UNIVERSITY COLLEGE OF ENGINEERING NAGERCOIL

(ANNA UNIVERSITY CONSTITUENT COLLEGE)

KONAM, NAGERCOIL – 629 004

RECORD NOTE BOOK

CCS334-BIG DATA ANALYTICS

Register No :

Name :

Year/Semester :

Department :

UNIVERSITY COLLEGE OF ENGINEERING
NAGERCOIL

(ANNA UNIVERSITY CONSTITUENT COLLEGE)

KONAM, NAGERCOIL – 629 004

Register No:

Certified that, this is the bonafide record of work done by ***Mr./Ms.* ………………………………………………………… *of*** *VI Semester in Computer Science and Engineering of this college, in the* ***CCS334-BIG DATA ANALYTICS*** *during academic year 2023-2024 in partial fulfillment of the requirements of the B.E Degree course of the Anna University Chennai.*

Staff-in-charge **Head of the Department**

This record is submitted for the University Practical Examination held on …………………………

Internal Examiner **External Examiner**

INDEX

EX NO:1	**DOWNLOADING AND INSTALLING HADOOP; UNDERSTANDING DIFFERENT HADOOP MODES. STARTUP SCRIPTS, CONFIGURATION FILES.**
DATE:	

AIM:

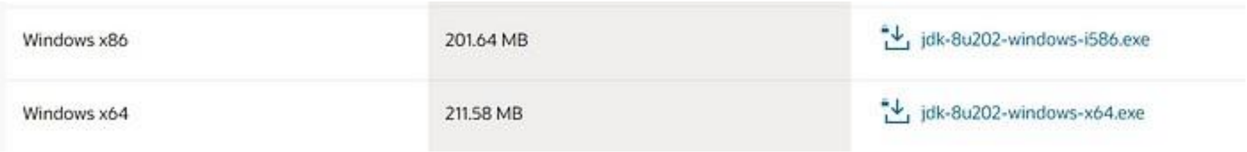
To download and install Hadoop and understand different modes in Hadoop, startup scripts and configuration files.

ALGORITHM:

Step 1: Install Java JDK 1.8

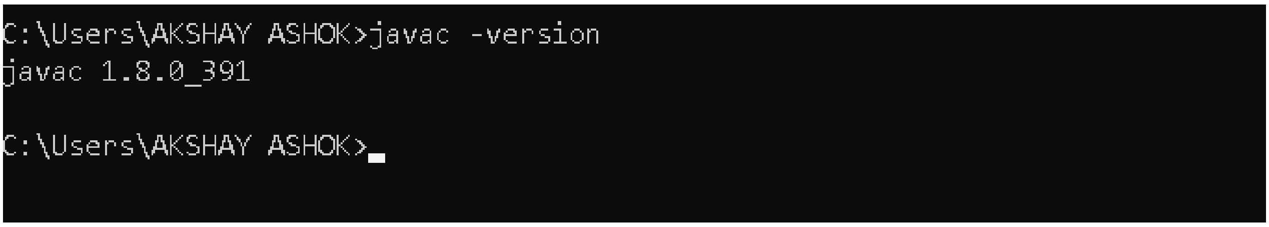
Download the java from the following link

https://www.oracle.com/java/technologies/downloads/#java8



Windows x86	201.64 MB	jdk-8u202-windows-i586.exe
Windows x64	211.58 MB	jdk-8u202-windows-x64.exe

After downloading and installing the Java, go to command prompt and check the Java version

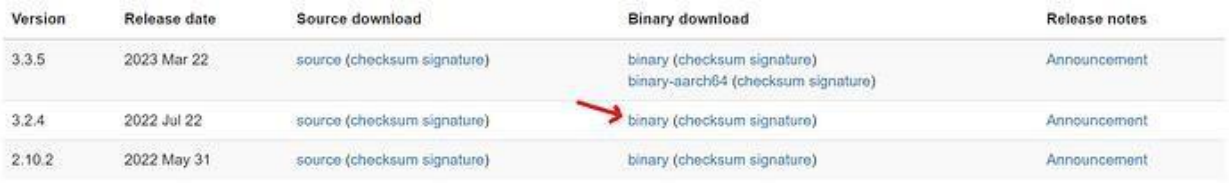


```
C:\Users\AKSHAY ASHOK>javac -version
javac 1.8.0_391

C:\Users\AKSHAY ASHOK>
```

Step 2: Download Hadoop and extract and put the C drive

Open Hadoop Website and download 3.2.4 version.

https://dlcdn.apache.org/hadoop/common/hadoop-3.2.4/hadoop-3.2.4.tar.gz



Version	Release date	Source download	Binary download	Release notes
3.3.5	2023 Mar 22	source (checksum signature)	binary (checksum signature) binary-aarch64 (checksum signature)	Announcement
3.2.4	2022 Jul 22	source (checksum signature)	binary (checksum signature)	Announcement
2.10.2	2022 May 31	source (checksum signature)	binary (checksum signature)	Announcement

After the download copy the zip file into the C drive and extract in here.

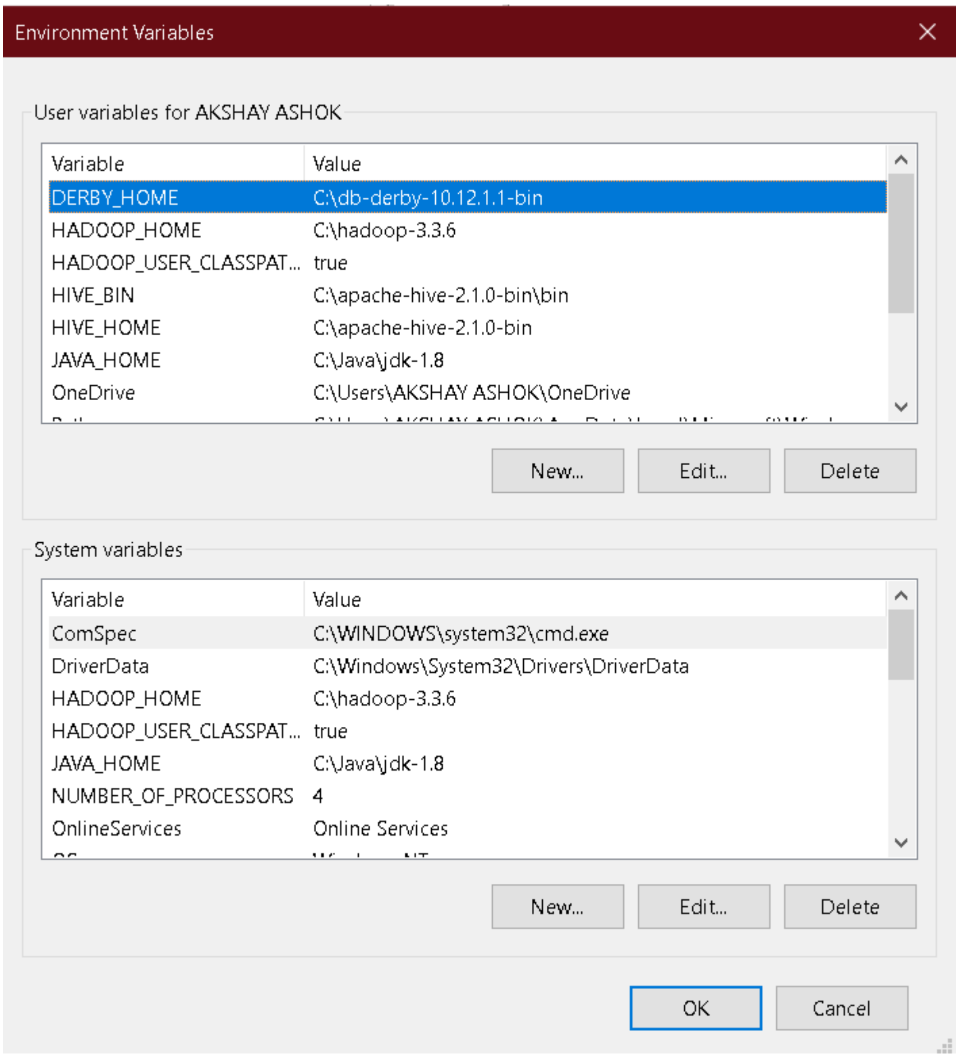
Step 3: Set Path in Environment Variables for Hadoop and Java

Step 3–1:

- Click Advanced System Settings
- Click Environment Variables
- Click User Variables New Button
- Add the Variable Name (HADOOP_HOME) + Variable Value (C:\hadoop-3.2.4)
- Click path into the User Variables and add the path (%HADOOP_HOME%\bin)

***Step 3–2*:**

- Click Advanced System Settings
- Click Environment Variables
- Click User Variables New Button
- Add the Variable Name (JAVA_HOME) + Variable Value (C:\Progra~1\Java\jre1.8)
- Click path into the User Variables and add the path (%JAVA_HOME%\bin)



Environment Variables

User variables for AKSHAY ASHOK

Variable	Value
DERBY_HOME	C:\db-derby-10.12.1.1-bin
HADOOP_HOME	C:\hadoop-3.3.6
HADOOP_USER_CLASSPAT...	true
HIVE_BIN	C:\apache-hive-2.1.0-bin\bin
HIVE_HOME	C:\apache-hive-2.1.0-bin
JAVA_HOME	C:\Java\jdk-1.8
OneDrive	C:\Users\AKSHAY ASHOK\OneDrive

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
HADOOP_HOME	C:\hadoop-3.3.6
HADOOP_USER_CLASSPAT...	true
JAVA_HOME	C:\Java\jdk-1.8
NUMBER_OF_PROCESSORS	4
OnlineServices	Online Services

Step 4: Config files under Hadoop directory

***Step 4–1*:** Configure core-site.xml document

- Enter: **C:\hadoop-3.2.4\etc\hadoop** and right click on to the core-site.xml document and click edit.On notepad document you will be add this code between the and save it.

