type       = 32-bit
17/11/66 01:55:12 INFO util.GSet: 0.02999999329447746% max memory 966.7 MB = 297.0 KB
17/11/66 01:55:12 INFO util.GSet: capacity      = 2^16 = 65536 entries
17/11/66 01:55:12 INFO namenode.FSImage: Allocated new BlockPoolId: BP-2001603931-127.0.1.1-1509962112981
17/11/66 01:55:13 INFO common.Storage: Storage directory /home/hduuser/hdata/dfs/name has been successfully formatted.
17/11/66 01:55:13 INFO namenode.FSImageFormatProtobuf: Saving image file /home/hduuser/hdata/dfs/name/current/fsimage.ckpt_00000000000000000000000000000000 using no compression
17/11/66 01:55:13 INFO namenode.FSImageFormatProtobuf: Image file /home/hduuser/hdata/dfs/name/current/fsimage.ckpt_00000000000000000000000000000000 of size 323 bytes saved in 0 seconds.
17/11/66 01:55:13 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
17/11/66 01:55:13 INFO util.ExitUtil: Exiting with status 0
17/11/66 01:55:13 INFO namenode.NameNode: SHUTDOWN_MSG:
*****
```

- Start the HDFS

\$start-all.sh

- Verify if all process started

\$ jps

6775 DataNode

7209 ResourceManager

7017 SecondaryNameNode

6651 NameNode

7339 NodeManager

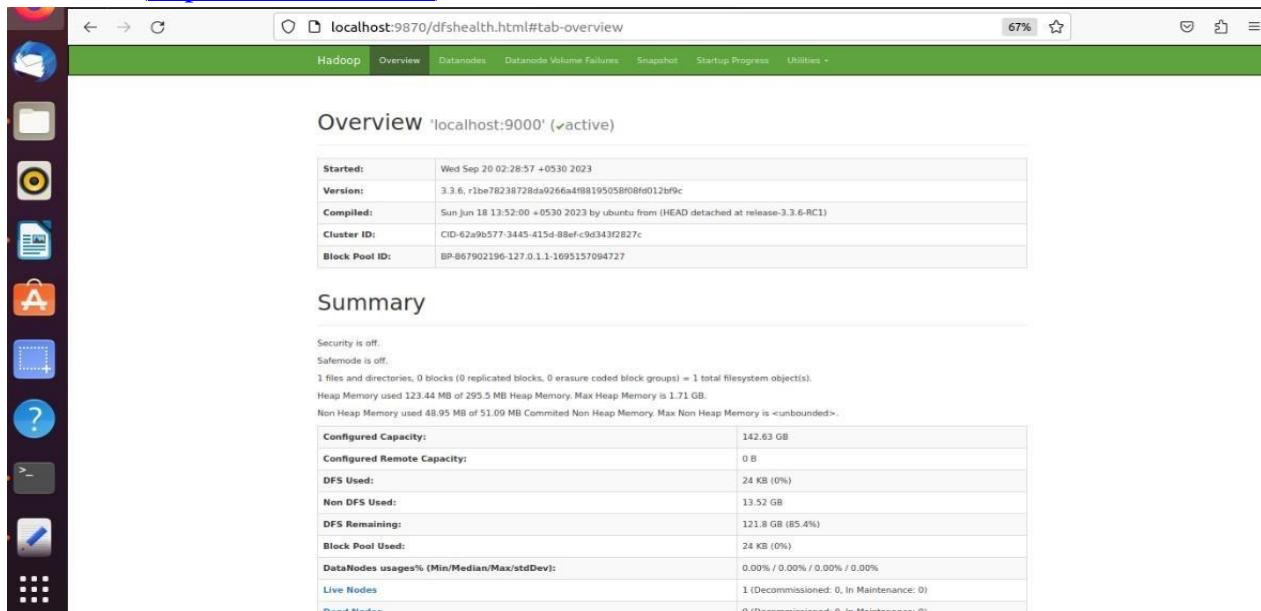
7663 Jps

```

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 NameNode
visit : (<http://localhost:9870>)



The screenshot shows a web browser window with the URL `localhost:9870/dfshealth.html#tab-overview`. The page has a green header bar with tabs for Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities.

Overview section:

Started:	Wed Sep 20 02:28:57 +0530 2023
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-62a9b577-3445-4154-88ef-c9d34f2827c
Block Pool ID:	BP-867902196-127.0.1.1-1695157094727

Summary section:

Security is off.
Safemode is off.
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).
Heap Memory used 123.44 MB of 295.5 MB Heap Memory. Max Heap Memory is 1.71 GB.
Non Heap Memory used 48.95 MB of 51.09 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	142.63 GB
Configured Remote Capacity:	0 B
DFS Used:	24 KB (0%)
Non DFS Used:	13.52 GB
DFS Remaining:	121.8 GB (85.4%)
Block Pool Used:	24 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)

Result:

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

EXPT.NO.2	Hadoop Implementation of file management tasks, such as Adding files and directories, retrieving files and Deleting files
DATE:	

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 scalable distributed filesystem designed to scale to petabytes of data while running on top of the underlying filesystem 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 dataset. This allows Hadoop to efficiently schedule tasks to those nodes that contain data, or which are nearest to it, optimizing bandwidth 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:

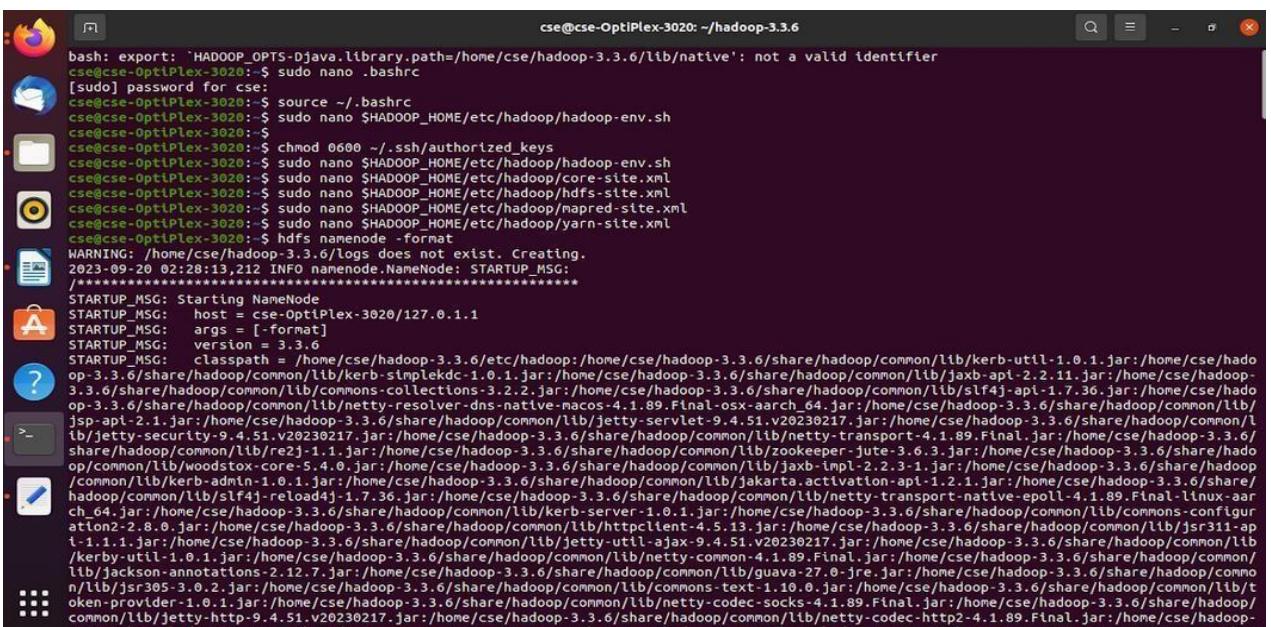
1. Adding files and directories to HDFS
2. Retrieving files from HDFS to local filesystem
3. Deleting files from HDFS

SYNTAX AND COMMANDS TO ADD, RETRIEVE AND DELETE DATA FROM HDFS

Step 1:Starting HDFS

Initially you have to format the configured HDFS file system, open namenode (HDFS server), and execute the following command.

\$ hadoop namenode -format



```
cse@cse-OptiPlex-3020: ~/hadoop-3.3.6
bash: export: `HADOOP_OPTS-Djava.library.path=/home/cse/hadoop-3.3.6/lib/native': not a valid identifier
cse@OptiPlex-3020: $ sudo nano .bashrc
[sudo] password for cse:
cse@OptiPlex-3020: $ source ~/.bashrc
cse@OptiPlex-3020: $ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
cse@OptiPlex-3020: $ chmod 0600 ~/_ssh/authorized_keys
cse@OptiPlex-3020: $ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
cse@OptiPlex-3020: $ sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
cse@OptiPlex-3020: $ sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
cse@OptiPlex-3020: $ sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
cse@OptiPlex-3020: $ sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
cse@OptiPlex-3020: $ hdfs namenode -format
WARNING: /home/cse/hadoop-3.3.6/logs does not exist. Creating.
2023-09-20 02:28:13,218 INFO namenode.Namenode: STARTUP_MSG:
 ****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = cse-OptiPlex-3020/127.0.1.1
STARTUP_MSG: args = [format]
STARTUP_MSG: version = 3.3.6
STARTUP_MSG: classpath = /home/cse/hadoop-3.3.6/etc/hadoop:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/kerb-util-1.0.1.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jaxb-api-2.2.11.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/commons-collections-3.2.2.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/slf4j-api-1.7.36.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/netty-resolver-dns-native-macos-4.1.89.Final-osx-aarch_64.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jsp-api-2.1.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jetty-servlet-9.4.51.v20230217.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jetty-security-9.4.51.v20230217.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/netty-transport-4.1.89.Final.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/woodstox-core-5.4.0.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jaxb-impl-2.2.3-1.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/kerb-admin-1.0.1.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jakarta.activation-api-1.2.1.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/netty-transport-native-epoll-4.1.89.Final-linux-aarch_64.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/kerb-server-1.0.1.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/commons-configuration2-2.8.0.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/httpclient-4.5.13.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jsr311-api-1.1.1.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jetty-util-ajax-9.4.51.v20230217.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/kerby-util-1.0.1.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/netty-common-4.1.89.Final.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jackson-annotations-2.12.7.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/guava-27.0-jre.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jsr305-3.0.2.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/commons-text-1.10.0.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/netty-codec-socks-4.1.89.Final.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/jetty-http-9.4.51.v20230217.jar:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/netty-codec-http2-4.1.89.Final.jar:/home/cse/hadoop-
```

After formatting the HDFS, start the distributed file system. The following command will start the namenode as well as the data nodes as cluster.

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

Listing Files in HDFS

After loading the information in the server, we can find the list of files in a directory, status of a file, using ls Given below is the syntax of ls that you can pass to a directory or a filename as an argument.

```
$ $HADOOP_HOME/bin/hadoop fs -ls <args>
```

```
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:~$
```

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 file system.

Step-2: Adding Files and Directories to HDFS

```
$ $HADOOP_HOME/bin/hadoop fs -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 applicable
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
```

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 verify the file using ls command.

```
$ $HADOOP_HOME/bin/hadoop fs -ls /user/input
```

Step 4 Retrieving Data from HDFS

Assume we have a file in HDFS called outfile. Given below is a simple demonstration for retrieving the required file from the Hadoop file system.

Initially, view the data from HDFS using cat command.

```
$ $HADOOP_HOME/bin/hadoop fs -cat /user/output/outfile
```

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

```
$ $HADOOP_HOME/bin/hadoop fs -get /user/output/ /home/hadoop_tp/
```

Step-5: Deleting Files from HDFS

```
$ hadoop fs -rm file.txt
```

Step 6:Shutting Down the HDFS

You can shut down the HDFS by using the following command.

```
$ stop-dfs.sh
```

Result:

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

EXPT.NO.3(a)

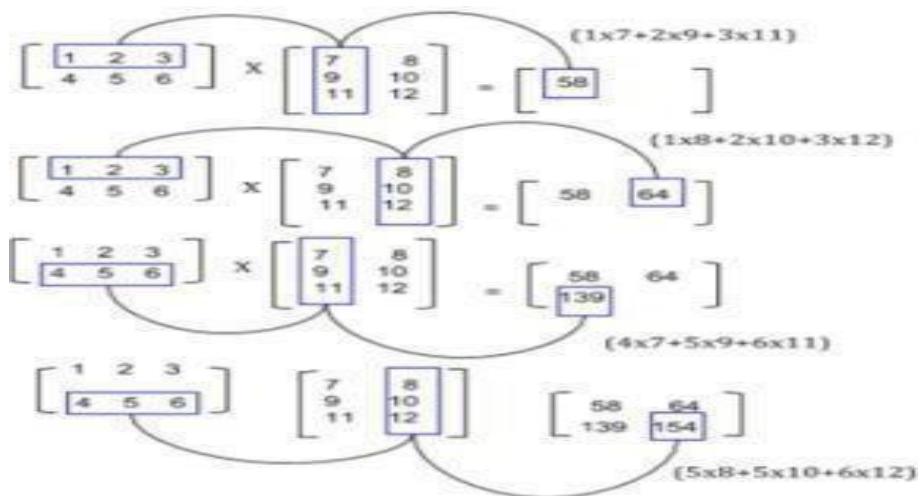
Implement of Matrix Multiplication with Hadoop Map Reduce

DATE:

AIM: To Develop a MapReduce 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** is an $n \times m$ matrix and **B** is an $m \times p$ matrix, their matrix product **AB** is an $n \times p$ matrix, in which the m entries across a row of **A** are multiplied with the m entries 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.

- 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**
- for each element n_{jk} of **N** do

produce (key,value) pairs as ((i,k),(N,j,N_{jk}), for i = 1,2,3,.. Upto the number of rows of M.

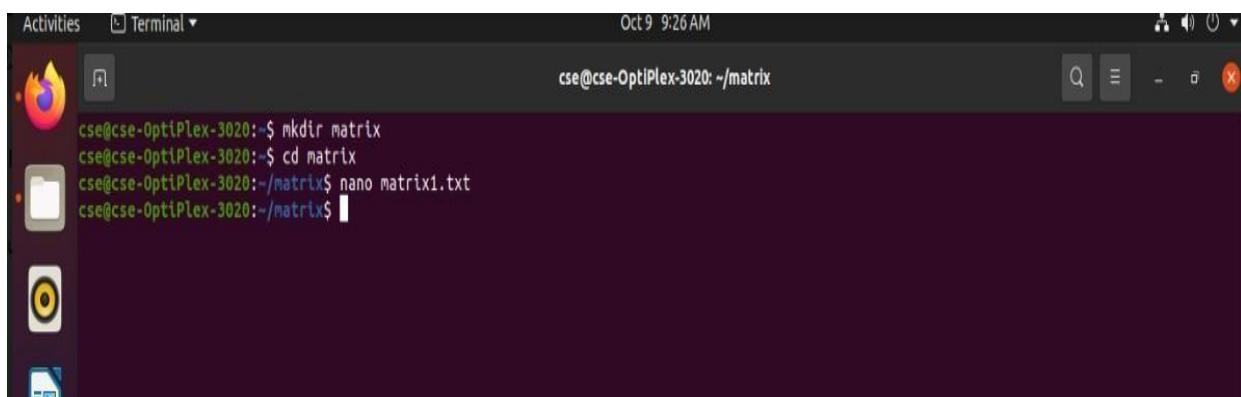
- c. return Set of (key,value) pairs that each key (i,k), has list with values (M,j,m_{ij}) and (N, j,n_{jk}) for all possible values of j.