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

Step 4–2: Configure mapred-site.xml document

- Enter: C:\hadoop-3.2.4\etc\hadoop and right click on to the mapred-site.xml document and clickedit. On notepad document you will be add this code between the and save it:

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

Step 5: Create folder datanode and namenode under data directory

We will create 3 folders:

- Create folder –data‖ under –C:\hadoop-3.2.4‖
- Create folder –datanode‖ under –C:\hadoop-3.2.4\data‖
- Create folder –namenode‖ under –C:\hadoop-3.2.4\data‖

Step 6: Edit HDFS and YARN files

***Step 6–1*:** Configure hdfs-site.xml document

Enter: **C:\hadoop-3.2.4\etc\hadoop** and right click on to the hdfs-site.xml document and click edit.On notepad document you will be add this code between the and save it:

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
         <name>dfs.namenode.name.dir</name>
         <value>hadoop-3.2.4/data/namenode</value>
    </property>
    <property>
          <name>dfs.datanode.data.dir</name>
          <value>hadoop-3.2.4/data/datanode</value>
     </property>
</configuration>
```

***Step 6–2*:** Configure yarn-site.xml document

Enter: **C:\hadoop-3.2.4\etc\hadoop** and right click on to the yarn-site.xml document and click edit.On notepad document you will be add this code between the and save it:

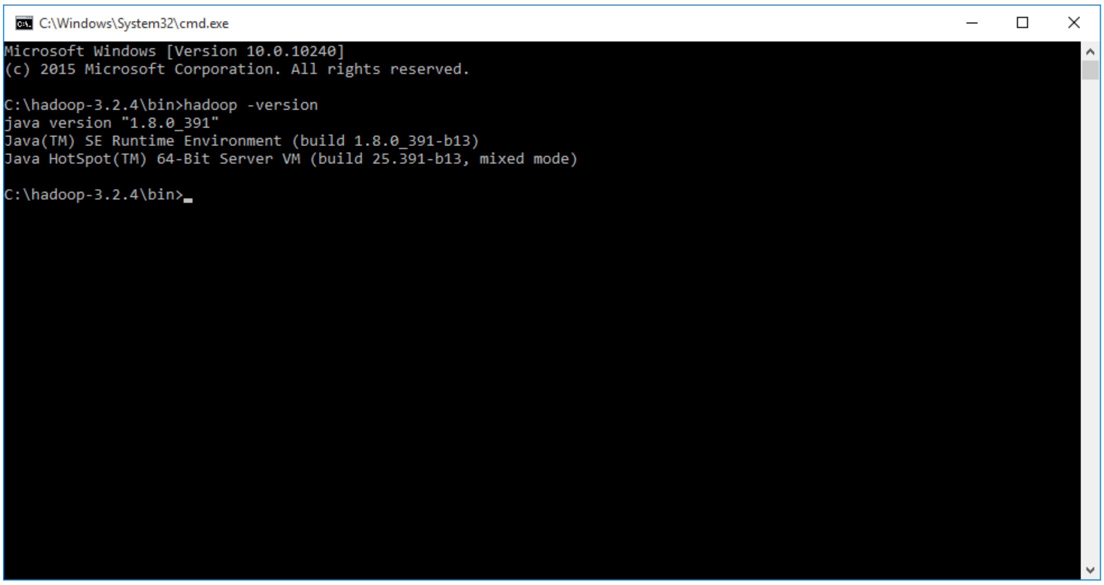
```
<configuration>
   <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
   </property>
   <property>
        <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>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

Step 7: Set Java Home environment in Hadoop environment

***Step 7–1*:** Enter: **C:\hadoop-3.2.4\etc\hadoop** and right click on to the hadoop-env.cmd documentand click edit. On notepad document you will be chane this code structure and save it:

@rem the java implementation to use. Required.
set **JAVA_HOME=C:\Progra~1\Java\jre1.8**

After the configuration if we start to made with **C:\Program Files\Java\jre1.8** for this configuration:
hadoop -version



```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\hadoop-3.2.4\bin>hadoop -version
java version "1.8.0_391"
Java(TM) SE Runtime Environment (build 1.8.0_391-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.391-b13, mixed mode)

C:\hadoop-3.2.4\bin>
```

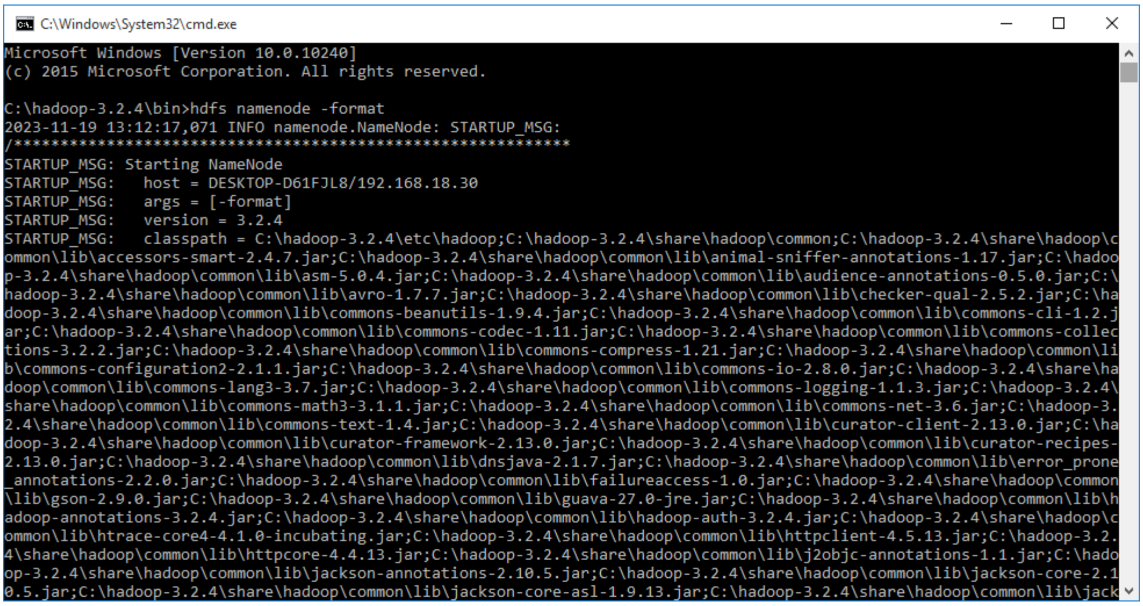
***Step 7–2*:** Enter website: https://github.com/s911415/apache-hadoop-3.1.0-winutils and downloadzip folder.

After the download, extract the zip folder, copy folder bin and replace existing bin folder inC:\hadoop-3.2.4\bin folder. (click replace 4 item and accept)

Step 8: Complete the set up and run to test it (start-all.cmd)

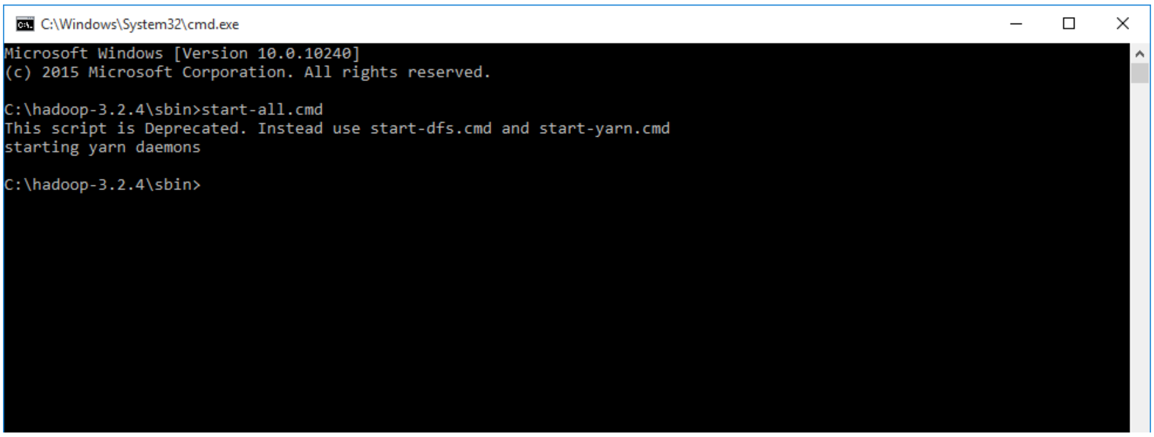
***Step 8–1*:** Enter C:\hadoop-3.2.4\bin when inside the folder, click the path area and write cmd: hdfs name node -format

and when we run the code and see this result.



```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\hadoop-3.2.4\bin>hdfs namenode -format
2023-11-19 13:12:17,071 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = DESKTOP-D61FJL8/192.168.18.30
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.2.4
STARTUP_MSG:   classpath = C:\hadoop-3.2.4\etc\hadoop;C:\hadoop-3.2.4\share\hadoop\common;C:\hadoop-3.2.4\share\hadoop\c
ommon\lib\accessors-smart-2.4.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoo
p-3.2.4\share\hadoop\common\lib\asm-5.0.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\
hadoop-3.2.4\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\ha
doop-3.2.4\share\hadoop\common\lib\commons-beanutils-1.9.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-cli-1.2.j
ar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-codec-1.11.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-collec
tions-3.2.2.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-compress-1.21.jar;C:\hadoop-3.2.4\share\hadoop\common\li
b\commons-configuration2-2.1.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-io-2.8.0.jar;C:\hadoop-3.2.4\share\ha
doop\common\lib\commons-lang3-3.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop-3.2.4\
share\hadoop\common\lib\commons-math3-3.1.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop-3.
2.4\share\hadoop\common\lib\commons-text-1.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\curator-client-2.13.0.jar;C:\ha
doop-3.2.4\share\hadoop\common\lib\curator-framework-2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\curator-recipes-
2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\dnsjava-2.1.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\error_prone
_annotations-2.2.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\failureaccess-1.0.jar;C:\hadoop-3.2.4\share\hadoop\common
\lib\gson-2.9.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\guava-27.0-jre.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\h
adoop-annotations-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\hadoop-auth-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\c
ommon\lib\htrace-core4-4.1.0-incubating.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\httpclient-4.5.13.jar;C:\hadoop-3.2.
4\share\hadoop\common\lib\httpcore-4.4.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\j2objc-annotations-1.1.jar;C:\hado
op-3.2.4\share\hadoop\common\lib\jackson-annotations-2.10.5.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-2.1
0.5.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-asl-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jack
```

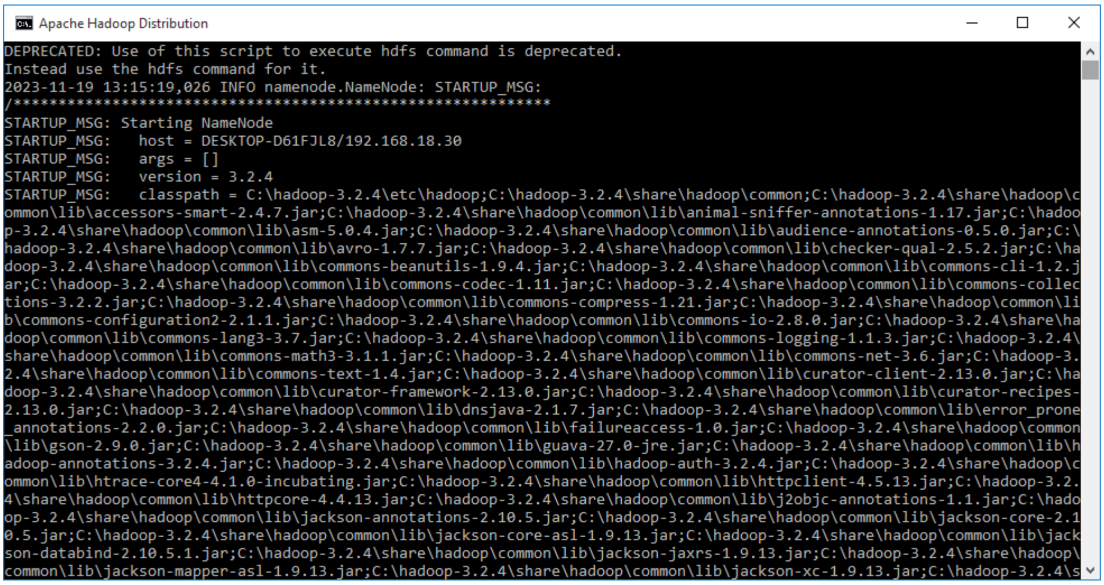
***Step 8–2*:** Enter C:\hadoop-3.2.4\sbin when inside the folder, click the path area and write cmd: start-all.cmd



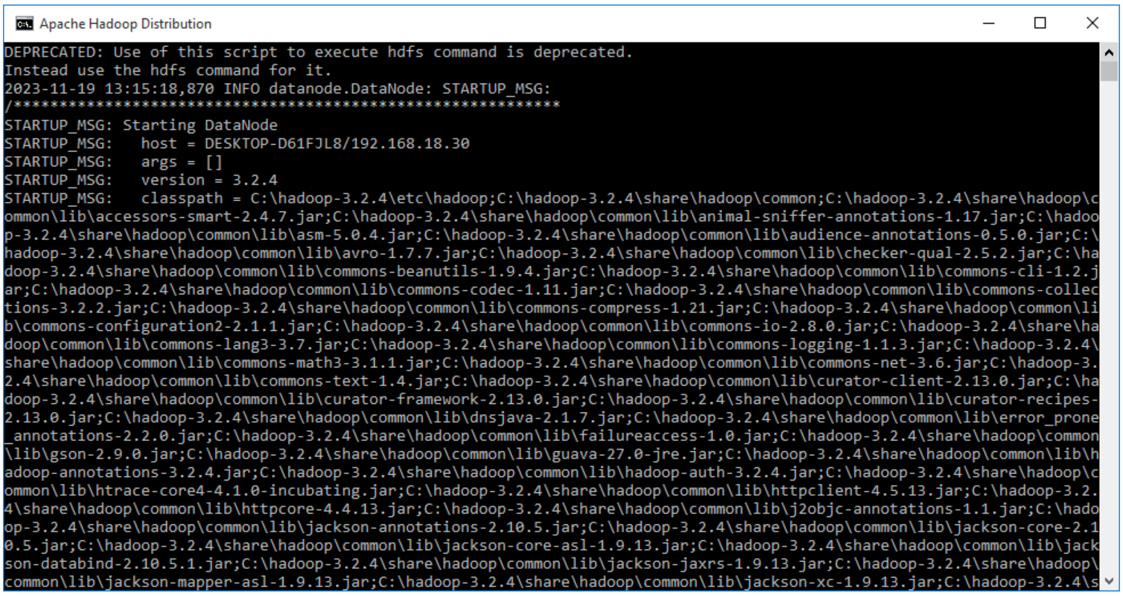
```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\hadoop-3.2.4\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\hadoop-3.2.4\sbin>
```

and will be see 4 result to ok for this step:

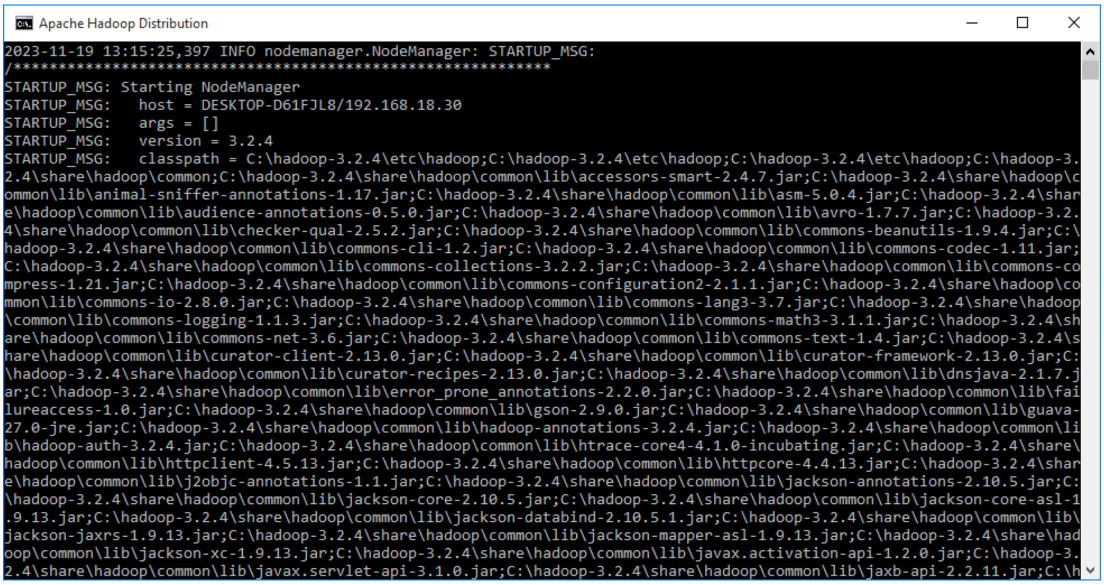
- Namenode cmd will be open and logs start to run
- Datanode cmd will be open and logs start to run
- Resourcemanager cmd will be open and logs start to run
- Nodemanager cmd will be open and logs start to run



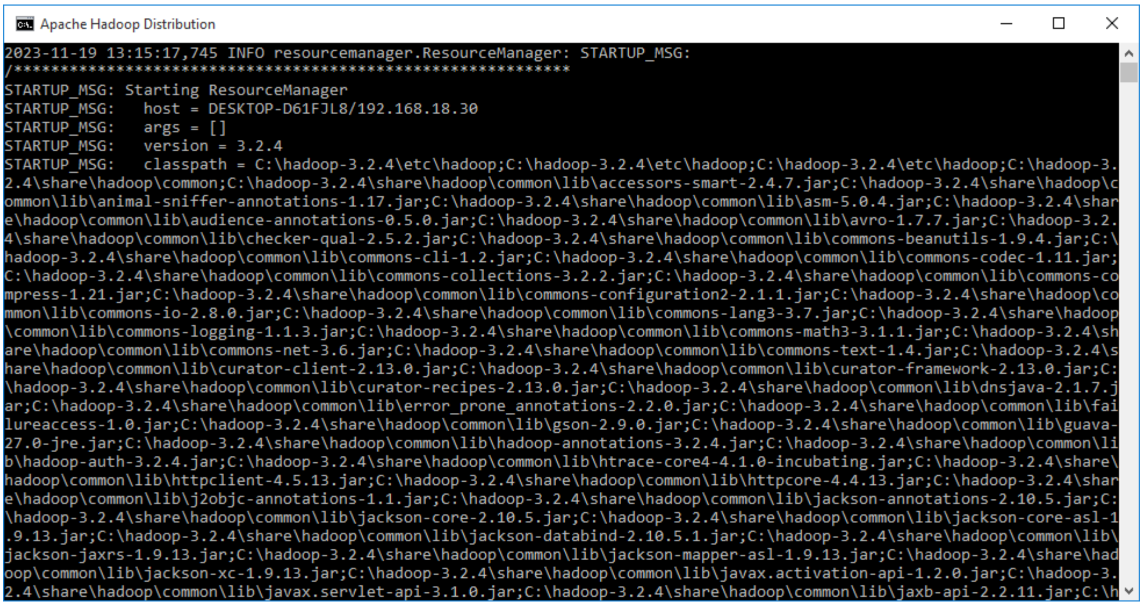
```
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
2023-11-19 13:15:19,026 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = DESKTOP-D61FJL8/192.168.18.30
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.2.4
STARTUP_MSG:   classpath = C:\hadoop-3.2.4\etc\hadoop;C:\hadoop-3.2.4\share\hadoop\common;C:\hadoop-3.2.4\share\hadoop\c
ommon\lib\accessors-smart-2.4.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoo
p-3.2.4\share\hadoop\common\lib\asm-5.0.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\
hadoop-3.2.4\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\ha
doop-3.2.4\share\hadoop\common\lib\commons-beanutils-1.9.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-cli-1.2.j
ar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-codec-1.11.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-collec
tions-3.2.2.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-compress-1.21.jar;C:\hadoop-3.2.4\share\hadoop\common\li
b\commons-configuration2-2.1.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-io-2.8.0.jar;C:\hadoop-3.2.4\share\ha
doop\common\lib\commons-lang3-3.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop-3.2.4\
share\hadoop\common\lib\commons-math3-3.1.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop-3.
2.4\share\hadoop\common\lib\commons-text-1.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\curator-client-2.13.0.jar;C:\ha
doop-3.2.4\share\hadoop\common\lib\curator-framework-2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\curator-recipes-
2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\dnsjava-2.1.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\error_prone
_annotations-2.2.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\failureaccess-1.0.jar;C:\hadoop-3.2.4\share\hadoop\common
\lib\gson-2.9.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\guava-27.0-jre.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\h
adoop-annotations-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\hadoop-auth-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\c
ommon\lib\htrace-core4-4.1.0-incubating.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\httpclient-4.5.13.jar;C:\hadoop-3.2.
4\share\hadoop\common\lib\httpcore-4.4.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\j2objc-annotations-1.1.jar;C:\hado
op-3.2.4\share\hadoop\common\lib\jackson-annotations-2.10.5.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-2.1
0.5.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-asl-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jack
son-databind-2.10.5.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-jaxrs-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\
common\lib\jackson-mapper-asl-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-xc-1.9.13.jar;C:\hadoop-3.2.4\s
```