Algorithm for Reduce Function.

- d. for each key (i,k) do
- e. sort values begin with M by j in listM sort values begin with N by j in list N multiply m_{ij} and n_{jk} for jth value of each list
- f. sum up m_{ij} x n_{jk} return (i,k), $\sum_{j=1}^n m_{ij} \times n_{jk}$

Step 1. Creating directory for matrix

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



```
Activities Terminal Oct 9 9:26 AM
cse@cse-OptiPlex-3020: ~/
cse@cse-OptiPlex-3020: $ mkdir matrix
cse@cse-OptiPlex-3020: $ cd matrix
cse@cse-OptiPlex-3020: ~/matrix$ nano matrix1.txt
cse@cse-OptiPlex-3020: ~/matrix$
```

Step 2. Creating Mapper file for Matrix Multiplication.

```
#!/usr/bin/env python
import 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 xrange(0, row_a):
            key = str(j) + "," + col
            print "%s\t%s\t%s" % (key, row, value)
```

Step 3. Creating reducer file for Matrix Multiplication.

```
#!/usr/bin/env python
import sys
from operator import itemgetter
prev_index = None
value_list = []
```

```

for line in sys.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
        while i < 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))
        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
    while i < 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

0	0	0	1
0	1	0	1
0	0	1	2
0	1	1	2
0	0	2	3
0	1	2	3
1	0	0	4
1	1	0	4
1	0	1	5
1	1	1	5
1	0	2	6
1	1	2	6
0	0	0	7
1	0	0	7
0	1	0	8
1	1	0	8
0	0	1	9
1	0	1	9
0	1	1	10
1	1	1	10
0	0	2	11
1	0	2	11
0	1	2	12
1	1	2	12

```
$ 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
```

Step 5: To view this full output

```
[14, 77]  
[194, 365]
```

Result:

Thus the MapReduce program to implement Matrix Multiplication was successfully executed

EXPT.NO.4	Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm
DATE:	

AIM: To Develop a MapReduce program to calculate the frequency of a given word in a given file
Map Function – It takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (Key-Value pair).

Example – (Map function in Word Count)

Input

Set of data

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

Output

Convert into another

set 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 – (Reduce function in Word Count)

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)

Output Converts into smaller set of tuples

(BUS,7), (CAR,7), (TRAIN,4)

Workflow of MapReduce consists of 5 steps

- 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 order to 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 each cluster) is combined 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

```
cse@cse-OptiPlex-3020: ~/Desktop
cse@cse-OptiPlex-3020: ~/Desktop$ hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /home/cse/hadoop-3.3.6/share/hadoop/common/hadoop-common-3.3.6.jar
cse@cse-OptiPlex-3020: ~/Desktop$
```

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

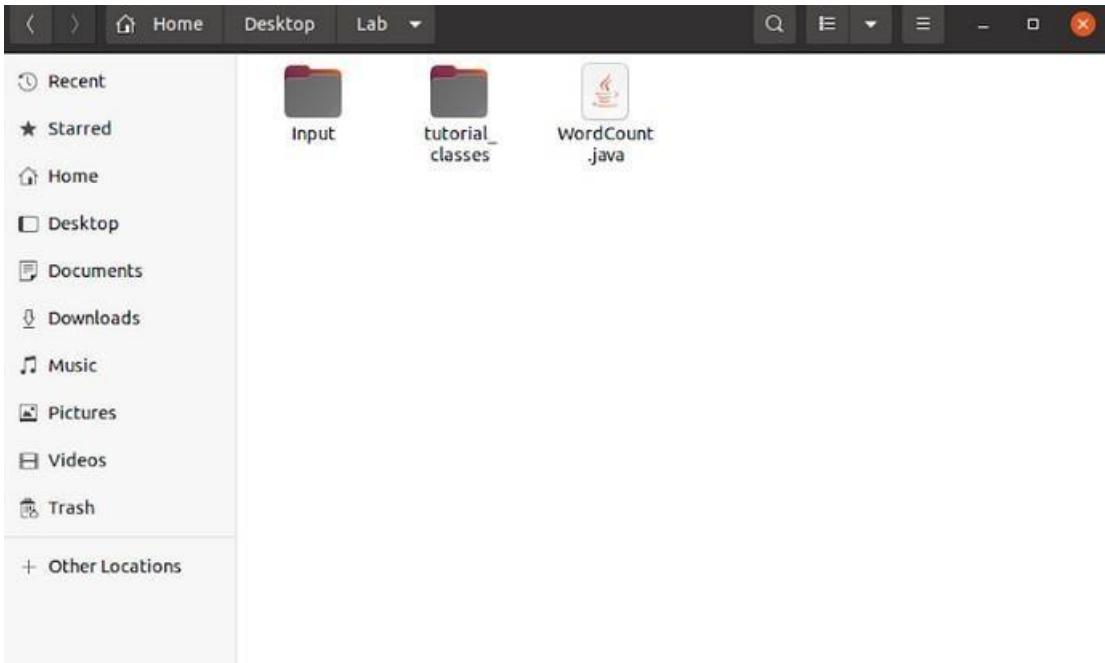
mkdir Lab/Input

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.



Step 5. Type the following command to export the hadoop classpath into bash.

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

Make sure it is now exported.

```
echo $HADOOP_CLASSPATH
```

Step 6. It is time to create these directories on HDFS rather than locally. Type the following commands.

```
hadoop fs -mkdir /WordCountTutorial
```

```
hadoop fs -mkdir /WordCountTutorial/Input
```

```
hadoop fs -put Lab/Input/input.txt /WordCountTutorial/Input
```

```
cse@cse-OptiPlex-3020: ~ cd Desktop
cse@cse-OptiPlex-3020: ~/Desktop$ hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /home/cse/hadoop-3.3.6/share/hadoop/common/hadoop-common-3.3.6.jar
cse@cse-OptiPlex-3020: ~/Desktop$ mkdir Lab/tutorial_classes
cse@cse-OptiPlex-3020: ~/Desktop$ export HADOOP_CLASSPATH=$(hadoop classpath)
cse@cse-OptiPlex-3020: ~/Desktop$ echo $HADOOP_CLASSPATH
/home/cse/hadoop-3.3.6/etc/hadoop:/home/cse/hadoop-3.3.6/share/hadoop/common/lib/*:/home/cse/hadoop-3.3.6/share/hadoop/common/*:/home/cse/hadoop-3.3.6/share/hadoop/hdfs:/home/cse/hadoop-3.3.6/share/hadoop/hdfs/lib/*:/home/cse/hadoop-3.3.6/share/hadoop/hdfs/*:/home/cse/hadoop-3.3.6/share/hadoop/mapreduce/*:/home/cse/hadoop-3.3.6/share/hadoop/yarn:/home/cse/hadoop-3.3.6/share/hadoop/yarn/lib/*:/home/cse/hadoop-3.3.6/share/hadoop/yarn/*
cse@cse-OptiPlex-3020: ~/Desktop$ hadoop fs -mkdir /WordCountTutorial
cse@cse-OptiPlex-3020: ~/Desktop$ hadoop fs -mkdir /WordCountTutorial/Input
cse@cse-OptiPlex-3020: ~/Desktop$ hadoop fs -put Lab/Input/input.txt /WordCountTutorial/Input
cse@cse-OptiPlex-3020: ~/Desktop$ cd Lab
cse@cse-OptiPlex-3020: ~/Desktop$ javac -classpath $HADOOP_CLASSPATH -d '/home/cse/Desktop/Lab/tutorial_classes' '/home/cse/Desktop/Lab/WordCount.java'
javac: invalid flag: -d'/home/cse/Desktop/Lab/tutorial_classes'
Usage: javac <options> <source files>
use -help for a list of possible options
cse@cse-OptiPlex-3020: ~/Desktop$ javac -classpath $HADOOP_CLASSPATH -d '/home/cse/Desktop/Lab/tutorial_classes' '/home/cse/Desktop/Lab/WordCount.java'
cse@cse-OptiPlex-3020: ~/Desktop$ jar -cvf WordCount.jar -C tutorial_classes .
added manifest
adding: WordCount$IntSumReducer.class(in = 1739) (out= 739)(deflated 57%)
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 754)(deflated 56%)
adding: WordCount.class(in = 1491) (out= 814)(deflated 45%)
cse@cse-OptiPlex-3020: ~/Desktop$
```

Step 7. Go to localhost:9870 from the browser, Open “Utilities→ Browse File System” and you should see the directories and files we placed in the file system.

The screenshot shows a browser window with the URL `localhost:9870/explorer.html#/`. The page title is "Browse Directory". The address bar shows the path `/WordCountTutorial`. The main content area displays a table of file entries:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
<input type="checkbox"/>	drwxr-xr-x	cse	supergroup	0 B	Sep 20 02:49	0	0 B	WordCountTutorial

Below the table, it says "Showing 1 to 1 of 1 entries". At the bottom, there are navigation links: "Previous", "1", and "Next". A footer message at the bottom left says "Hadoop, 2023."

This screenshot shows the same Hadoop File Explorer interface, but with a modal dialog box open over the main content. The dialog is titled "File information - input.txt" and is focused on "Block information - Block 0". The modal contains the following details:

- Block ID: 1073741825
- Block Pool ID: BP-867902196-127.0.1.1-1695157094727
- Generation Stamp: 1001
- Size: 24
- Availability:
 - cse-OptiPlex-3020

At the bottom right of the modal is a "Close" button.

Step 8. Then, back to local machine where we will compile the WordCount.java file.

Assuming we are currently in the Desktop directory.

cd Lab

javac -classpath \$HADOOP_CLASSPATH -d tutorial_classes WordCount.java

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

jar -cvf WordCount.jar -C tutorial_classes .

Step 9. Now, we run the jar file on Hadoop.

hadoop jar WordCount.jar WordCount /WordCountTutorial/Input /WordCountTutorial/Output

```

cse@cse-OptiPlex-3020: ~/Desktop/Lab
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.apache.hadoop.util.RunJar.run(RunJar.java:328)
at org.apache.hadoop.util.RunJar.main(RunJar.java:241)
cse@cse-OptiPlex-3020:~/Desktop/Lab$ hadoop jar WordCount.jar WordCount /WordCountTutorial/Input /WordCountTutorial/Output
2023-09-20 23:38:38,288 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2023-09-20 23:38:38,755 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2023-09-20 23:38:38,796 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/cse/.staging/job_1695232596846_0001
32596846_0001
2023-09-20 23:38:39,682 INFO input.FileInputFormat: Total input files to process : 1
2023-09-20 23:38:40,719 INFO mapreduce.JobSubmitter: number of splits:1
2023-09-20 23:38:41,002 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1695232596846_0001
2023-09-20 23:38:41,002 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-09-20 23:38:41,203 INFO conf.Configuration: resource-types.xml not found
2023-09-20 23:38:41,203 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-09-20 23:38:41,631 INFO impl.YarnClientImpl: Submitted application application_1695232596846_0001
2023-09-20 23:38:41,679 INFO mapreduce.Job: The url to track the job: http://cse-OptiPlex-3020:8088/proxy/application_1695232596846_0001/
2023-09-20 23:38:41,679 INFO mapreduce.Job: Running job: job_1695232596846_0001
2023-09-20 23:38:48,867 INFO mapreduce.Job: Job job_1695232596846_0001 running in uber mode : false
2023-09-20 23:38:48,869 INFO mapreduce.Job: map 0% reduce 0%
2023-09-20 23:38:54,004 INFO mapreduce.Job: map 100% reduce 0%
2023-09-20 23:38:59,035 INFO mapreduce.Job: map 100% reduce 100%
2023-09-20 23:38:59,046 INFO mapreduce.Job: Job job_1695232596846_0001 completed successfully
2023-09-20 23:38:59,140 INFO mapreduce.Job: Counters: 54
File System Counters
FILE: Number of bytes read=48
FILE: Number of bytes written=551467
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=144
HDFS: Number of bytes written=26
HDFS: Number of read operations=8
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
HDFS: Number of bytes read erasure-coded=0
Job Counters
```

Step 10. Output the result:

```

cse@cse-OptiPlex-3020: ~/Desktop/Lab
GC time elapsed (ms)=89
CPU time spent (ms)=980
Physical memory (bytes) snapshot=540114944
Virtual memory (bytes) snapshot=5090471936
Total committed heap usage (bytes)=460849152
Peak Map Physical memory (bytes)=318799872
Peak Map Virtual memory (bytes)=2540630016
Peak Reduce Physical memory (bytes)=221315072
Peak Reduce Virtual memory (bytes)=2549841920
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=24
File Output Format Counters
Bytes Written=26
cse@cse-OptiPlex-3020:~/Desktop/Lab$ hadoop dfs -cat /WordCountTutorial/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.
as      3
fgvu   1
hbjk   1
hgjh   1
```

hadoop dfs -cat /WordCountTutorial/Output/*

Program: Step 5. Type following Program :

```

package PackageDemo;
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.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class WordCount {
public static void main(String [] args) throws Exception
{
Configuration c=new Configuration();
String[] files=new
GenericOptionsParser(c,args).getRemainingArgs(); Path
input=new Path(files[0]);
Path output=new
Path(files[1]); Job j=new
Job(c,"wordcount");
j.setJarByClass(WordCount.c
lass);
j.setMapperClass(MapForWordCount.class);
j.setReducerClass(ReduceForWordCount.class);
j.setOutputKeyClass(Text.class);
j.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(j, input);
FileOutputFormat.setOutputPath(j, output);
System.exit(j.waitForCompletion(true)?0:1);
}
public static class MapForWordCount extends Mapper<LongWritable, Text,
Text,IntWritable>{
public void map(LongWritable key, Text value, Context con) throws IOException,
InterruptedException
{
String line = value.toString();
String[]
words=line.split(",");
for(String word: words
)
{
Text outputKey = new
Text(word.toUpperCase().trim());IntWritable
outputValue = new IntWritable(1);
con.write(outputKey, outputValue);
}
}
}
}
public static class ReduceForWordCount extends Reducer<Text, IntWritable,
Text,IntWritable>
{
public void reduce(Text word, Iterable<IntWritable> values, Context con)
throws IOException,
```

```
InterruptedException
{
    int sum = 0;
    for(IntWritable value : values)
    {
        sum += value.get();
    }
    con.write(word, new IntWritable(sum));
}
```

The output is stored in /r_output/part-00000

```
cse@cse-OptiPlex-3020:~/Desktop/Lab$ hadoop dfs -cat /WordCountTutorial/Output/*
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