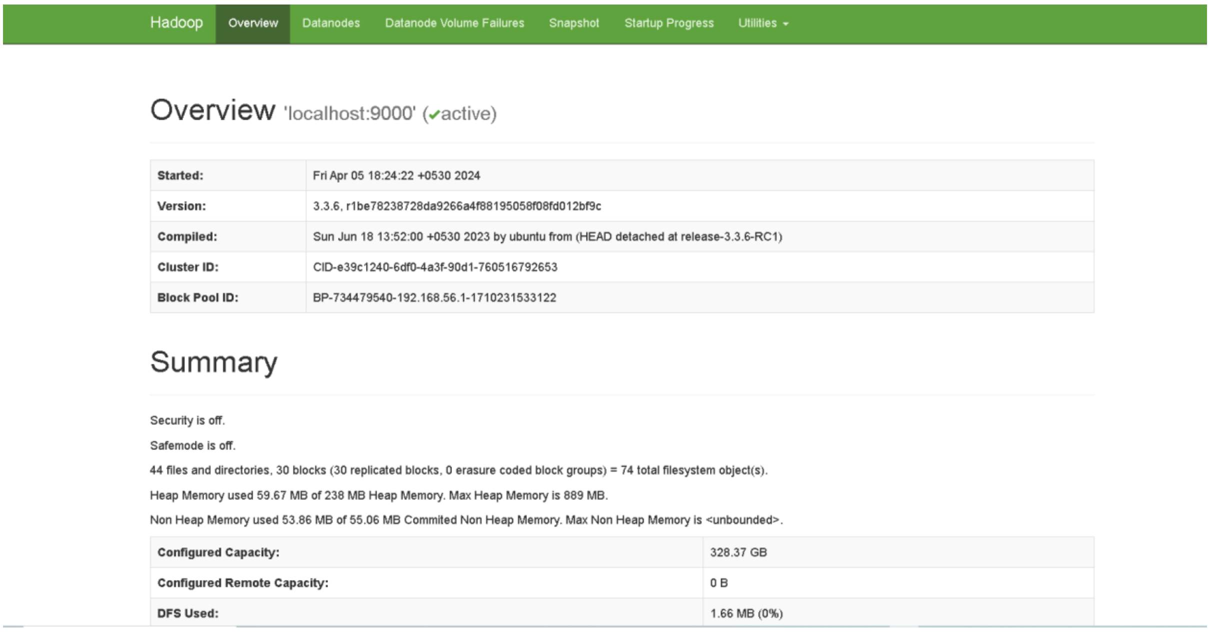
```
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
2023-11-19 13:15:18,870 INFO datanode.DataNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting DataNode
STARTUP_MSG:   host = DESKTOP-D61FJL8/192.168.18.30
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.2.4
STARTUP_MSG:   classpath = C:\hadoop-3.2.4\etc\hadoop;C:\hadoop-3.2.4\share\hadoop\common;C:\hadoop-3.2.4\share\hadoop\c
ommon\lib\accessors-smart-2.4.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoo
p-3.2.4\share\hadoop\common\lib\asm-5.0.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\
hadoop-3.2.4\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\ha
doop-3.2.4\share\hadoop\common\lib\commons-beanutils-1.9.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-cli-1.2.j
ar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-codec-1.11.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-collec
tions-3.2.2.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-compress-1.21.jar;C:\hadoop-3.2.4\share\hadoop\common\li
b\commons-configuration2-2.1.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-io-2.8.0.jar;C:\hadoop-3.2.4\share\ha
doop\common\lib\commons-lang3-3.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-logging-1.1.3.jar;C:\hadoop-3.2.4\
share\hadoop\common\lib\commons-math3-3.1.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop-3.
2.4\share\hadoop\common\lib\commons-text-1.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\curator-client-2.13.0.jar;C:\ha
doop-3.2.4\share\hadoop\common\lib\curator-framework-2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\curator-recipes-
2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\dnsjava-2.1.7.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\error_prone
_annotations-2.2.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\failureaccess-1.0.jar;C:\hadoop-3.2.4\share\hadoop\common
\lib\gson-2.9.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\guava-27.0-jre.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\h
adoop-annotations-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\hadoop-auth-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\c
ommon\lib\htrace-core4-4.1.0-incubating.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\httpclient-4.5.13.jar;C:\hadoop-3.2.
4\share\hadoop\common\lib\httpcore-4.4.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\j2objc-annotations-1.1.jar;C:\hado
op-3.2.4\share\hadoop\common\lib\jackson-annotations-2.10.5.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-2.1
0.5.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-asl-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jack
son-databind-2.10.5.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-jaxrs-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\
common\lib\jackson-mapper-asl-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-xc-1.9.13.jar;C:\hadoop-3.2.4\s
```



```
2023-11-19 13:15:25,397 INFO nodemanager.NodeManager: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NodeManager
STARTUP_MSG:   host = DESKTOP-D61FJL8/192.168.18.30
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.2.4
STARTUP_MSG:   classpath = C:\hadoop-3.2.4\etc\hadoop;C:\hadoop-3.2.4\etc\hadoop;C:\hadoop-3.2.4\etc\hadoop;C:\hadoop-3.
2.4\share\hadoop\common;C:\hadoop-3.2.4\share\hadoop\common\lib\accessors-smart-2.4.7.jar;C:\hadoop-3.2.4\share\hadoop\c
ommon\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\asm-5.0.4.jar;C:\hadoop-3.2.4\shar
e\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop-3.2.
4\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-beanutils-1.9.4.jar;C:\
hadoop-3.2.4\share\hadoop\common\lib\commons-cli-1.2.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-codec-1.11.jar;
C:\hadoop-3.2.4\share\hadoop\common\lib\commons-collections-3.2.2.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-co
mpress-1.21.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-configuration2-2.1.1.jar;C:\hadoop-3.2.4\share\hadoop\co
mmon\lib\commons-io-2.8.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-lang3-3.7.jar;C:\hadoop-3.2.4\share\hadoop
\common\lib\commons-logging-1.1.3.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-math3-3.1.1.jar;C:\hadoop-3.2.4\sh
are\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-text-1.4.jar;C:\hadoop-3.2.4\s
hare\hadoop\common\lib\curator-client-2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\curator-framework-2.13.0.jar;C:
\hadoop-3.2.4\share\hadoop\common\lib\curator-recipes-2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\dnsjava-2.1.7.j
ar;C:\hadoop-3.2.4\share\hadoop\common\lib\error_prone_annotations-2.2.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\fai
lureaccess-1.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\gson-2.9.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\guava-
27.0-jre.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\hadoop-annotations-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\common\li
b\hadoop-auth-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\htrace-core4-4.1.0-incubating.jar;C:\hadoop-3.2.4\share\
hadoop\common\lib\httpclient-4.5.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\httpcore-4.4.13.jar;C:\hadoop-3.2.4\shar
e\hadoop\common\lib\j2objc-annotations-1.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-annotations-2.10.5.jar;C:
\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-2.10.5.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-asl-1
.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-databind-2.10.5.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\
jackson-jaxrs-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-mapper-asl-1.9.13.jar;C:\hadoop-3.2.4\share\had
oop\common\lib\jackson-xc-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\javax.activation-api-1.2.0.jar;C:\hadoop-3.
2.4\share\hadoop\common\lib\javax.servlet-api-3.1.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jaxb-api-2.2.11.jar;C:\h
```



```
2023-11-19 13:15:17,745 INFO resourcemanager.ResourceManager: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting ResourceManager
STARTUP_MSG:   host = DESKTOP-D61FJL8/192.168.18.30
STARTUP_MSG:   args = []
STARTUP_MSG:   version = 3.2.4
STARTUP_MSG:   classpath = C:\hadoop-3.2.4\etc\hadoop;C:\hadoop-3.2.4\etc\hadoop;C:\hadoop-3.2.4\etc\hadoop;C:\hadoop-3.
2.4\share\hadoop\common;C:\hadoop-3.2.4\share\hadoop\common\lib\accessors-smart-2.4.7.jar;C:\hadoop-3.2.4\share\hadoop\c
ommon\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\asm-5.0.4.jar;C:\hadoop-3.2.4\shar
e\hadoop\common\lib\audience-annotations-0.5.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop-3.2.
4\share\hadoop\common\lib\checker-qual-2.5.2.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-beanutils-1.9.4.jar;C:\
hadoop-3.2.4\share\hadoop\common\lib\commons-cli-1.2.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-codec-1.11.jar;
C:\hadoop-3.2.4\share\hadoop\common\lib\commons-collections-3.2.2.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-co
mpress-1.21.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-configuration2-2.1.1.jar;C:\hadoop-3.2.4\share\hadoop\co
mmon\lib\commons-io-2.8.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-lang3-3.7.jar;C:\hadoop-3.2.4\share\hadoop
\common\lib\commons-logging-1.1.3.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-math3-3.1.1.jar;C:\hadoop-3.2.4\sh
are\hadoop\common\lib\commons-net-3.6.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\commons-text-1.4.jar;C:\hadoop-3.2.4\s
hare\hadoop\common\lib\curator-client-2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\curator-framework-2.13.0.jar;C:
\hadoop-3.2.4\share\hadoop\common\lib\curator-recipes-2.13.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\dnsjava-2.1.7.j
ar;C:\hadoop-3.2.4\share\hadoop\common\lib\error_prone_annotations-2.2.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\fai
lureaccess-1.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\gson-2.9.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\guava-
27.0-jre.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\hadoop-annotations-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\common\li
b\hadoop-auth-3.2.4.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\htrace-core4-4.1.0-incubating.jar;C:\hadoop-3.2.4\share\
hadoop\common\lib\httpclient-4.5.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\httpcore-4.4.13.jar;C:\hadoop-3.2.4\shar
e\hadoop\common\lib\j2objc-annotations-1.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-annotations-2.10.5.jar;C:
\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-2.10.5.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-core-asl-1
.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-databind-2.10.5.1.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\
jackson-jaxrs-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jackson-mapper-asl-1.9.13.jar;C:\hadoop-3.2.4\share\had
oop\common\lib\jackson-xc-1.9.13.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\javax.activation-api-1.2.0.jar;C:\hadoop-3.
2.4\share\hadoop\common\lib\javax.servlet-api-3.1.0.jar;C:\hadoop-3.2.4\share\hadoop\common\lib\jaxb-api-2.2.11.jar;C:\h
```



Overview 'localhost:9000' (✓active)

Started:	Fri Apr 05 18:24:22 +0530 2024
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-e39c1240-6df0-4a3f-90d1-760516792653
Block Pool ID:	BP-734479540-192.168.56.1-1710231533122

Summary

Security is off.