? as      3
    fvgu   1
    hbjk   1
    hgjh   1
```

OUTPUT:

The screenshot shows the Hadoop File Browser interface at localhost:9870/explorer.html#/WordCountTutorial/Output. The top navigation bar includes links for Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main title is "Browse Directory". A search bar contains the path "/WordCountTutorial/Output". Below the search bar are buttons for file operations: copy, move, delete, and refresh. A dropdown menu shows "Show 25 entries". A search input field is labeled "Search:". The table below lists two entries:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	-r--r--r--	cse	supergroup	0 B	Sep 20 23:38	1	128 MB	SUCCESS	<input type="button" value="Delete"/>
<input type="checkbox"/>	-r--r--r--	cse	supergroup	26 B	Sep 20 23:38	1	128 MB	part-r-00000	<input type="button" value="Delete"/>

At the bottom, it says "Showing 1 to 2 of 2 entries" and has navigation buttons for "Previous" (disabled), "1" (selected), and "Next".

Result:

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

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

AIM: To installing hive with example

Steps for hive installation

- Download and Unzip Hive
- Edit .bashrc file
- Edit hive-config.sh file
- Create Hive directories in HDFS
- Initiate Derby database
- Configure hive-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 .bashrc file

```
=====
sudo nano .bashrc
```

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

```
cse@cse-OptiPlex-3020: ~
.bashrc
=====
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

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

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

export HADOOP_HOME=/home/cse/hadoop-3.3.6
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"
#hive related path
export HIVE_HOME="/home/cse/apache-hive-3.1.3"
export PATH=$PATH:$HIVE_HOME/bin
```

step 3:

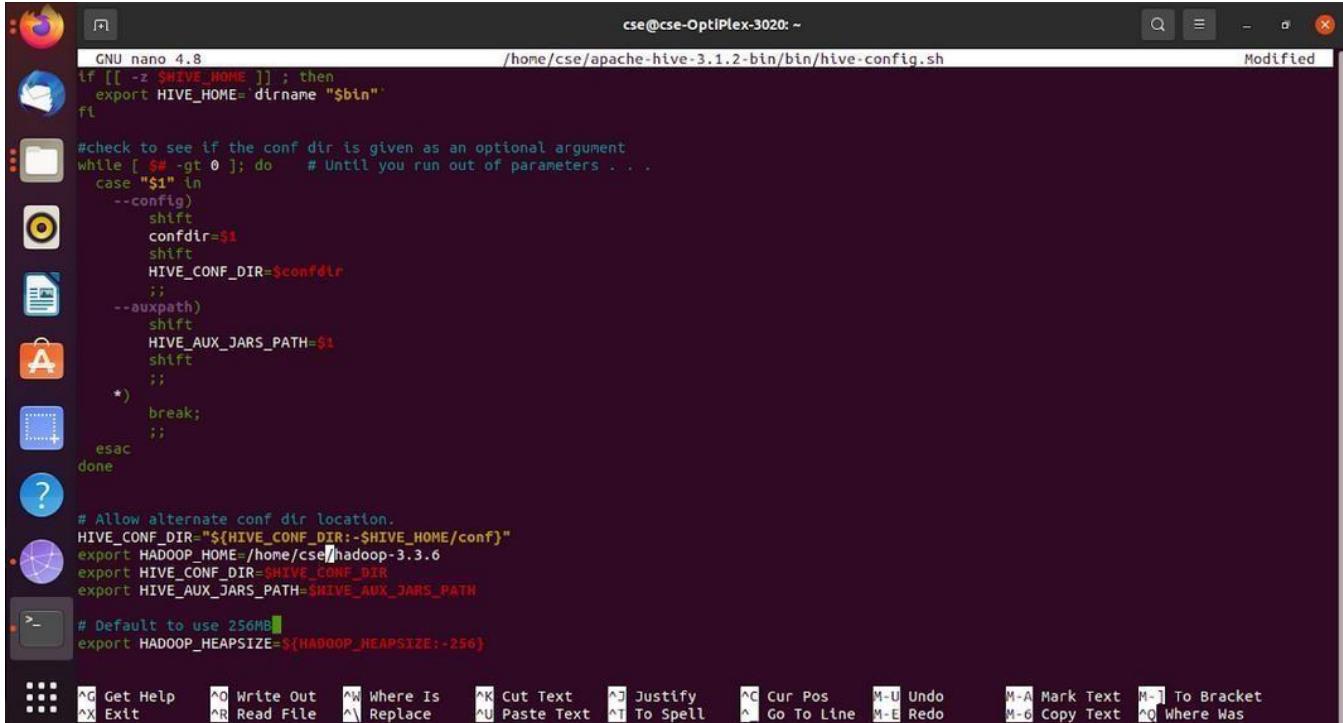
```
source ~/.bashrc
```

step 4:

Edit hive-config.sh file

```
=====
```

```
sudo nano $HIVE_HOME/bin/hive-config.sh  
export HADOOP_HOME=/home/cse/hadoop-3.3.6
```

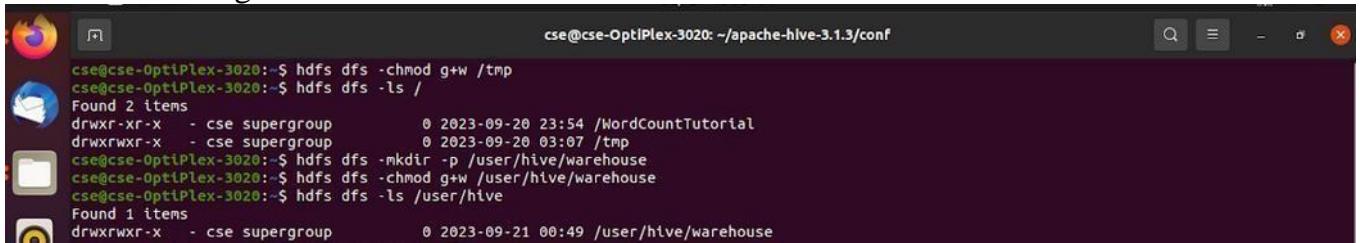


```
GNU nano 4.8                                         /home/cse/apache-hive-3.1.2-bin/bin/hive-config.sh                                         Modified  
  
if [[ -z $HIVE_HOME ]]; then  
    export HIVE_HOME=`dirname "$bin"`  
fi  
  
#check to see if the conf dir is given as an optional argument  
while [ $# -gt 0 ]; do    # Until you run out of parameters . . .  
    case "$1" in  
        --config)  
            shift  
            confdir=$1  
            shift  
            HIVE_CONF_DIR=$confdir  
            ;;  
        --auxpath)  
            shift  
            HIVE_AUX_JARS_PATH=$1  
            shift  
            ;;  
        *)  
            break;  
            ;;  
    esac  
done  
  
# Allow alternate conf dir location.  
HIVE_CONF_DIR="${HIVE_CONF_DIR:-$HIVE_HOME/conf}"  
export HADOOP_HOME=/home/cse/hadoop-3.3.6  
export HIVE_CONF_DIR=$HIVE_CONF_DIR  
export HIVE_AUX_JARS_PATH=$HIVE_AUX_JARS_PATH  
  
# Default to use 256MB  
export HADOOP_HEAPSIZE=${HADOOP_HEAPSIZE:-256}  
  
=====  
^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   M-U Undo  
^X Exit      ^R Read File   ^\ Replace   ^U Paste Text  ^T To Spell   ^I Go To Line  M-E Redo  
M-A Mark Text  M-L To Bracket  
M-D Copy Text M-Q Where Was
```

step 5:

Create Hive directories in HDFS

```
hdfs dfs -mkdir /tmp  
hdfs dfs -chmod g+w /tmp  
hdfs dfs -mkdir -p /user/hive/warehouse  
hdfs dfs -chmod g+w /user/hive/warehouse
```



```
cse@cse-OptiPlex-3020:~$ hdfs dfs -chmod g+w /tmp  
cse@cse-OptiPlex-3020:~$ hdfs dfs -ls /  
Found 2 items  
drwxr-xr-x - cse supergroup          0 2023-09-20 23:54 /WordCountTutorial  
drwxrwxr-x - cse supergroup          0 2023-09-20 03:07 /tmp  
cse@cse-OptiPlex-3020:~$ hdfs dfs -mkdir -p /user/hive/warehouse  
cse@cse-OptiPlex-3020:~$ hdfs dfs -chmod g+w /user/hive/warehouse  
cse@cse-OptiPlex-3020:~$ hdfs dfs -ls /user/hive  
Found 1 items  
drwxrwxr-x - cse supergroup          0 2023-09-21 00:49 /user/hive/warehouse
```

step 6:

Fixing guava problem – 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/
```

step 7: Configure hive-site.xml File (Optional)

Use the following command to locate the correct file:

```
cd $HIVE_HOME/conf
```

List the files contained in the folder using the ls command.

```
cse@cse-OptiPlex-3020:~$ cd apache-hive-3.1.3
cse@cse-OptiPlex-3020:~/apache-hive-3.1.3$ cd conf
cse@cse-OptiPlex-3020:~/apache-hive-3.1.3/conf$ ls
beeline-log4j2.properties.template    hive-exec-log4j2.properties.template    llap-cli-log4j2.properties.template
hive-default.xml.template              hive-log4j2.properties.template      llap-daemon-log4j2.properties.template
hive-env.sh.template                 ivysettings.xml                      parquet-logging.properties
cse@cse-OptiPlex-3020:~/apache-hive-3.1.3/conf$ cp hive-default.xml.template hive-site.xml
cse@cse-OptiPlex-3020:~/apache-hive-3.1.3/conf$
```

Use the hive-default.xml.template to create the hive-site.xml file:

cp hive-default.xml.template hive-site.xml

Access the hive-site.xml file using the nano text editor:

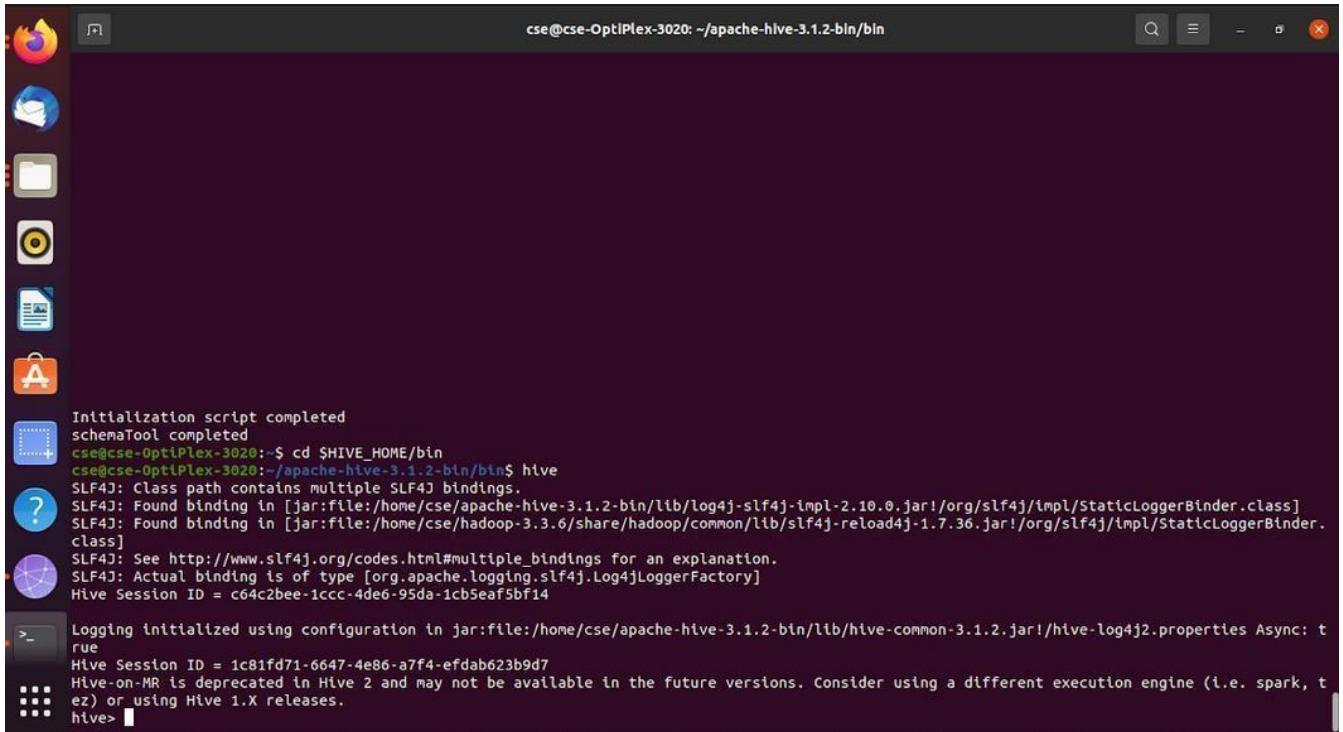
sudo nano hive-site.xml

```
<value>DERBY</value>
<description>
  Expects one of [derby, oracle, mysql, mssql, postgres].
  Type of database used by the metastore. Information schema & JDBCStop
</description>
</property>
<property>
  <name>hive.metastore.warehouse.dir</name>
  <value>/user/hive/warehouse</value>
  <description>location of default database for the warehouse</description>
</property>
<property>
  <name>hive.metastore.warehouse.external.dir</name>
  <value/>
  <description>Default location for external tables created in the warehouse</description>
</property>
<property>
  <name>hive.metastore.uris</name>
  <value/>
  <description>Thrift URI for the remote metastore. Used by metastore client</description>
</property>
```

Step 8: Initiate Derby Database

```
$HIVE_HOME/bin/schematool -dbType derby –initSchema
```

```
cse@cse-OptiPlex-3020: ~/apache-hive-3.1.2-bin/conf
Initialization script completed
schemaTool completed
cse@cse-OptiPlex-3020:~/apache-hive-3.1.2-bin/conf$
```



The screenshot shows a terminal window on an Ubuntu desktop. The terminal title is "cse@cse-OptiPlex-3020: ~/apache-hive-3.1.2-bin/bin". The window contains the following text:

```
Initialization script completed
schemaTool completed
cse@cse-OptiPlex-3020:~$ cd $HIVE_HOME/bin
cse@cse-OptiPlex-3020:~/apache-hive-3.1.2-bin$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/cse/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/cse/hadoop-3.3.6/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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 = c64c2bee-1ccc-4de6-95da-1cb5eaf5bf14
Logging initialized using configuration in jar:file:/home/cse/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = 1c81fd71-6647-4e86-a7f4-efdab623b9d7
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 1.X releases.
hive> ■
```

Result:

Thus Installation hive was successfully installed and executed

EXPT.NO.5(b)

hive with examples

DATE:

AIM:

To installing hive with example

Create Database from Hive Beeline shell

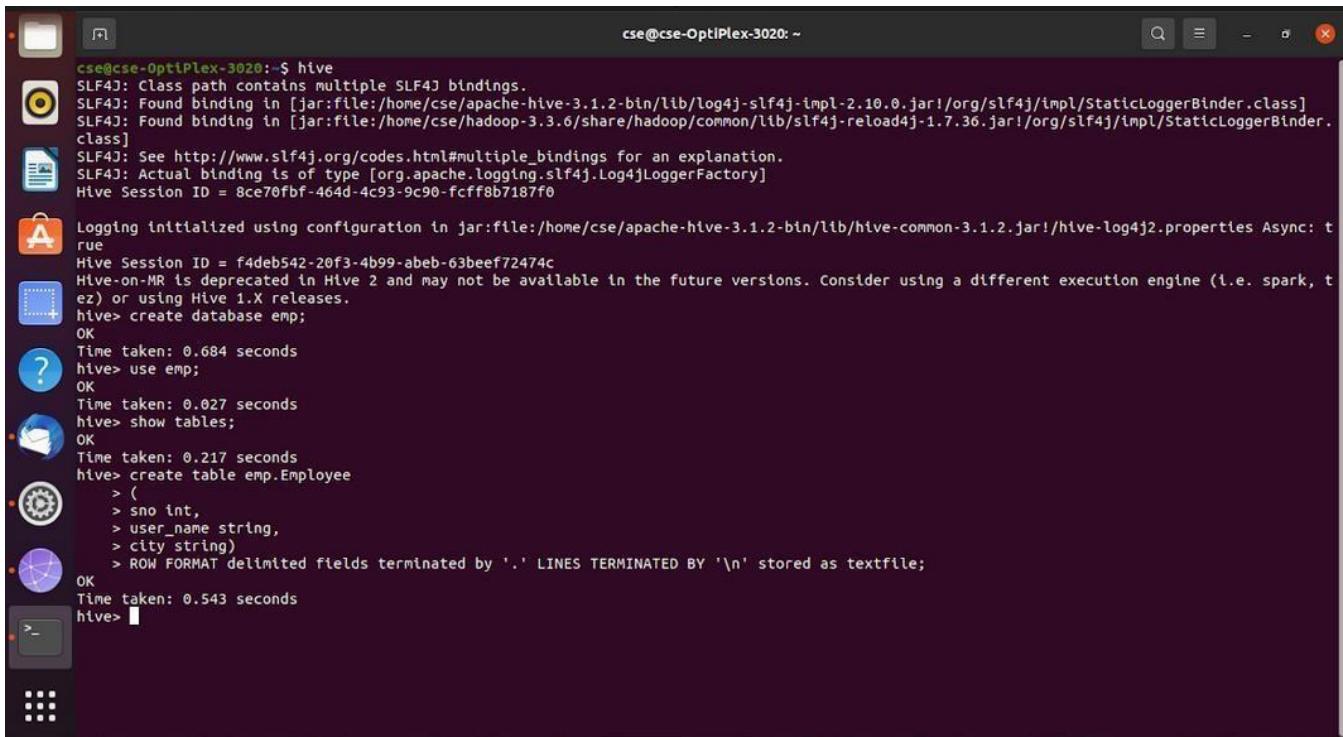
1.Create database database_name;

Ex:

>Create database Emp;

>use Emp;

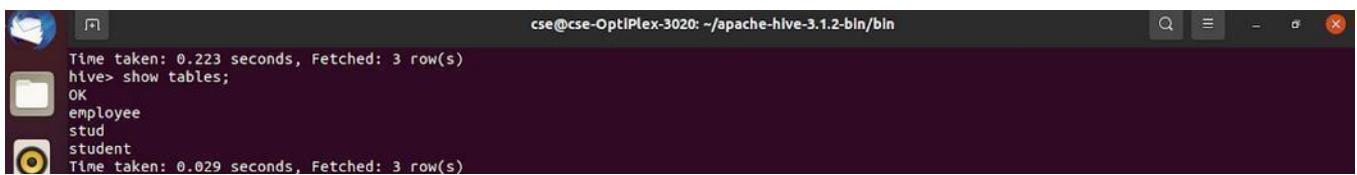
>create table emp.employee(sno int,user String,city String)Row format delimited fields terminated by '/n' stored as textfile;



```
cse@cse-OptiPlex-3020:~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/cse/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/cse/hadoop-3.3.6/share/hadoop/common/lib/slf4j-reload4j-1.7.36.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 = 8ce70fbf-464d-4c93-9c90-fcff8b7187f0

A Logging initialized using configuration in jar:file:/home/cse/apache-hive-3.1.2-bin/lib/hive-common-3.1.2.jar!/hive-log4j2.properties Async: true
Hive Session ID = f4deb542-20f3-4b99-abeb-63beef72474c
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 1.X releases.
hive> create database emp;
OK
Time taken: 0.684 seconds
hive> use emp;
OK
Time taken: 0.027 seconds
hive> show tables;
OK
Time taken: 0.217 seconds
hive> create table emp.Employee
    > (
    > sno int,
    > user_name string,
    > city string)
    > ROW FORMAT delimited fields terminated by '.' LINES TERMINATED BY '\n' stored as textfile;
OK
Time taken: 0.543 seconds
hive> 
```

Show Database



```
cse@cse-OptiPlex-3020:~/apache-hive-3.1.2-bin/bin
Time taken: 0.223 seconds, Fetched: 3 row(s)
hive> show databases;
OK
employee
stud
student
Time taken: 0.029 seconds, Fetched: 3 row(s)
```

Result:

Thus Installation hive was successfully installed and executed with example

EXPT.NO.6(a)

DATE:

Installation of HBase, Installing thrift along with Practice examples

AIM:

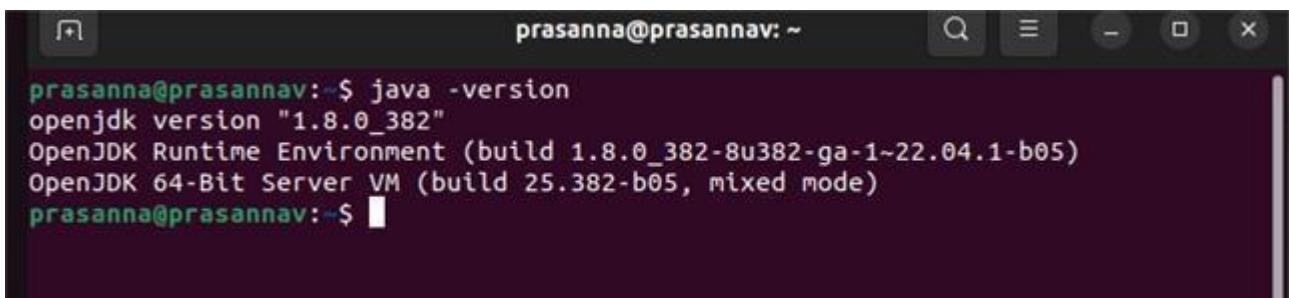
To Install HBase on Ubuntu 18.04 HBase in Standalone Mode

PROCEDURE:

Pre-requisite:

Ubuntu 16.04 or higher installed on a virtual machine.

step-1: Make sure that java has installed in your machine to verify that run java –version



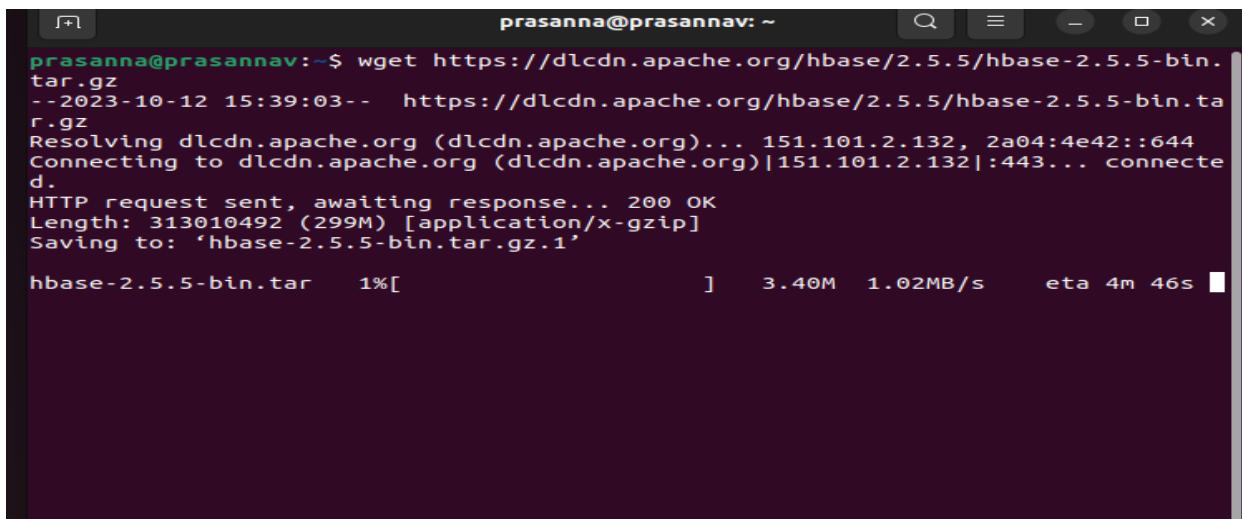
A terminal window titled "prasanna@prasannav: ~". The command "java -version" is run, displaying the following output:
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-8u382-ga-1~22.04.1-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
prasanna@prasannav:~\$

If any Error Occured 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

wget <https://dlcdn.apache.org/hbase/2.5.5/hbase-2.5.5-bin.tar.gz>



A terminal window titled "prasanna@prasannav: ~". The command "wget https://dlcdn.apache.org/hbase/2.5.5/hbase-2.5.5-bin.tar.gz" is run, displaying the following output:
prasanna@prasannav:~\$ wget https://dlcdn.apache.org/hbase/2.5.5/hbase-2.5.5-bin.tar.gz
--2023-10-12 15:39:03-- https://dlcdn.apache.org/hbase/2.5.5/hbase-2.5.5-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 313010492 (299M) [application/x-gzip]
Saving to: 'hbase-2.5.5-bin.tar.gz.1'

hbase-2.5.5-bin.tar 1%[] 3.40M 1.02MB/s eta 4m 46s

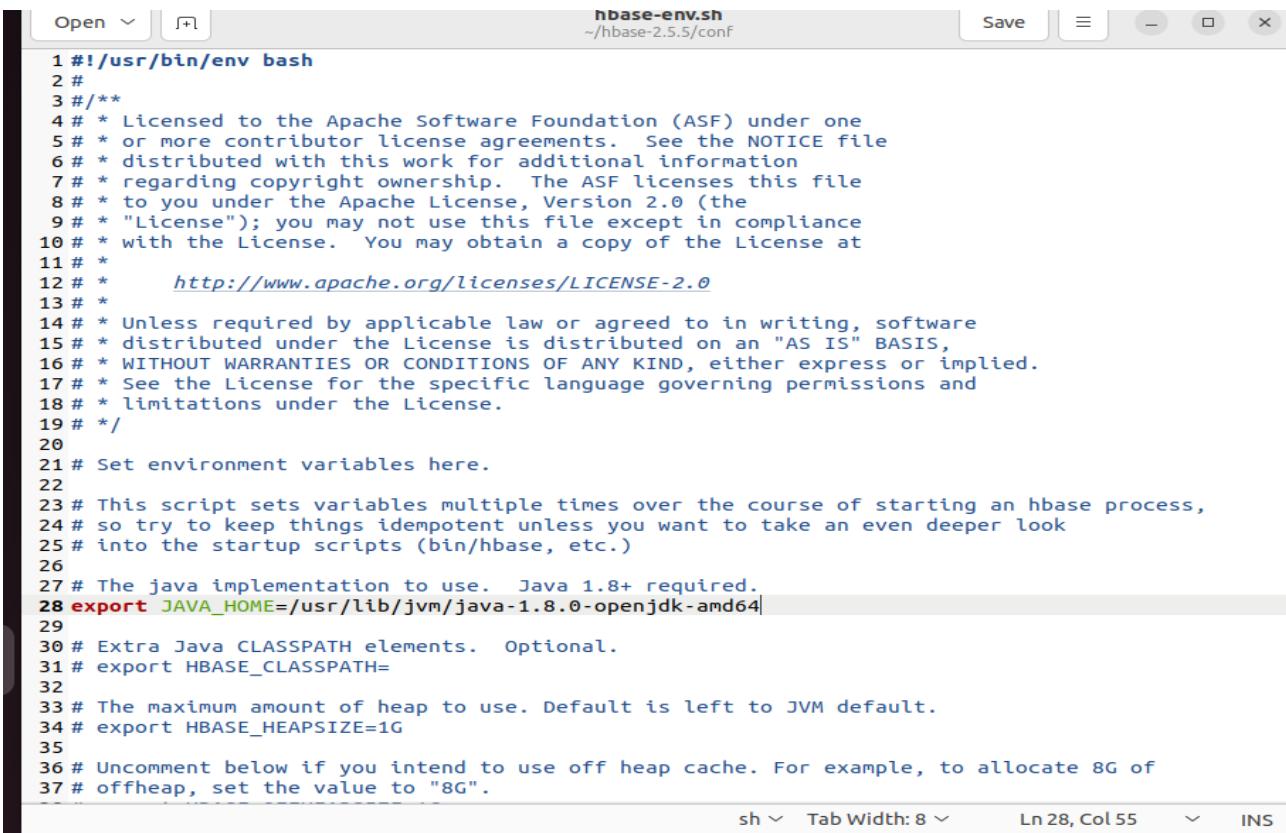
Step-3:Extract The hbase-2.5.5-bin.tar.gz file by using the command tar xvf hbase-2.5.5-bin.tar.gz

```
prasanna@prasannav:~$ ls
Desktop   HBASE          hbase-2.5.5-bin.tar.gz.1  Public    Videos
Documents hbase-2.5.5      Music                  snap
Downloads hbase-2.5.5-bin.tar.gz  Pictures            Templates
prasanna@prasannav:~$ tar xvf hbase-2.5.5-bin.tar.gz
hbase-2.5.5/LICENSE.txt
hbase-2.5.5/NOTICE.txt
hbase-2.5.5/LEGAL
hbase-2.5.5/docs/
hbase-2.5.5/docs/apidocs/
hbase-2.5.5/docs/apidocs/org/
hbase-2.5.5/docs/apidocs/org/apache/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/backup/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/chaos/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/chaos/class-use/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/class-use/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/client/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/client/backoff/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/client/backoff/class-use/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/client/class-use/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/client/example/
hbase-2.5.5/docs/apidocs/org/apache/hadoop/hbase/client/locking/
```

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

```
prasanna@prasannav:~$ ls
Desktop   HBASE          hbase-2.5.5-bin.tar.gz.1  Public    Videos
Documents hbase-2.5.5      Music                  snap
Downloads hbase-2.5.5-bin.tar.gz  Pictures            Templates
prasanna@prasannav:~$ cd hbase-2.5.5/
prasanna@prasannav:~/hbase-2.5.5$ cd conf/
prasanna@prasannav:~/hbase-2.5.5/conf$ ls
hadoop-metrics2-hbase.properties  hbase-policy.xml      log4j2.properties
hbase-env.cmd                      hbase-site.xml       regionservers
hbase-env.sh                        log4j2-hbttop.properties
prasanna@prasannav:~/hbase-2.5.5/conf$ gedit hbase-env.sh
```

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
```



```
Open ▾  hbase-env.sh  Save  -  x
1 #!/usr/bin/env bash
2 #
3 #/**
4 # * Licensed to the Apache Software Foundation (ASF) under one
5 # * or more contributor license agreements. See the NOTICE file
6 # * distributed with this work for additional information
7 # * regarding copyright ownership. The ASF licenses this file
8 # * to you under the Apache License, Version 2.0 (the
9 # * "License"); you may not use this file except in compliance
10 # * with the License. You may obtain a copy of the License at
11 # *
12 # *      http://www.apache.org/licenses/LICENSE-2.0
13 # *
14 # * Unless required by applicable law or agreed to in writing, software
15 # * distributed under the License is distributed on an "AS IS" BASIS,
16 # * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
17 # * See the License for the specific language governing permissions and
18 # * limitations under the License.
19 # */
20
21 # Set environment variables here.
22
23 # This script sets variables multiple times over the course of starting an hbase process,
24 # so try to keep things idempotent unless you want to take an even deeper look
25 # into the startup scripts (bin/hbase, etc.)
26
27 # The java implementation to use. Java 1.8+ required.
28 export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
29
30 # Extra Java CLASSPATH elements. Optional.
31 # export HBASE_CLASSPATH=
32
33 # The maximum amount of heap to use. Default is left to JVM default.
34 # export HBASE_HEAPSIZE=1G
35
36 # Uncomment below if you intend to use off heap cache. For example, to allocate 8G of
37 # offheap, set the value to "8G".
```

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-2.5.5
```

here you can change name 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 folderin home directory before setting HBASE_HOME path , if not then move the hbase-2.5.5 file to home directory*

The screenshot shows a terminal window titled "prasanna@prasannav: ~". It displays the contents of the .bashrc file in a nano editor. The file includes comments about bash aliases and programmable completion features, followed by configuration for HBase. The bottom of the screen shows the nano editor's command bar with various keyboard shortcuts.

```
GNU nano 6.2          .bashrc
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

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

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
export HBASE_HOME=/home/prasanna/hbase-2.5.5
export PATH= $PATH:$HBASE_HOME/bin
```

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

The screenshot shows a terminal window with the command "gedit hbase-site.xml" run in the directory "hbase-2.5.5/conf". The terminal prompt is "prasanna@prasannav:~/hbase-2.5.5/conf\$".