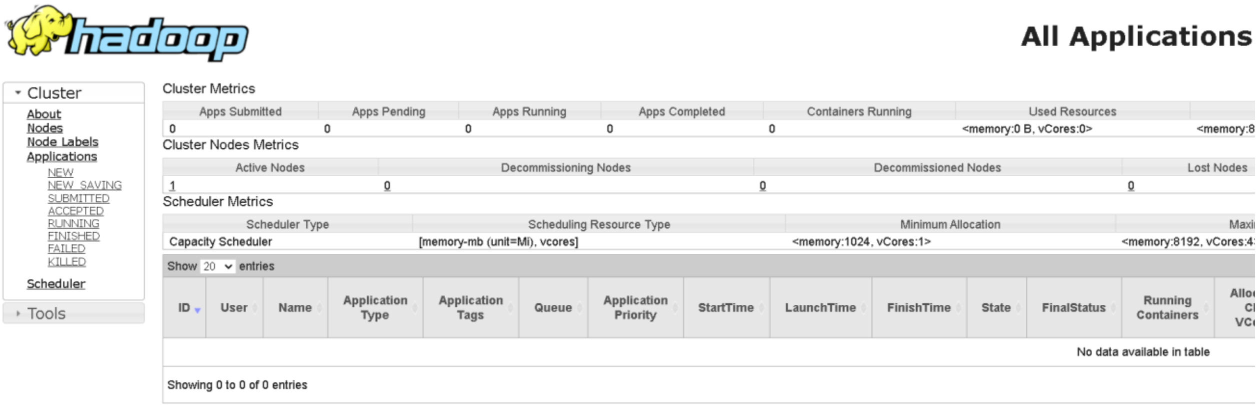
Safemode is off.

44 files and directories, 30 blocks (30 replicated blocks, 0 erasure coded block groups) = 74 total filesystem object(s).

Heap Memory used 59.67 MB of 238 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 53.86 MB of 55.06 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	328.37 GB
Configured Remote Capacity:	0 B
DFS Used:	1.66 MB (0%)



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	
0	0	0	0	0	<memory:0 B, vCores:0>	<memory:8

Cluster Nodes Metrics

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

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maxi
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Alloc C VC
											No data available in table		

Showing 0 to 0 of 0 entries

RESULT:

Thus, the Hadoop is downloaded and installed also understand different Hadoop modes, Startup scripts and Configuration files are successfully implemented.

Ex.No: Date:	HADOOP IMPLEMENTATION OF FILE MANAGEMENT TASKS, SUCH AS ADDING FILES AND DIRECTORIES, RETRIEVING FILES AND DELETING FILES

AIM:

To implement the following file management tasks in Hadoop:

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

1. Create a directory in HDFS at the given path(s).

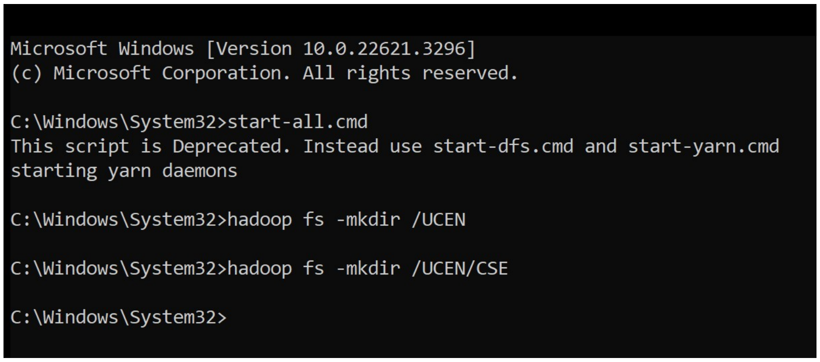
Usage:

hadoop fs -mkdir <path>

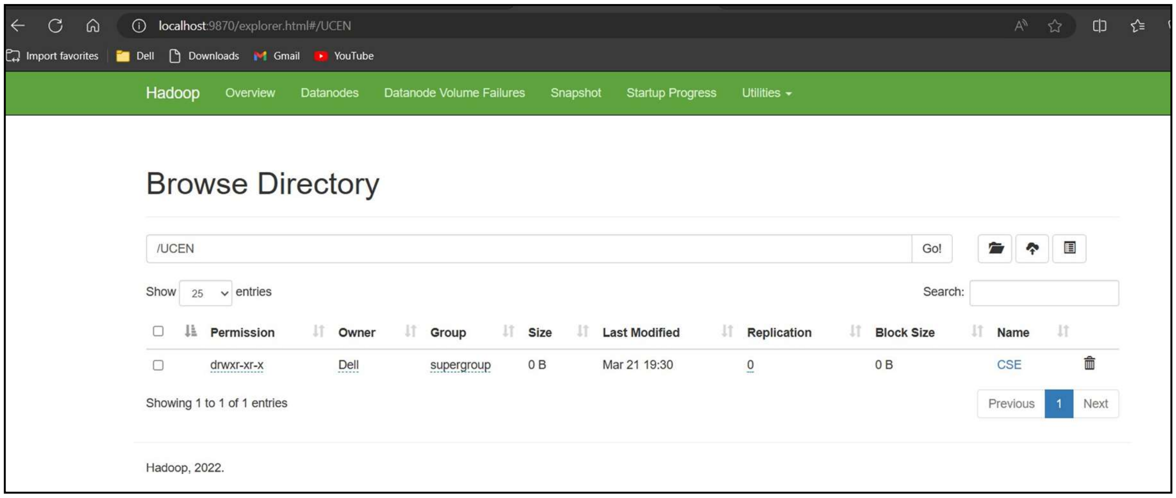
Example:

Hadoop fs -mkdir /UCEN/CSE

OUTPUT:



```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Windows\System32>hadoop fs -mkdir /UCEN

C:\Windows\System32>hadoop fs -mkdir /UCEN/CSE

C:\Windows\System32>
```



Browse Directory

/UCEN

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	drwxr-xr-x	Dell	supergroup	0 B	Mar 21 19:30	0	0 B	CSE

Showing 1 to 1 of 1 entries

Hadoop, 2022.

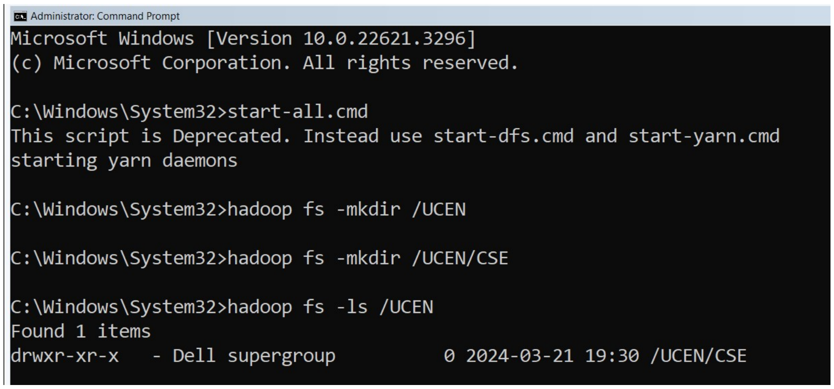
2. List the contents of a directory.

Usage:

hadoop fs -ls <args>

Example:

Hadoop fs -ls /UCEN



```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Windows\System32>hadoop fs -mkdir /UCEN

C:\Windows\System32>hadoop fs -mkdir /UCEN/CSE

C:\Windows\System32>hadoop fs -ls /UCEN
Found 1 items
drwxr-xr-x   - Dell supergroup          0 2024-03-21 19:30 /UCEN/CSE
```

3. Upload and download a file in HDFS.

Upload:

hadoop fs -put:

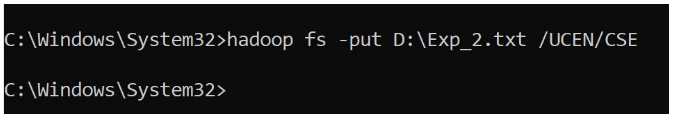
Copy a single src file, or multiple src files from the local file system to the Hadoop data file system.

Usage:

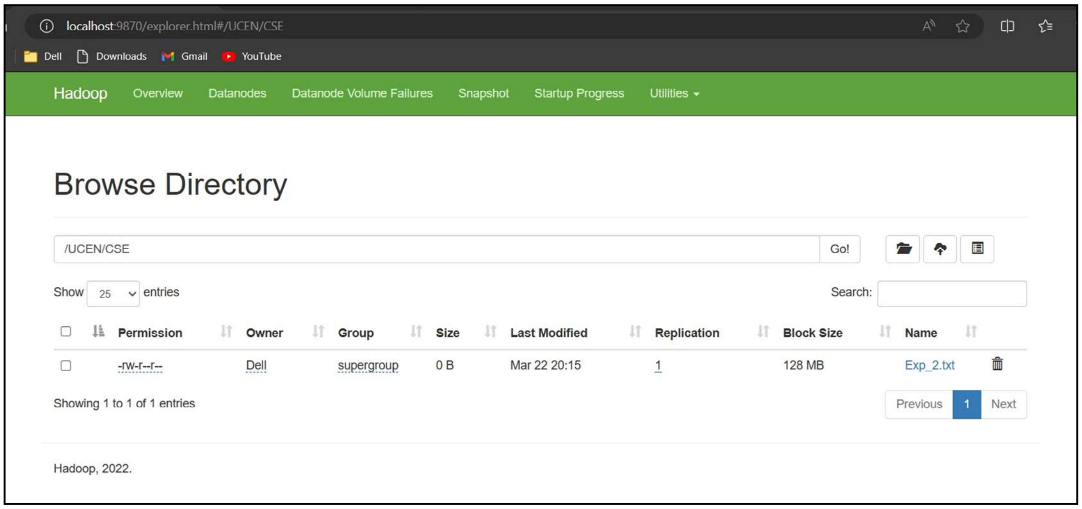
hadoop fs -put <localscr> … <HDFS_dest_Path>

Example:

hadoop fs -put D:\Exp_2.txt /UCEN/CSE



```
C:\Windows\System32>hadoop fs -put D:\Exp_2.txt /UCEN/CSE

C:\Windows\System32>
```



Browse Directory

/UCEN/CSE

Show 25 entries

Search:

☐	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
☐	-rw-r--r--	Dell	supergroup	0 B	Mar 22 20:15	1	128 MB	Exp_2.txt

Showing 1 to 1 of 1 entries

Hadoop, 2022.

Download:

hadoop fs -get:

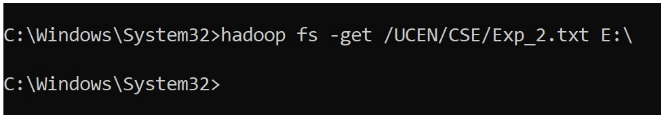
Copies/Downloads files to the local file system.

Usage:

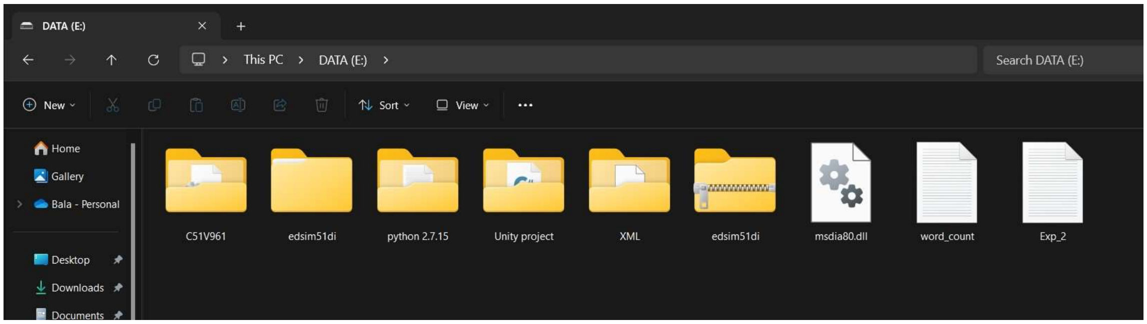
hadoop fs -get <HDFS_src> <localst>

Example:

hadoop fs -get /UCEN/CSE/Exp_2.txt E:\



```
C:\Windows\System32>hadoop fs -get /UCEN/CSE/Exp_2.txt E:\

C:\Windows\System32>
```



4. See the contents of a file

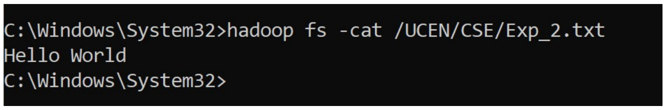
Same as the Unix command

Usage:

hadoop fs -cut <path[filename]>

Example:

Hadoop fs -cut /UCEN/CSE/Exp_2.txt



```
C:\Windows\System32>hadoop fs -cat /UCEN/CSE/Exp_2.txt
Hello World
C:\Windows\System32>
```

5. Remove a file or directory in HDFS.

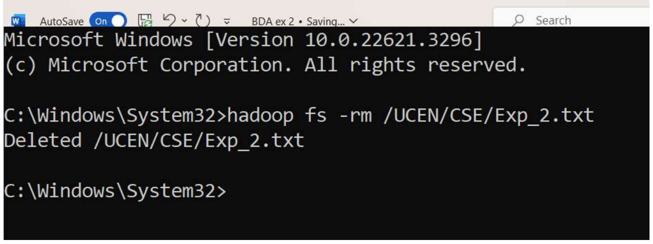
Remove files specified as arguments. Deletes directory only when it is empty

Usage:

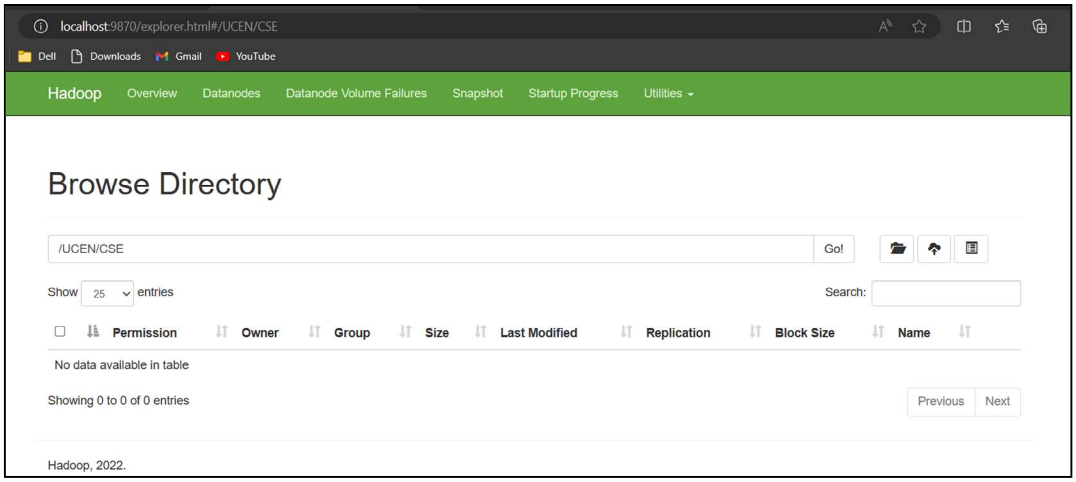
hadoop fs -rm <arg>

Example:

hadoop fs -rm /UCEN/CSE/Exp_2.txt



```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>hadoop fs -rm /UCEN/CSE/Exp_2.txt
Deleted /UCEN/CSE/Exp_2.txt

C:\Windows\System32>
```



Browse Directory

/UCEN/CSE

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
No data available in table								

Showing 0 to 0 of 0 entries

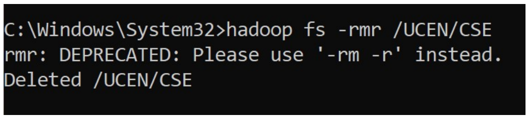
Hadoop, 2022.

Recursive version of deleteUsage:

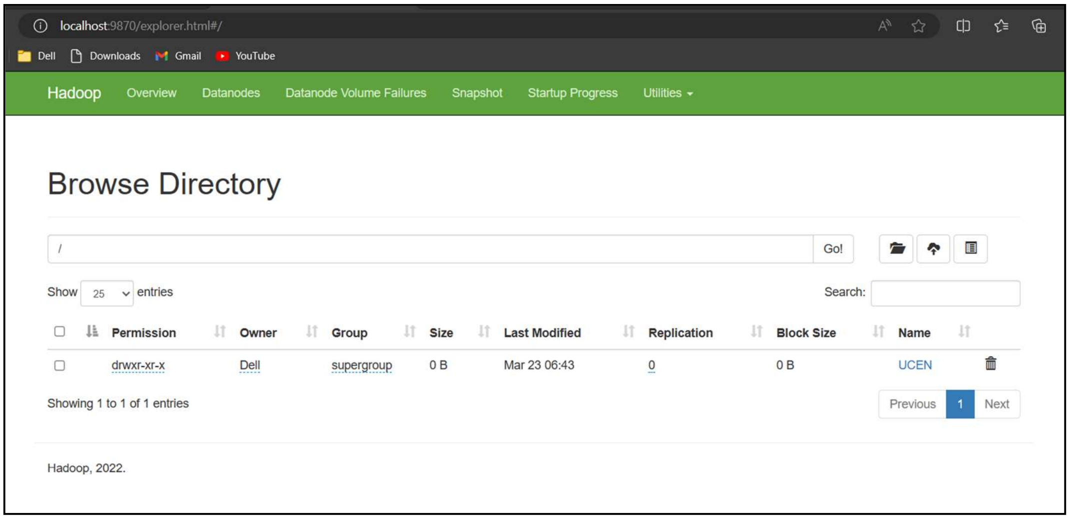
hadoop fs -rmr <arg>

Example:

hadoop fs -rmr /UCEN/CSE



```
C:\Windows\System32>hadoop fs -rmr /UCEN/CSE
rmr: DEPRECATED: Please use '-rm -r' instead.
Deleted /UCEN/CSE
```



Browse Directory

/

Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	drwxr-xr-x	Dell	supergroup	0 B	Mar 23 06:43	0	0 B	UCEN

Showing 1 to 1 of 1 entries

Hadoop, 2022.

RESULT:

Thus, the Hadoop Implementation of file management tasks, such as Adding files and directories, retrieving files and Deleting files is executed successfully.

EX NO:3	**IMPLEMENT MATRIX MULTIPLICATION WITH HADOOP MAP REDUCE**
DATE:	

AIM:

To write a MapReduce program that implements Matrix Multiplication.

ALGORITHM:

We assume that the input matrices are already stored in Hadoop Distributed File System (HDFS) in a suitable format (e.g., CSV, TSV) where each row represents a matrix element. The matrices are compatible for multiplication (the number of columns in the first matrix is equal to the number of rows in the second matrix).

STEP 1: MAPPER

The mapper will take the input matrices and emit key-value pairs for each element in the result matrix. The key will be the (row, column) index of the result element, and the value will be the corresponding element value.

STEP 2: REDUCER

The reducer will take the key-value pairs emitted by the mapper and calculate the partial sum for each element in the result matrix.

STEP 3: MAIN DRIVER

The main driver class sets up the Hadoop job configuration and specifies the input and output paths for the matrices.

STEP 4: RUNNING THE JOB

To run the MapReduce job, you need to package your classes into a JAR file and then submit it to Hadoop using the hadoop jar command. Make sure to replace input_path and output_path with the actual HDFS paths to your input matrices and desired output directory.