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

```
<property>
<name>hbase.rootdir</name>
<value>file:///home/prasanna/HBASE/hbase</value>
</property>
<property>
<name>hbase.zookeeper.property.dataDir</name>
<value>/home/prasanna/HBASE/zookeeper</value>
</property>
```

step-7: Goto To /etc/ folder and run the following command and configure

The screenshot shows a terminal window with the command "gedit hosts" run in the directory "/etc". The terminal prompt is "prasanna@prasannav:/etc\$".

```

hosts [Read-Only] /etc
1 127.0.0.1      localhost
2 127.0.0.1      prasannav.myguest.virtualbox.org      prasannav
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1             ip6-localhost ip6-loopback
6 ::1             localhost
7 ff00::0          ip6-mcastprefix
8 ff02::1          ip6-allnodes
9 ff02::2          ip6-allrouters

```

Plain Text Tab Width: 8 Ln 2, Col 9 INS

change in line no-2 by default the ip is 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

```

prasanna@prasannav:~$ cd hbase-2.5.5/bin/
prasanna@prasannav:~/hbase-2.5.5/bin$ ./start-hbase.sh
running master, logging to /home/prasanna/hbase-2.5.5/logs/hbase-prasanna-master-prasannav.out
prasanna@prasannav:~/hbase-2.5.5/bin$ 

```

After this run jps command to ensure that hbase is running

```

prasanna@prasannav:~/hbase-2.5.5/bin$ jps
5126 Jps
4729 HMaster
prasanna@prasannav:~/hbase-2.5.5/bin$ 

```

run <http://localhost:16010> to see hbase web UI

Base Stats	Memory	Requests	Storefiles	Compactions	Replications
ServerName	◆ Start time	◆ Last contact	◆ Version	◆ Requests Per Second	◆ Num. Regions
localhost,16020,1697107453860	Thu Oct 12 16:14:13 IST 2023	2 s	2.5.5	0	4
Total:1				0	4

Region Servers

ServerName	◆ Start time	◆ Last contact	◆ Version	◆ Requests Per Second	◆ Num. Regions
localhost,16020,1697107453860	Thu Oct 12 16:14:13 IST 2023	2 s	2.5.5	0	4
Total:1				0	4

Backup Masters

ServerName	Port	Start Time
Total:0		

Tables

User Tables	System Tables	Snapshots					
2 table(s) in set. [Details] . Click count below to see list of regions currently in 'state' designated by the column title. For 'Other' Region state, browse to hbase:meta and adjust filter on 'Meta Entries' to query on states other than those listed here. Queries may take a while if the hbase:meta table is large.							
Namespace	Name	State					
OPEN	OPENING	CLOSED	CLOSING	OFFLINE	SPLIT	Other	Description
default	p	ENABLED	1	0	0	0	{'p': {TABLE_ATTRIBUTES => {METADATA => {'hbase.store.file-tracker.impl' => 'DEFAULT'}}}, {NAME => 'c'}}

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

```
prasanna@prasannav:~/hbase-2.5.5/bin$ ./hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.5.5, r7ebd4381261fef78fc2acf258a95184f4147cee, Thu Jun  1 17:42:49 PDT 2023
Took 0.0011 seconds
hbase:001:0> █
```

Result:

HBase was successfully installed on Ubuntu 18.04.

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

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:postOperationResult(3591)) - Operation: CREATE, Table Name: default:aamec, procId: 100 completed
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

syntax:

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

here **row_key** is a unique key to retrive data

code :

this data will enter data into the dept column family

```
put 'aamec','cse','dept:studentname','prasanna'
put 'aamec','cse','dept:year','third'
put 'aamec','cse','dept:section','A'
```

```
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>
```

This data will enter data into the year column family

```
put 'aamec','cse','year:joinedyear','2021'  
put 'aamec','cse','year:finishingyear','2025'
```

```
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>
```

4) Scan Table

same as desc in RDBMS

syntax:

```
scan 'table_name'
```

code:

```
scan 'aamec'
```

```
hbase:022:0> scan 'aamec'  
ROW COLUMN+CELL  
cse column=dept:section, timestamp=2023-10-13T12:14:26.734, value=A  
cse column=dept:studentname, timestamp=2023-10-13T12:13:11.914, value=pras  
anna  
cse column=dept:year, timestamp=2023-10-13T12:13:41.018, value=third  
cse column=year:finishingyear, timestamp=2023-10-13T12:16:57.291, value=20  
25  
cse column=year:joinedyear, timestamp=2023-10-13T12:16:41.876, value=2021  
it column=dept:section, timestamp=2023-10-13T12:20:40.016, value=A  
it column=dept:studentname, timestamp=2023-10-13T12:20:06.012, value=amba  
lavan  
it column=dept:year, timestamp=2023-10-13T12:20:19.978, value=third  
it column=year:finishingyear, timestamp=2023-10-13T12:21:02.654, value=20  
25  
it column=year:joinedyear, timestamp=2023-10-13T12:20:53.094, value=2021  
2 row(s)  
Took 0.0810 seconds  
hbase:023:0>
```

5) To get specific data

syntax:

```
get 'table_name','row_key',[optional column family: attribute]
```

code :

```
get 'aamec','cse'
```

```

hbase:023:0> get 'aamec','cse'
COLUMN                                CELL
dept:section                           timestamp=2023-10-13T12:14:26.734, value=A
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 0.0908 seconds

```

6.update table value

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

```

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

```

previously the value for the section in cse is A ,But after running this command the value will be changed into B

```

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>

```

7)To Delete Data

syntax:

delete 'table_name','row_key','column_family:attribute'

code :

delete 'aamec','cse','year:joinedyear'

```

Took 0.0138 seconds
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> 

```

8.Delete Table

first we need to disable the table before dropping it

To Disable:

syntax:

```
disable 'table_name'
```

code:

```
disable 'aamec'
```

```
Took 0.0088 seconds
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>
```

Result:

HBase was successfully installed with an example on Ubuntu 18.04.

EXPT.NO.7	
DATE:	Practice importing and exporting data from various databases.

To import or export, the order of columns in MySQL and Hive

Pre-requisite

Hadoop and Java

MySQL

Hive

SQOOP

Step 1: To start hdfs

```
ambal2@Ubuntu:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as ambal2 in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
```

Step 2: MySQL Installation

sudo apt install mysql-server (use this command to install MySQL server)

```
root@Ubuntu:/home/ambal2# mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.34-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

COMMANDS:

~\$ sudo su

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 'bigdata'@'localhost' IDENTIFIED BY 'bigdata';
```

```
Mysql>grant all privileges on *.* to bigdata@localhost;
```

Note: This step is not required if you just use the root user to make CRUD operations in the MySQL

```
Mysql> CREATE USER 'bigdata'@'127.0.0.1' IDENTIFIED BY ' bigdata';
```

```
Mysql>grant all privileges on *.* 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

Step 3: Create a database and table and insert data.

Example:

```
create database Employe;
```

```
create table Employe.Emp(author_name varchar(65), total_no_of_articles int, phone_no int, address varchar(65));
```

```
insert into Emp values("Rohan",10,123456789,"Lucknow");
```

Step 3: Create a database and table in the hive where data should be imported.

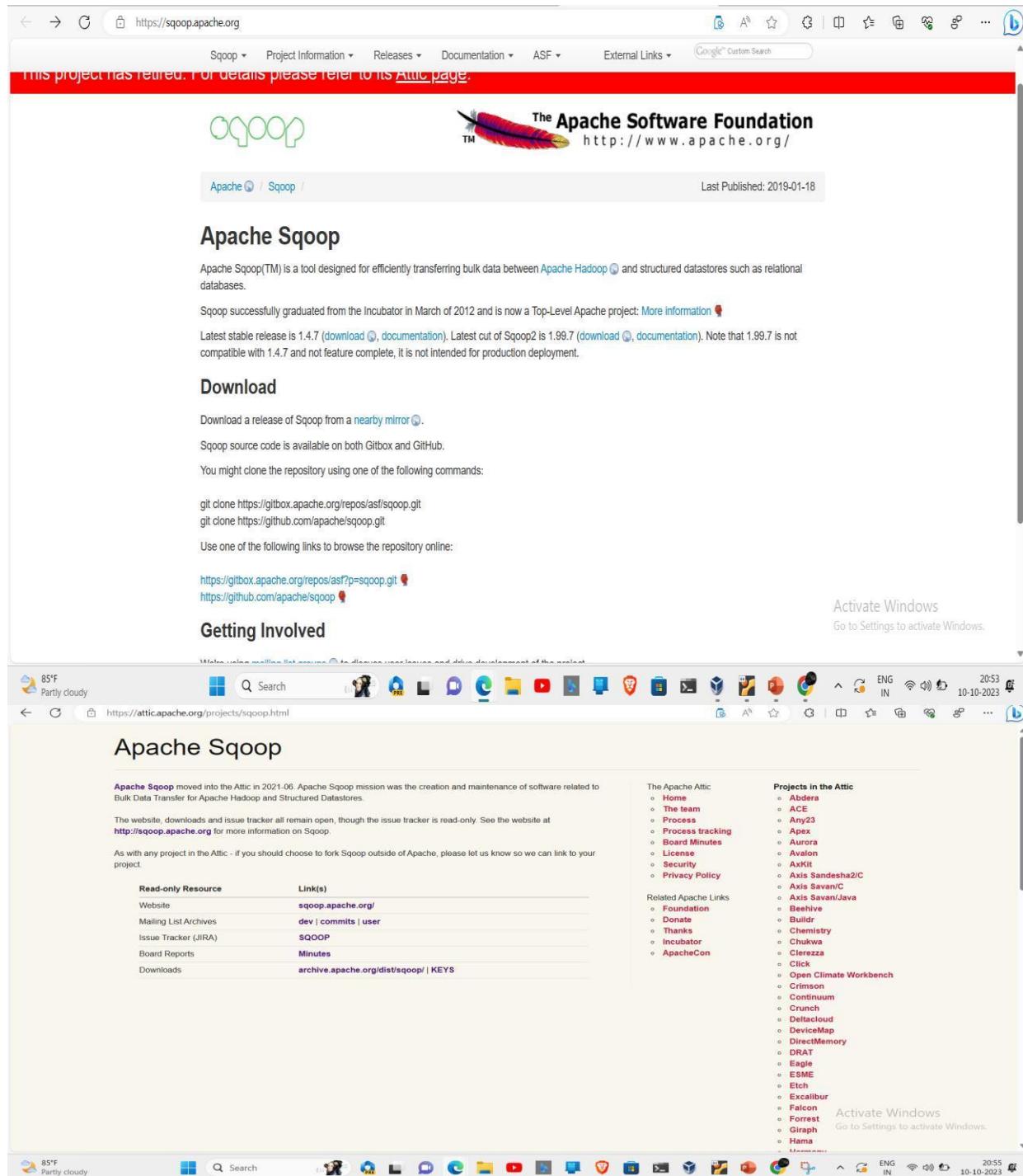
```
create table geeks_hive_table(name string, total_articles int, phone_no int, address string)
row format delimited fields terminated by ',';
```

```
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: SQUOOP INSTALLATION :



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
```

Then enter into the super user : \$ su

Next to move that to the usr/lib which requires a super user privilege

```
$ mv sqoop-1.4.4-bin__hadoop-2.0.4-alpha /usr/lib/sqoop
```

Then exit : \$ exit

Goto .bashrc: \$ sudo nano .bashrc , and then add the following

```
export SQUIOP_HOME=/usr/lib/sqoop
```

```
export PATH=$PATH:$SQUIOP_HOME/bin
```

\$ source ~/.bashrc

Then configure the sqoop, goto the directory of the config folder of sqoop_home and then move the contents of template file to the environment file.

```
$ cd $SQUIOP_HOME/conf
```

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

Then open the sqoop-environment file and then add the following,

```
export HADOOP_COMMON_HOME=/usr/local/Hadoop
```

```
export 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 on your installation.

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

We can download mysql-connector-java-5.1.30.tar.gz file from the following [link](#).