Map.java:

```
    package com.mapreduce.wc;
    import org.apache.hadoop.conf.*;
    import org.apache.hadoop.io.LongWritable;
    import org.apache.hadoop.io.Text;
    //import org.apache.hadoop.mapreduce.Mapper;
    import java.io.IOException;
    public class Map extends org.apache.hadoop.mapreduce.Mapper<LongWritable, Text, Text,
Text>
    {
    @Override
    public void map(LongWritable key, Text value, Context context)
    throws IOException, InterruptedException {
            Configuration conf = context.getConfiguration();
            int m = Integer.parseInt(conf.get("m"));
            int p = Integer.parseInt(conf.get("p"));
    String line = value.toString();
    // (M, i, j, Mij);
    String[] indicesAndValue = line.split(",");
    Text outputKey = new Text();
    Text outputValue = new Text();
    if (indicesAndValue[0].equals("M")) {
    for (int k = 0; k < p; k++) {
    outputKey.set(indicesAndValue[1] + "," + k);
    // outputKey.set(i,k);
    outputValue.set(indicesAndValue[0] + "," + indicesAndValue[2]
    + "," + indicesAndValue[3]);
    // outputValue.set(M,j,Mij);
    context.write(outputKey, outputValue);
    }
    } else {
    // (N, j, k, Njk);

    for (int i = 0; i < m; i++) {
    outputKey.set(i + "," + indicesAndValue[2]); outputValue.set("N," + indicesAndValue[1] +
","
    + indicesAndValue[3]); context.write(outputKey, outputValue);
    }
    }
    }
    }
```

MatrixMultiply.java:

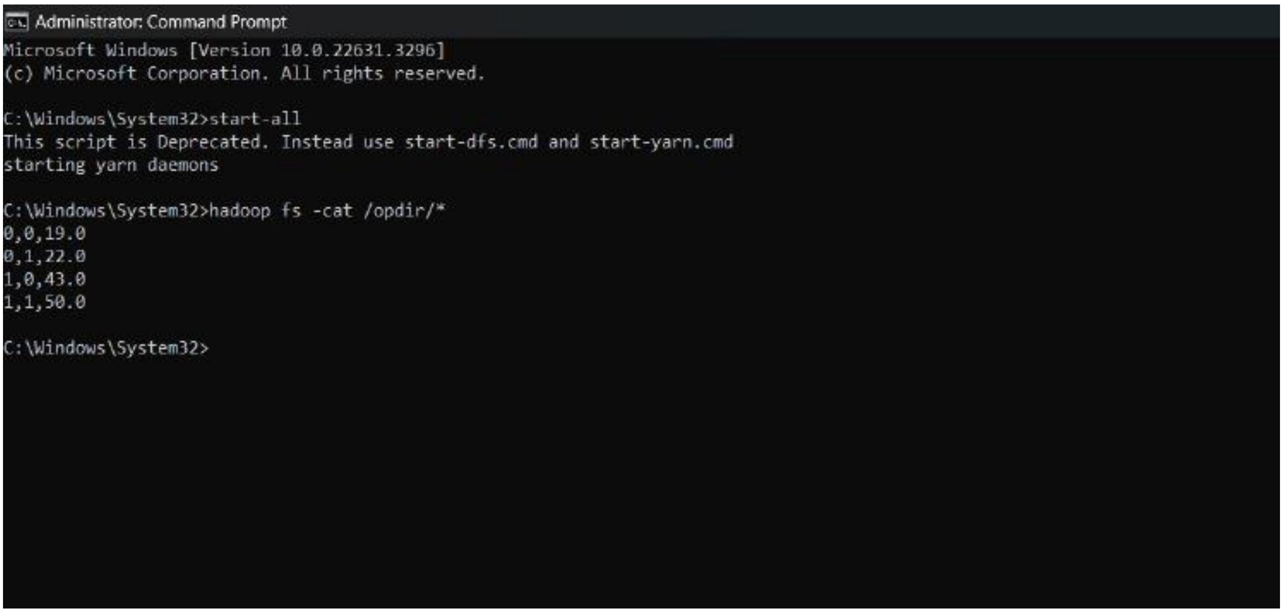
```
  package com.mapreduce.wc;

  import org.apache.hadoop.conf.*;
  import org.apache.hadoop.fs.Path;
  import org.apache.hadoop.io.*;
  import org.apache.hadoop.mapreduce.*;
  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
  import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
  import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
```

```
public class MatrixMultiply {
public static void main(String[] args) throws Exception { if (args.length != 2) {
System.err.println("Usage: MatrixMultiply <in_dir> <out_dir>");
System.exit(2);
}
Configuration conf = new Configuration();
conf.set("m", "1000");
conf.set("n", "100");
conf.set("p", "1000");
@SuppressWarnings("deprecation")
Job job = new Job(conf, "MatrixMultiply");
job.setJarByClass(MatrixMultiply.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(Text.class);
job.setMapperClass(Map.class);
job.setReducerClass(Reduce.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
job.waitForCompletion(true);
}
}
```

Reducer.java:

```
  package com.mapreduce.wc;


  import org.apache.hadoop.io.Text;
  // import org.apache.hadoop.mapreduce.Reducer;
  import java.io.IOException;
  import java.util.HashMap;
  public class Reduce
  extends org.apache.hadoop.mapreduce.Reducer<Text, Text, Text, Text> { @Override
  public void reduce(Text key, Iterable<Text> values, Context context)
  throws IOException, InterruptedException {
  String[] value;
  //key=(i,k),
  //Values = [(M/N,j,V/W),..]
  HashMap<Integer, Float> hashA = new HashMap<Integer, Float>(); HashMap<Integer,
Float> hashB = new HashMap<Integer, Float>(); for (Text val : values) {
  value = val.toString().split(",");
  if (value[0].equals("M")) {
  hashA.put(Integer.parseInt(value[1]), Float.parseFloat(value[2])); } else {
  hashB.put(Integer.parseInt(value[1]), Float.parseFloat(value[2]));
  }
  }
  int n = Integer.parseInt(context.getConfiguration().get("n"));
  float result = 0.0f;
  float m_ij;
  float n_jk;
for (int j = 0; j < n; j++) {
m_ij = hashA.containsKey(j) ? hashA.get(j) : 0.0f; n_jk = hashB.containsKey(j) ? hashB.get(j)
: 0.0f; result += m_ij * n_jk;
}
if (result != 0.0f) {
context.write(null,
new Text(key.toString() + "," + Float.toString(result)));
}
}
}
```

OUTPUT:



```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>start-all
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Windows\System32>hadoop fs -cat /opdir/*
0,0,19.0
0,1,22.0
1,0,43.0
1,1,50.0

C:\Windows\System32>
```

RESULT:

Thus, the MapReduce program that implements matrix multiplication was verified and executed successfully.

EXP.NO:4 **DATE**:	**RUN A BASIC WORDCOUNT MAP REDUCE PROGRAM TO UNDERSTAND MAP RDUCE PARADIGM**

AIM:

To write a basic WordCount program to understand Map Reduce Paradigm.

PROCEDURE:

Step 1: Setting up the Eclipse Workspace

- Launch Eclipse and select an appropriate workspace directory where your project will reside.

Step 2: Creating the MapReduce Project

- Navigate through File > New > Java Project to initiate a new project setup. Assign a project name and finalize the creation.

Step 3: Structuring the Project

- Create a new package within the project to maintain a clean project structure.
- Add the necessary Hadoop library jars to your project's build path, including client, common, HDFS, MapReduce, and YARN components, to ensure all necessary functionalities are accessible.

Step 4: Developing the Mapper Class

- Implement the WordCountMapper class, responsible for mapping input key-value pairs to a set of intermediate key-value pairs.
- The mapper reads text input and emits each word accompanied by an integer (typically 1) to indicate an occurrence.

Step 5: Implementing the Reducer Class

- Create the WordCountReducer class, which reduces a set of intermediate values sharing a key to a smaller set of values.
- The reducer sums up the counts for each word, resulting in a final count per word across the dataset.

Step 6: Configuring the Driver Class

- Develop the WordCount driver class, encapsulating the job's configuration such as setting the job's input and output formats, specifying the mapper and reducer classes, and initiating the job processing.

Step 7: Executing the Project

- Compile and run the project in Eclipse, specifying the input and output paths for the job. The output will be the word count for each unique word found in the input dataset.

ALGORITHM:

Step 1: Mapper Code

- The Mapper class in our project processes raw text as input. It reads each line of text and emits key-value pairs, where the key represents a word and the value is always set to 1, indicating the presence of that word in the input data.

Step 2: Reducer Code

- The Reducer class receives input as pairs of words and lists of counts. It aggregates the counts for each word, summing up occurrences across all input data chunks. Finally, it emits key-value pairs where the key is a word and the value is the total count of occurrences of that word across the dataset.

Step 3: Driver Code

The Driver code configures and initiates the MapReduce job. It sets up job-specific parameters and classes, such as the Mapper, Reducer, and input/output formats.

1. **Mapper Input:** The input to the mapper is raw text.
2. **Mapper Output:** Outputs <key, value> pairs where the key is a word and the value is an integer (1).
3. **Reducer Input:** Input to the reducer is the <key, list(values)> pairs.
4. **Reducer Output:** Aggregates the counts for each word and emits <word, total count> as output.