Next, to extract the file and place it to the lib folder of sqoop

```
$ 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

Step 3: hive database Creation

```
hive> create database sqoop_example;
```

```
hive>use sqoop_example;
```

```
hive>create table sqoop(usr_name string,no_ops int,ops_names string);
```

Hive commands much more alike mysql commands. Here, we just create the structure to store the data which we want to import in hive.

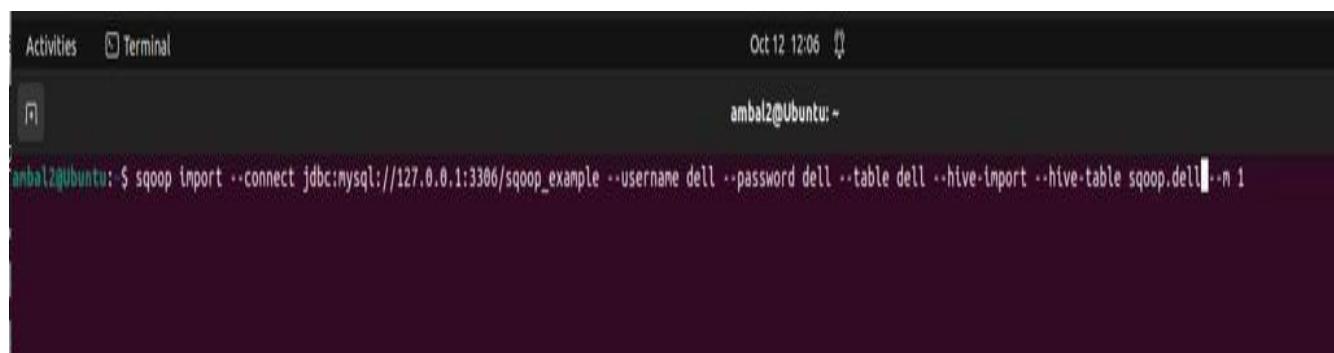
```
ambal2@Ubuntu: ~$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/ambal2/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/ambal2/hadoop-3.2.3/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.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
scoop
Time taken: 0.683 seconds, Fetched: 2 row(s)
hive> use scoop;
OK
Time taken: 0.08 seconds
hive> show tables;
OK
bigdata
scoop
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 \
--username root --password cloudera \
--table table_name_in_mysql \
--hive-import --hive-table database_name_in_hive.table_name_in_hive \
--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
isched 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:

To Install and Create database and Collection CRUD Document: Insert, Query, Update and Delete Document.

PROCEDURE:

Install Mongo DB in Windows

The website of MongoDB provides all the installation instructions, and MongoDB is supported by Windows, Linux as well as Mac OS.

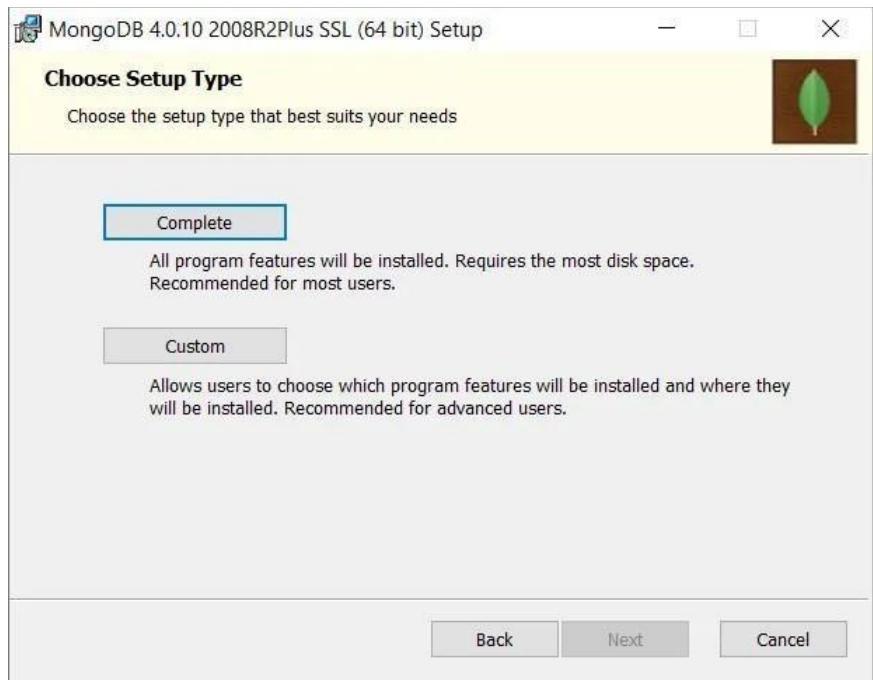
It is to be noted that, MongoDB will not run in Windows XP; so you need to install higher versions of windows to use this database.

Once you visit the link (<http://www.mongodb.org/downloads>), click the download button.

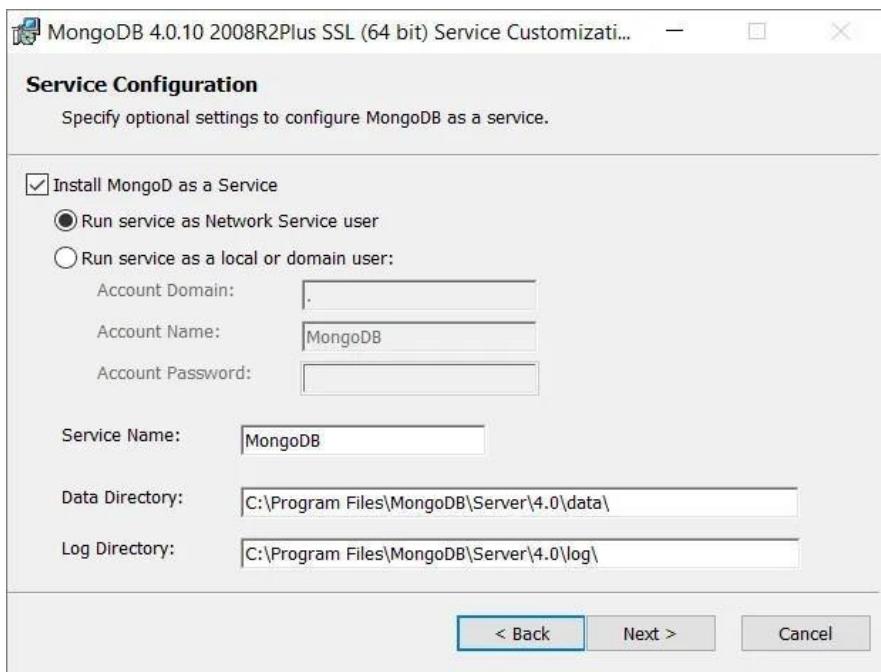
- Once the download is complete, double click this setup file to install it.

- Follow the steps:

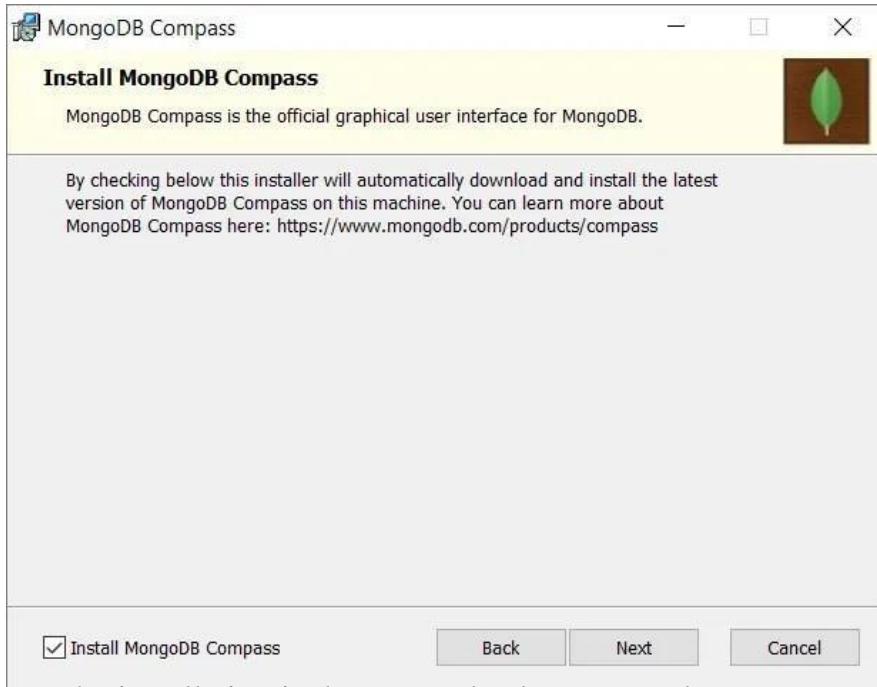




4. Then, select the radio button "Run services as Network service user."



5. The setup system will also prompt you to install MongoDB Compass, which is MongoDB 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 MongoDB 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 MongoDB primary client system, you have to use the command:

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

Result:

Downloading and Installation and Creation of database and Collection CRUD Document: Insert, Query, Update and Delete Document has been successfully completed.

EXPT.NO.9	Create -file, data in memory, other RDD. Lazy Execution, Persisti RDD , RDD, Actions and Transformation on 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 from Seq or List (using Parallelize)
2. Creating an RDD from a text file
3. Creating from another RDD
4. Creating from existing DataFrames and DataSet

Spark Create RDD from Seq or List (using Parallelize)

RDD's are generally created by parallelized collection i.e. by taking an existing collection from driver program (scala, python e.t.c) and passing it to SparkContext's parallelize() method. This method is used only for testingbut not in realtime as the entire data will reside on one node which is not ideal for production.

```
val rdd=spark.sparkContext.parallelize(Seq(("Java", 20000),
("Python", 100000), ("Scala", 3000)))
```

```
rdd.foreach(println
```

n)Outputs:

```
(Python,100000)
```

```
(Scala,3000)
```

```
(Java,20000)
```

- Mostly for production systems, we create RDD's from files. here will see how to create an RDD by reading data from a file.

```
val rdd = spark.sparkContext.textFile("/path/textFile.txt")
```

This creates an RDD for which each record represents a line in a file.

If you want to read the entire content of a file as a single record usewholeTextFiles() method on sparkContext.

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

In this case, each text file is a single record. In this, the name of the file is the first column and the value of the text file is the second column.

Creating from another RDD

You can use transformations like map, flatmap, filter to create a new RDDfrom an existing one.

```
val rdd3 = rdd.map(row=>{(row._1,row._2+100)})
```

Above, creates a new RDD “rdd3” by adding 100 to each record on RDD. this example outputs below.

(Python,100100)

(Scala,3100)

(Java,20100)

From existing DataFrames and DataSet

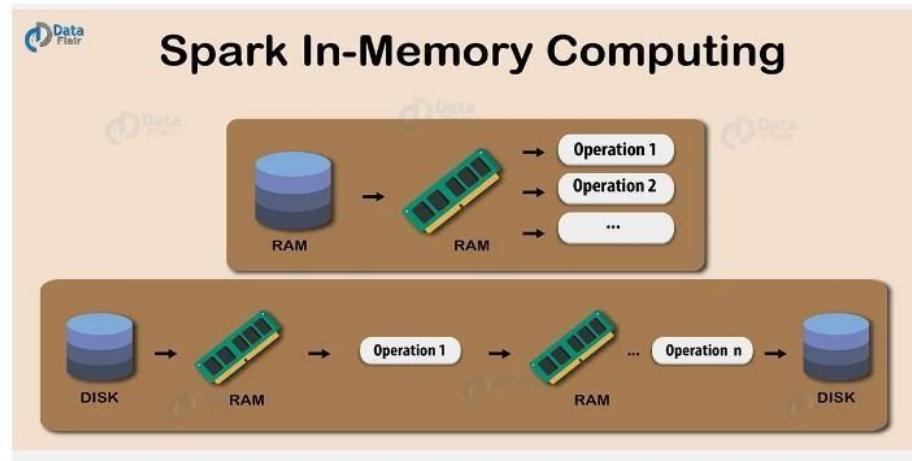
To convert DataSet or DataFrame to RDD just use rdd() method on any ofthese data types.

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

toDF() creates a DataFrame and by calling rdd on DataFrame returns backRDD.

SPARK IN-MEMORY COMPUTING

The data is kept in random access memory(RAM) instead of some slow disk drives and is processed in parallel. Using this we can detect a pattern, analyze large data. This has become popular because it reduces the cost of memory. So, in-memory processing is economic for applications. The two main columns of in-memory computation are- Parallel distributed processing.



Spark In-Memory Computing

Keeping the data in-memory improves the performance by an order of magnitudes. The main abstraction of Spark is its RDDs. And the RDDs are cached using the `cache()` or `persist()` method.

When we use `cache()` method, all the RDD stores in-memory. When RDD stores the value in memory, the data that does not fit in memory is either recalculated or the excess data is sent to disk. Whenever we want RDD, it can be extracted without going to disk. This reduces the space-time complexity and overhead of disk storage.

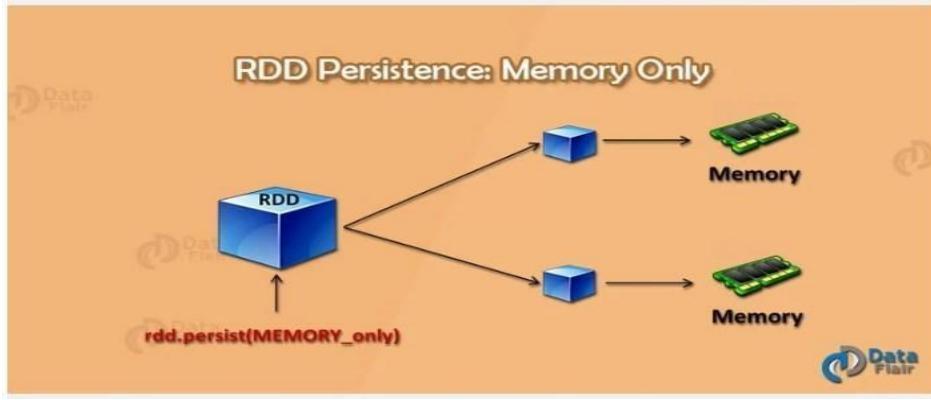
The in-memory capability of Spark is good for machine learning and micro-batch processing. It provides faster execution for iterative jobs.

When we use `persist()` method the RDDs can also be stored in-memory, we can use it across parallel operations. The difference between `cache()` and `persist()` is that using `cache()` the default storage level is `MEMORY_ONLY` while using `persist()` we can use various storage levels.

Storage levels of RDD Persist() in Spark

The various storage levels of `persist()` method in Apache Spark RDD are:

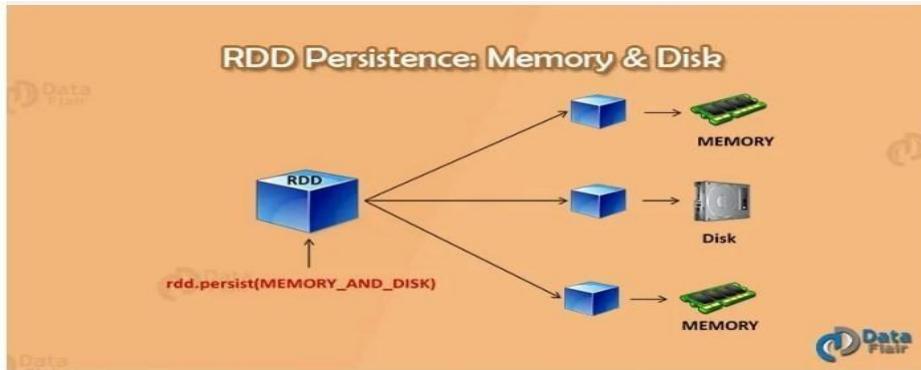
- ❖ `MEMORY_ONLY`
- ❖ `MEMORY_AND_DISK`
- ❖ `MEMORY_ONLY_SER`
- ❖ `MEMORY_AND_DISK_SER`
- ❖ `DISK_ONLY`
- ❖ `MEMORY_ONLY_2` and `MEMORY_AND_DISK_2`



Spark storage level – memory only

In this storage level Spark, RDD store as deserialized JAVA object in JVM. If RDD does not fit in memory, then the remaining will re-compute each time they are needed.

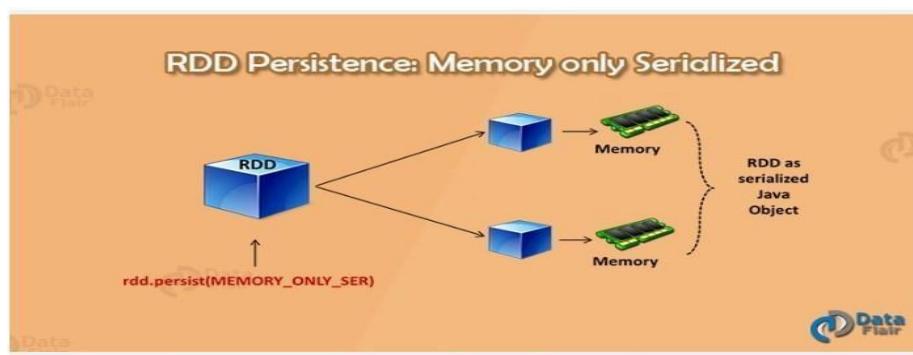
MEMORY AND DISK



Spark storage level-memory and disk

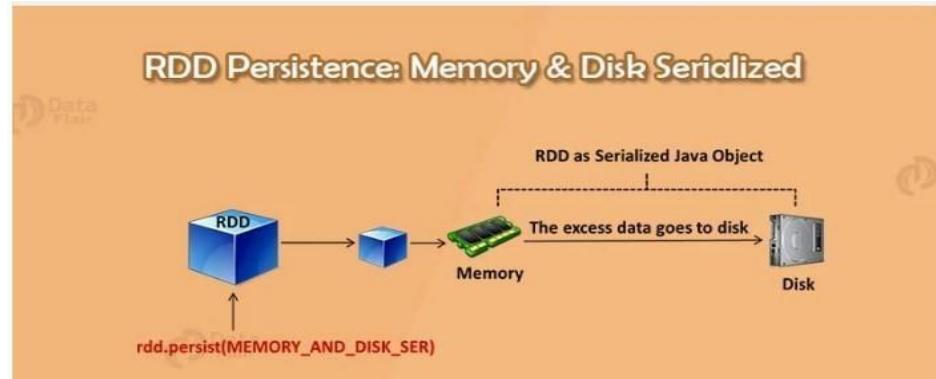
In this level, RDD is stored as deserialized JAVA object in JVM. If the fullRDD does not fit in memory then the remaining partition is stored on disk, instead of recomputing it every time when it is needed.

MEMORY ONLY SER



especially when we use fast serializer.

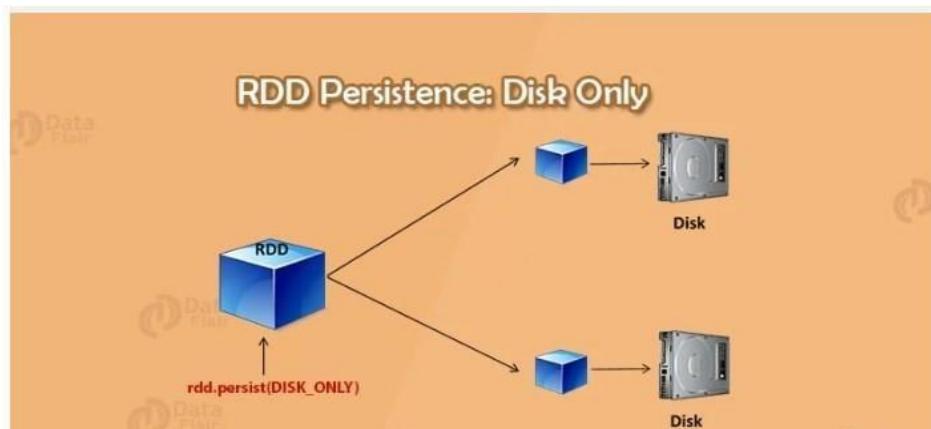
MEMORY AND DISK SER



Spark storage level – memory and disk serialized

This level stores RDD as serialized JAVA object. If the full RDD does not fit in the memory then it stores the remaining partition on the disk, instead of recomputing it every time when we need.

DISK_ONLY



Spark storage level-disk-only

This storage level stores the RDD partitions only on disk.

MEMORY ONLY 2 and MEMORY AND DISK 2

It is like MEMORY_ONLY and MEMORY_AND_DISK. The only difference is that each partition gets replicated on two nodes in the cluster.

Advantages of In-memory Processing

After studying Spark in-memory computing introduction and various storage levels in detail, let's discuss the advantages of in-memory computation

1. can retrieve it easily.
2. It is good for real-time risk management and fraud detection.
3. The data becomes highly accessible.
4. The computation speed of the system increases.
5. Improves complex event processing.
6. Cached a large amount of data.
7. It is economic, as the cost of RAM has fallen over a period of time.

Result:

Thus to Create -file, data in memory, other RDD. Lazy Execution, Persisti RDD , RDD, Actions and Transformation on RDD installed Hadoop has been verified sucessfully

EXPT.NO.10	Create Stories using Tableau ,Connect to data, Build Charts and Analyze Data, Create Dashboard.
DATE:	

AIM:

To Create Stories using Tableau ,Connect to data, Build Charts and Analyze Data, Create Dashboard.

PROCEDURE:

What is Custom SQL Query?

Tableau queries each data source using SQL that's specific to the data type. Tableau allows the SQL used to query a data source to be customized in order to manipulate the joins, filters, and field lengths and types produce a more accurate output.

Sample SQL query:

```
SELECT ['us states data 1$].[State] AS [State], ['us states data
1$].[Population] AS [Population],[us states data 1$].[Region] AS
[Region]

FROM ['us states data 1$]UNION

SELECT ['us states data 2$].[State] AS [State], ['us states data
2$].[Population] AS [Population],[us states data 2$].[Region] AS
[Region]
```

```
FROM ['us states data 2$]UNION
```

```
SELECT ['us states data 3$].[State] AS [State], ['us states data
3$].[Population] AS [Population],
```

UNION

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

FROM ['us states data

Steps to visualize usir

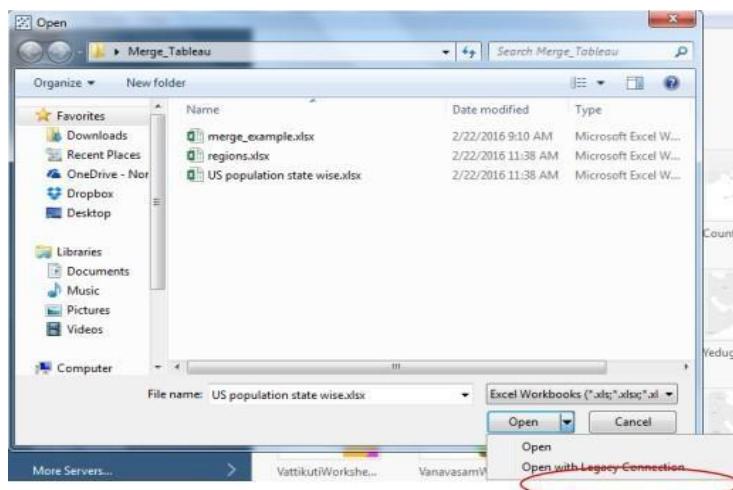


1. The following excel workbook consists of 4 sheets consisting of US population state wise.

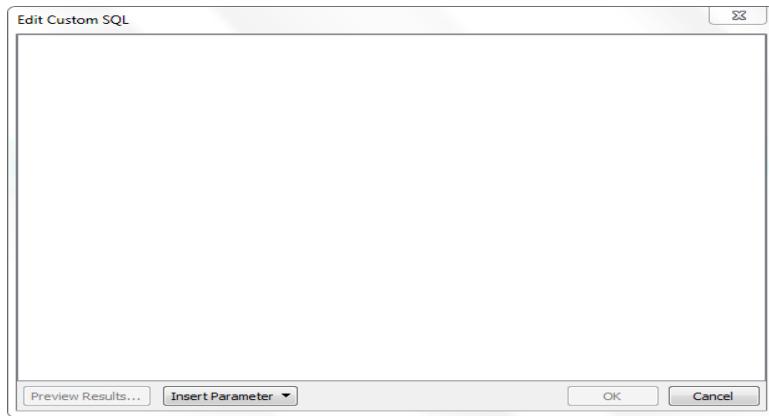


US population state wise.xlsx

2. Open Tableau and connect the Excel data as Open with Legacy Connection.



3. Now your Tableau window looks like this with an additional sheet named as New Custom SQL.



5. Now we need to write SQL query to perform union operation on the excel sheets and generate a combined sheet.

Ex:

```
SELECT [Sheet1$].[ID] AS [ID],  
[Sheet1$].[Type] AS  
[Type], [Sheet1$].[Value]  
AS [Value]FROM  
[Sheet1$]  
UNION ALL  
SELECT [Sheet2$].[ID] AS [ID],  
[Sheet2$].[Type] AS  
[Type], [Sheet2$].[Value]  
AS [Value]FROM  
[Sheet2$]
```

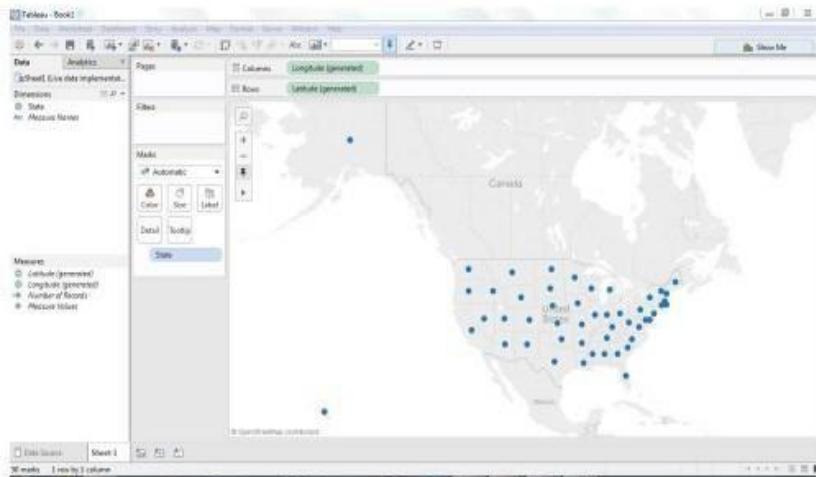
The above SQL query performs union operation on two sheets namely Sheet1 and Sheet2 with ID, Type, and Value as their columns.

6. After writing the query click on Preview Results on the Edit Custom SQL window pane to check whether the query worked fine or not.
7. The resulting data should contain all the states, population and region name to which it belong to.

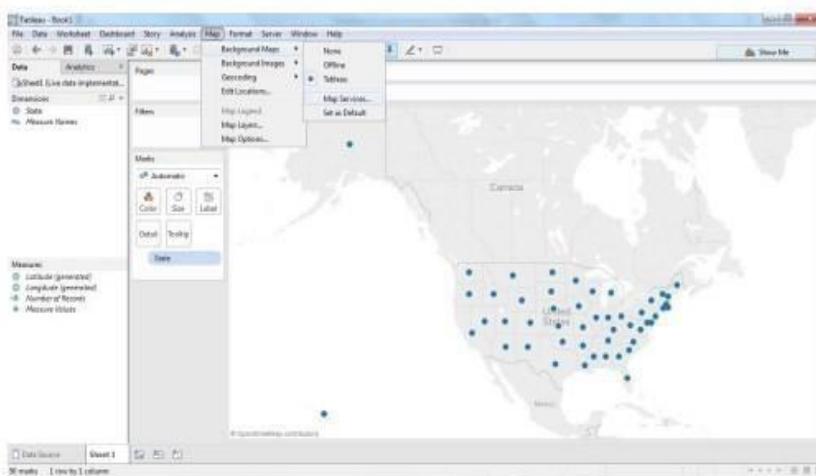
8. Click on the Tableau sheet to visualize the data.
9. Now you will see State, Region as Dimensions and Population as Measures.

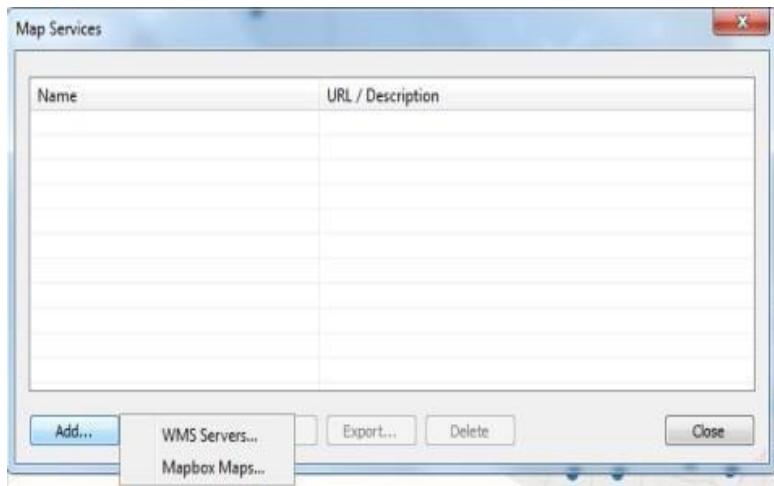
Live Data or Real-time data (RTD) denotes information that is delivered immediately after collection. There is no delay in the timeliness of the information provided. Real-time data is often used for navigation or tracking. A Web Map Service (WMS) is a standard protocol for serving georeferenced map images which a map server generates using data from a GIS database. A Web Map Service (WMS) produces an image (e.g. GIF, JPG) of geospatial data. Steps to implement live map using Tableau:

1. Open Tableau and connect the Excel data.
2. Drag and drop state Dimension in Detail menu provided in Marks window pane.
3. Now you can see Symbol map representing the states of United States. It looks as shown below



4. In this we will be using OGCWMS server to get the live geospatial data.
5. Click on Map item in the menu bar, then select Background Maps and select WMS Server as shown

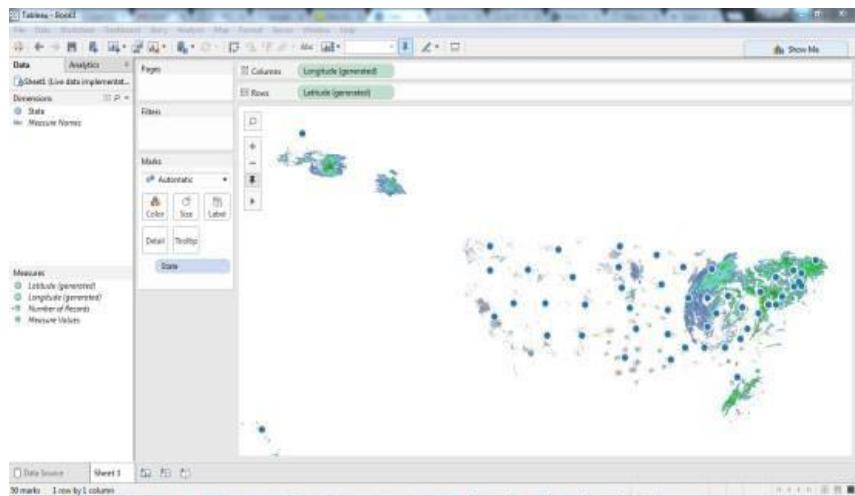




7. Click on Add and select WMS Servers. After clicking on WMS Servers a window pane as shown below will be prompted to enter the URL of the server.
http://nowcoast.noaa.gov/arcgis/services/nowcoast/radar_meteo_imagery_nexrad_time/MapServer/WMServer?

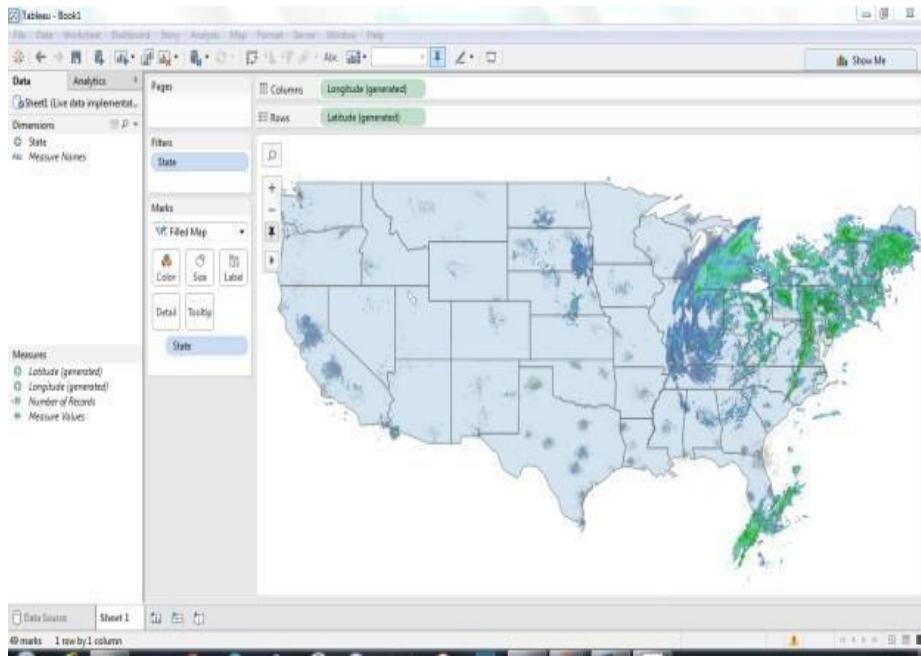


8. Now click on OK and then close the WMS server window pane and a map can be seen as shown



10. Change the color transparency to less than 20% so that the required map is visible.

11. Then required map looks like this



Result:

To Create Stories using Tableau ,Connect to data, Build Charts and Analyze Data, Create Dashboard has been verified successfully.

VIVA QUESTIONS

VIVA QUESTIONS AND ANSWERS

1. What is the port number for NameNode, Task Tracker and Job Tracker?

- NameNode 50070
- Job Tracker 50030
- Task Tracker 50060

2. Explain about the process of inter cluster data copying.

HDFS provides a distributed data copying facility through the DistCP from source to destination. If this data copying is within the hadoop cluster then it is referred to as inter cluster data copying. DistCP requires both source and destination to have a compatible or same version of hadoop.

3. How can you overwrite the replication factors in HDFS?

The replication factor in HDFS can be modified or overwritten in 2 ways-

1)Using the Hadoop FS Shell, replication factor can be changed per file basis using the below command-

\$Shadoop fs –setrep –w 2 /my/test_file (test_file is the filename whose replication factor will be set to 2)

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

3)\$hadoop fs –setrep –w 5 /my/test_dir (test_dir is the name of the directory and all the files in this directory will have a replication factor set to 5)

4. Explain the difference between NAS and HDFS.

NAS runs on a single machine and thus there is no probability of data redundancy whereas HDFS runs on a cluster of different machines thus there is data redundancy because of the replication protocol.

NAS stores data on a dedicated hardware whereas in HDFS all the data blocks are distributed across local drives of the machines.

In NAS data is stored independent of the computation and hence Hadoop MapReduce cannot be used for processing whereas HDFS works with Hadoop MapReduce as the computations in HDFS are moved to data.

5. Explain what happens if during the PUT operation, HDFS block is assigned a replication factor 1 instead of the default value 3.

Replication factor is a property of HDFS that can be set accordingly for the entire cluster to adjust the number of times the blocks are to be replicated to ensure high data availability. For every block that is stored in HDFS the cluster will have n-1 duplicated blocks. So, if the replication factor during the PUT operation is set to 1 instead of the default value 3, then it will have a single copy of data. Under these circumstances when the replication factor is set to 1 ,if the DataNode crashes under any circumstances, then only single copy of the data would be lost.

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

HDFS does not support modifications at arbitrary offsets in the file or multiple writers but files are written by a single writer in append only format i.e. writes to a file in HDFS are always made at the end of the file.

7. Explain about the indexing process in HDFS.

Indexing process in HDFS depends on the block size. HDFS stores the last part of the data that further points to the address where the next part of data chunk is stored.

8. What is a rack awareness and on what basis is data stored in a rack?

All the data nodes put together form a storage area i.e. the physical location of the data nodes is referred to as Rack in HDFS. The rack information i.e. the rack id of each data node is acquired by the NameNode. The process of selecting closer data nodes depending on the rack information is known as Rack Awareness.

The contents present in the file are divided into data block as soon as the client is ready to load the file into the hadoop cluster. After consulting with the NameNode, client allocates 3 data nodes for each data block. For

each data block, there exists 2 copies in one rack and the third copy is present in another rack. This is generally referred to as the Replica Placement Policy.

9. What happens to a NameNode that has no data?

There does not exist any NameNode without data. If it is a NameNode then it should have some sort of data in it.

10. What happens when a user submits a Hadoop job when the NameNode is down- does the job get in to hold or does it fail.

The Hadoop job fails when the NameNode is down.

11. What happens when a user submits a Hadoop job when the Job Tracker is down- does the job get in to hold or does it fail.

The Hadoop job fails when the Job Tracker is down.

12. Whenever a client submits a hadoop job, who receives it?

NameNode receives the Hadoop job which then looks for the data requested by the client and provides the block information. JobTracker takes care of resource allocation of the hadoop job to ensure timely completion.

13. What is a heartbeat in HDFS?

Heartbeats in HDFS are the signals that are sent by DataNodes to the NameNode to indicate that it is functioning properly (alive). By default, the heartbeat interval is 3 seconds, which can be configured using `dfs.heartbeat.interval` in `hdfs-site.xml`.

14. How would you check whether your NameNode is working or not?

There are many ways to check the status of the NameNode. Most commonly, one uses the `jps` command to check the status of all the daemons running in the HDFS. Alternatively, one can visit the NameNode's Web UI for the same.

15. What is a block?

You should begin the answer with a general definition of a block. Then, you should explain in brief about the blocks present in HDFS and also mention their default size.

Blocks are the smallest continuous location on your hard drive where data is stored. HDFS stores each file as blocks, and distribute it across the Hadoop cluster. The default size of a block in HDFS is 128 MB (Hadoop 2.x) and 64 MB (Hadoop 1.x) which is much larger as compared to the Linux system where the block size is 4KB. The reason of having this huge block size is to minimize the cost of seek and reduce the meta data information generated per block.

16. Suppose there is file of size 514 MB stored in HDFS (Hadoop 2.x) using default block size configuration and default replication factor. Then, how many blocks will be created in total and what will be the size of each block?

Default block size in Hadoop 2.x is 128 MB. So, a file of size 514 MB will be divided into 5 blocks (514 MB/128 MB) where the first four blocks will be of 128 MB and the last block will be of 2 MB only. Since, we are using the default replication factor i.e. 3, each block will be replicated thrice. Therefore, we will have 15 blocks in total where 12 blocks will be of size 128 MB each and 3 blocks of size 2 MB each.

17. How to copy a file into HDFS with a different block size to that of existing block size configuration?

•Tip: You should start the answer with the command for changing the block size and then, you should explain the whole procedure with an example. This is how you should answer this question:

Yes, one can copy a file into HDFS with a different block size by using '`-Ddfs.blocksize=block_size`' where the `block_size` is specified in Bytes.

Let me explain it with an example: Suppose, I want to copy a file called `test.txt` of size, say of 120 MB, into the HDFS and I want the block size for this file to be 32 MB (33554432 Bytes) instead of the default (128

MB). So, I would issue the following command:

```
hadoop fs -Ddfs.blocksize=33554432 -copyFromLocal /home/edureka/test.txt /sample_hdfs
```

Now, I can check the HDFS block size associated with this file by:

```
hadoop fs -stat %o /sample_hdfs/test.txt
```

Else, I can also use the NameNode web UI for seeing the HDFS directory.

Tip: You can go through the blog on Hadoop Shell Commands where you will find various Hadoop commands, explained with an example.

18. Can you change the block size of HDFS files?

Yes, I can change the block size of HDFS files by changing the default size parameter present in hdfs-site.xml.

But, I will have to restart the cluster for this property change to take effect.

19. What is JobTracker?

JobTracker is a Hadoop service used for the processing of MapReduce jobs in the cluster. It submits and tracks the jobs to specific nodes having data. Only one JobTracker runs on single Hadoop cluster on its own JVM process. If JobTracker goes down, all the jobs halt.

14. Explain job scheduling through JobTracker.

JobTracker communicates with NameNode to identify data location and submits the work to TaskTracker node. The TaskTracker plays a major role as it notifies the JobTracker for any job failure. It actually is referred to the heartbeat reporter reassuring the JobTracker that it is still alive. Later, the JobTracker is responsible for the actions as in it may either resubmit the job or mark a specific record as unreliable or blacklist it.

20. What is SequenceFileInputFormat?

A compressed binary output file format to read in sequence files and extends the FileInputFormat. It passes data between output-input (between output of one MapReduce job to input of another MapReduce job) phases of MapReduce jobs.

21. How to set mappers and reducers for Hadoop jobs?

Users can configure JobConf variable to set number of mappers and reducers.

```
1 job.setNumMapTasks()
```

```
2 job.setNumReduceTasks()
```

22. Explain JobConf in MapReduce.

It is a primary interface to define a map-reduce job in the Hadoop for job execution. JobConf specifies mapper, Combiner, partitioner, Reducer, InputFormat, OutputFormat implementations and other advanced job facets like Comparators.

These are described in Mapreduce's online reference guide and on Mapreduce community.

23. What is a MapReduce Combiner?

Also known as semi-reducer, Combiner is an optional class to combine the map out records using the same key. The main function of a combiner is to accept inputs from Map Class and pass those key-value pairs to Reducer class.

24. What is RecordReader in a Map Reduce?

RecordReader is used to read key/value pairs from the InputSplit by converting the byte-oriented view and presenting record-oriented view to Mapper.

25. Define Writable data types in MapReduce.

Hadoop reads and writes data in a serialized form in writable interface. The Writable interface has several classes like Text (storing String data), IntWritable, LongWritable, FloatWritable, BooleanWritable. users are free to define their personal Writable classes as well.

26. What are the components used in Hive Query Processor?

Following are the components of a Hive Query Processor:

Parse and Semantic Analysis (ql/parse)

Metadata Layer (ql/metadata)

Type Interfaces (ql/typeinfo)

Sessions (ql/session)

Map/Reduce Execution Engine (ql/exec)

Plan Components (ql/plan)

Hive Function Framework (ql/udf)

Tools (ql/tools)

Optimizer (ql/optimizer)

27. What are Buckets in Hive?

Buckets in Hive are used in segregating Hive table data into multiple files or directories. They are used for efficient querying.

28. What kind of data warehouse application is suitable for Hive? What are the types of tables in Hive?

Hive is not considered a full database. The design rules and regulations of Hadoop and HDFS have put restrictions on what Hive can do. However, Hive is most suitable for data warehouse applications because it:
Analyzes relatively static data

Has less responsive time

Does not make rapid changes in data

Although Hive doesn't provide fundamental features required for Online Transaction Processing (OLTP), it is suitable for data warehouse applications in large datasets. There are two types of tables in Hive:

Managed tables

External tables

29. What is the definition of Hive? What is the present version of Hive? Explain ACID transactions in Hive.

Hive is an open-source data warehouse system. We can use Hive for analyzing and querying large datasets. It's similar to SQL. The present version of Hive is 0.13.1. Hive supports ACID (Atomicity, Consistency, Isolation, and Durability) transactions. ACID transactions are provided at row levels. Following are the options Hive uses to support ACID transactions:

Insert

Delete

Update

30. What is the maximum size of a string data type supported by Hive? Explain how Hive supports binary formats.

The maximum size of a string data type supported by Hive is 2 GB. Hive supports the text file format by default, and it also supports the binary format sequence files, ORC files, Avro data files, and Parquet files.

Sequence file: It is a splittable, compressible, and row-oriented file with a general binary format.

ORC file: Optimized row columnar (ORC) format file is a record-columnar and column-oriented storage file. It divides the table in row split. Each split stores the value of the first row in the first column and follows subsequently.

Avro data file: It is the same as a sequence file that is splittable, compressible, and row-oriented but without the support of schema evolution and multilingual binding.

Parquet file: In Parquet format, along with storing rows of data adjacent to one another, we can also store

column values adjacent to each other such that both horizontally and vertically datasets are partitioned.

31. Give the name of the key components of HBase

The key components of HBase are Zookeeper, RegionServer, Region, Catalog Tables and HBase Master.

32. What is S3?

S3 stands for simple storage service and it is a one of the file system used by hbase.

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

get() method is used to read the data from the table.

34. What is the reason of using HBase?

HBase is used because it provides random read and write operations and it can perform a number of operation per second on a large data sets.

35. In how many modes HBase can run?

There are two run modes of HBase i.e. standalone and distributed.

36. Define the difference between hive and HBase?

HBase is used to support record level operations but hive does not support record level operations.

Compare Hive and HBase: which one is best for your needs? Learn more in our Hive vs Hbase blog now!

37. Define column families?

It is a collection of columns whereas row is a collection of column families.

38. Define standalone mode in HBase?

It is a default mode of HBase. In standalone mode, HBase does not use HDFS—it uses the local filesystem instead and it runs all HBase daemons and a local ZooKeeper in the same JVM process.

39. What is decorating Filters?

It is useful to modify, or extend, the behavior of a filter to gain additional control over the returned data.

40. What is the full form of YCSB?

YCSB stands for Yahoo! Cloud Serving Benchmark.

41. What is the use of YCSB?

It can be used to run comparable workloads against different storage systems.

42. What is the Hive ObjectInspector function?

The Hive ObjectInspector function is an important feature of the Apache Hive data warehouse software used to examine and modify the internal data structures of tables. It is a vital element in the process of executing Hive queries, as it transforms the raw data saved in Hive tables into a format that Hive operators can handle. The ObjectInspector function is responsible for creating a standard interface for Hive operators to access table data, and it supports multiple data types, including user-defined, complex, and primitive types. Hive operators extensively use this function in HiveQL queries, and it is a valuable tool for manipulating data within Hive.