PROGRAM:

WordCountMapper.java

```
import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.io.LongWritable;

public class WordCountMapper extends Mapper <LongWritable, Text, Text, IntWritable>
```

```
{
private Text wordToken = new Text();
public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException
{
StringTokenizer tokens = new StringTokenizer(value.toString()); //Dividing String into tokens
while (tokens.hasMoreTokens())
{
wordToken.set(tokens.nextToken());
context.write(wordToken, new IntWritable(1));
}
}
}
```

WordCountReducer.java

```
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class WordCountReducer extends Reducer <Text, IntWritable, Text, IntWritable>
{
private IntWritable count = new IntWritable();
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException
{

int valueSum = 0;
for (IntWritable val : values)
{
valueSum += val.get();
}
count.set(valueSum);
context.write(key, count);
}
}
```

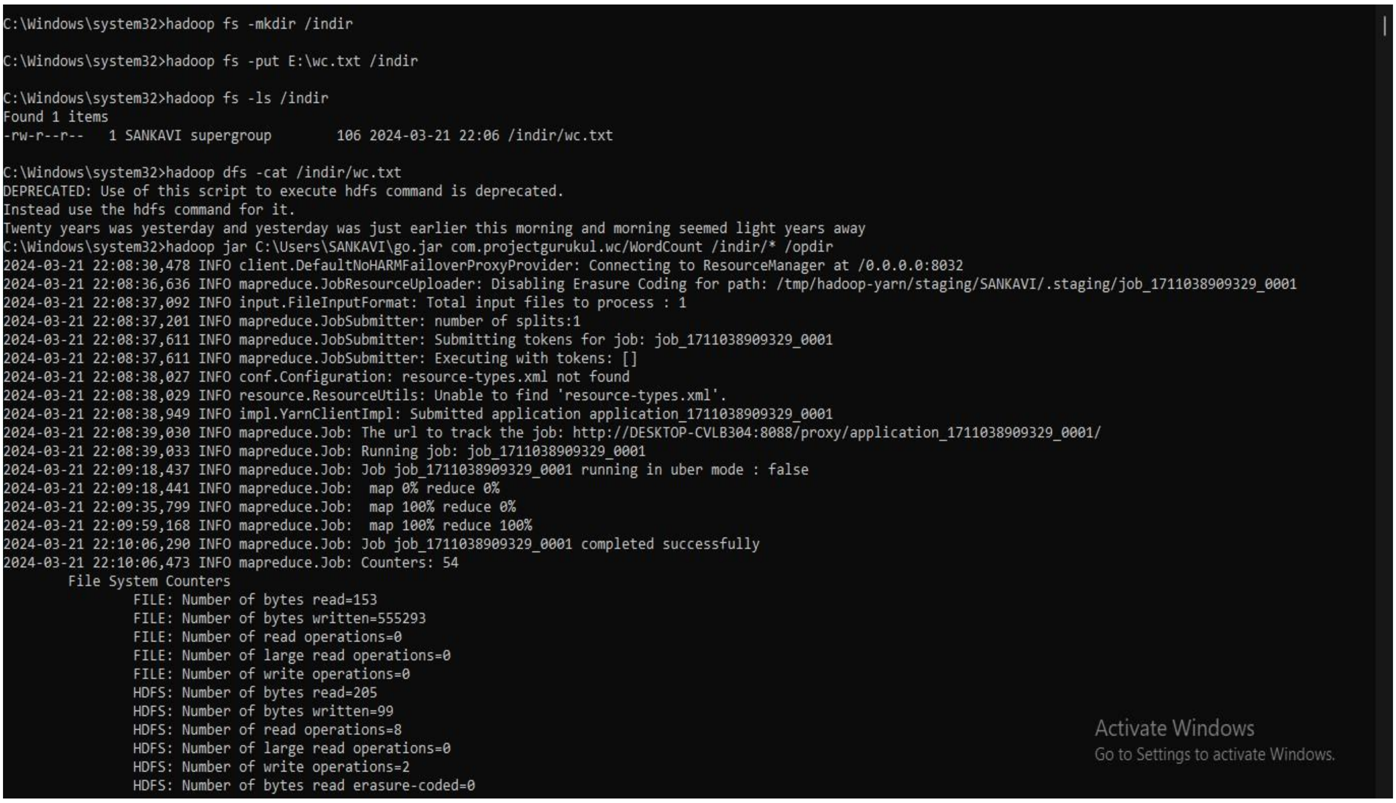
WordCount.java

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class WordCount
{
public static void main(String[] args) throws Exception
{
Configuration conf = new Configuration();
String[] pathArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
if (pathArgs.length < 2)
{
System.err.println("MR Project Usage: wordcount <input-path> […] <output-path>");
System.exit(2);
}
Job wcJob = Job.getInstance(conf, "MapReduce WordCount");
wcJob.setJarByClass(WordCount.class);
wcJob.setMapperClass(WordCountMapper.class);
wcJob.setCombinerClass(WordCountReducer.class);
wcJob.setReducerClass(WordCountReducer.class);
wcJob.setOutputKeyClass(Text.class);
wcJob.setOutputValueClass(IntWritable.class);
for (int i = 0; i < pathArgs.length — 1; ++i)
{
FileInputFormat.addInputPath(wcJob, new Path(pathArgs[i]));
}
FileOutputFormat.setOutputPath(wcJob, new Path(pathArgs[pathArgs.length — 1]));
System.exit(wcJob.waitForCompletion(true) ? 0 : 1);
}
}
```

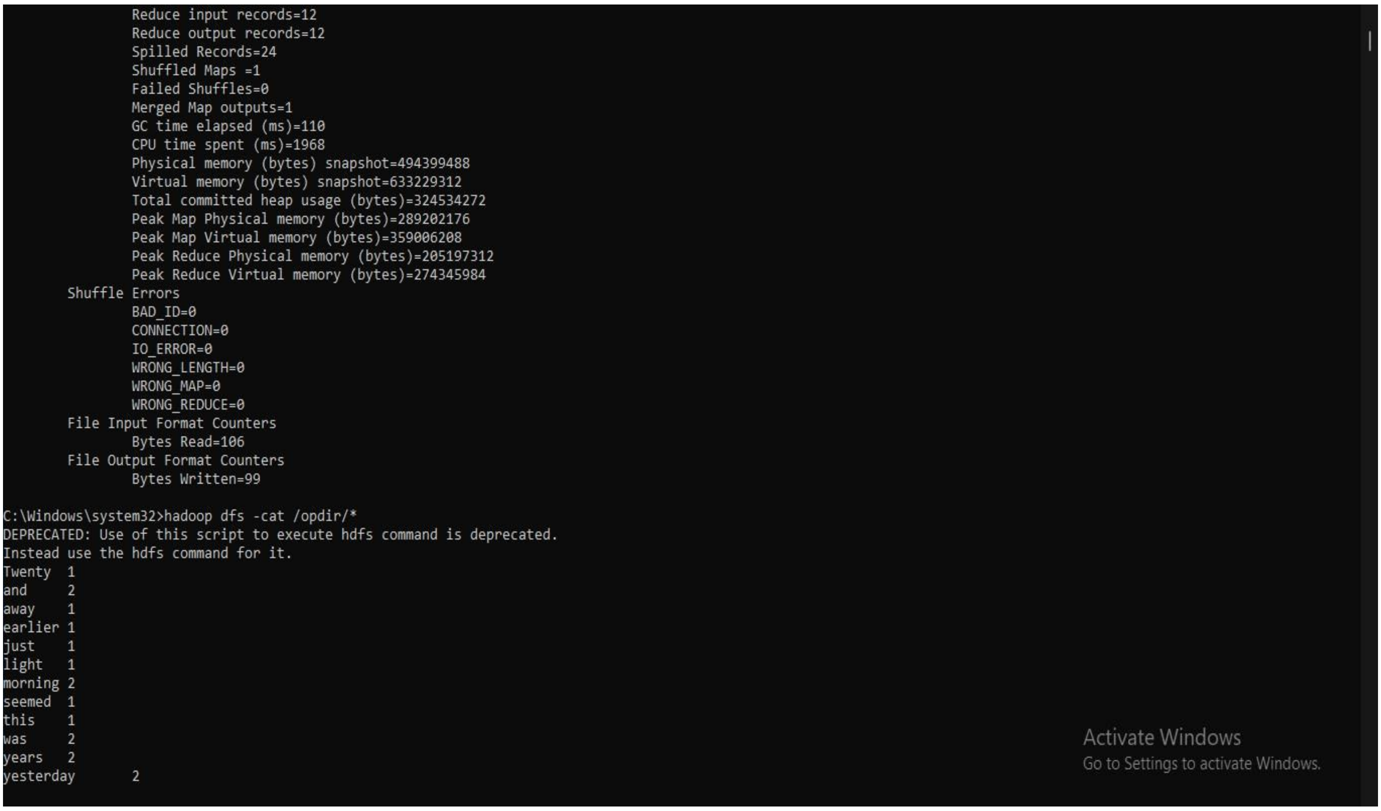
Executing Hadoop MapReduce Jobs from Command Line:

1. Create a Directory on HDFS:
   ```
   hdfs dfs -mkdir /input
   ```
2. Upload Input Text File to HDFS:
   ```
   hdfs dfs -put /path/to/your/input.txt /input
   ```
3. Check if the File is Uploaded:
   ```
   hdfs dfs -ls /input
   ```
4. View the Contents of the Uploaded Text File:
   ```
   hdfs dfs -cat /input/input.txt
   ```
5. Run Your Hadoop MapReduce Job:
   ```
   hadoop jar /path/to/your/jarfile.jar com.mapreduce.wc.WordCount /input /output
   ```
   Here, replace /path/to/your/jarfile.jar with the actual path to your compiled JAR file, and com.project.wc.WordCount with the fully qualified name of your Driver class. /input and /output are the input and output directories on HDFS, respectively.
6. Check the Output:
7. hdfs dfs -cat /output/*

OUTPUT:



```
C:\Windows\system32>hadoop fs -mkdir /indir

C:\Windows\system32>hadoop fs -put E:\wc.txt /indir

C:\Windows\system32>hadoop fs -ls /indir
Found 1 items
-rw-r--r--   1 SANKAVI supergroup        106 2024-03-21 22:06 /indir/wc.txt

C:\Windows\system32>hadoop dfs -cat /indir/wc.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Twenty years was yesterday and yesterday was just earlier this morning and morning seemed light years away
C:\Windows\system32>hadoop jar C:\Users\SANKAVI\go.jar com.projectgurukul.wc/WordCount /indir/* /opdir
2024-03-21 22:08:30,478 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-03-21 22:08:36,636 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/SANKAVI/.staging/job_1711038909329_0001
2024-03-21 22:08:37,092 INFO input.FileInputFormat: Total input files to process : 1
2024-03-21 22:08:37,201 INFO mapreduce.JobSubmitter: number of splits:1
2024-03-21 22:08:37,611 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1711038909329_0001
2024-03-21 22:08:37,611 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-03-21 22:08:38,027 INFO conf.Configuration: resource-types.xml not found
2024-03-21 22:08:38,029 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-03-21 22:08:38,949 INFO impl.YarnClientImpl: Submitted application application_1711038909329_0001
2024-03-21 22:08:39,030 INFO mapreduce.Job: The url to track the job: http://DESKTOP-CVLB304:8088/proxy/application_1711038909329_0001/
2024-03-21 22:08:39,033 INFO mapreduce.Job: Running job: job_1711038909329_0001
2024-03-21 22:09:18,437 INFO mapreduce.Job: Job job_1711038909329_0001 running in uber mode : false
2024-03-21 22:09:18,441 INFO mapreduce.Job:  map 0% reduce 0%
2024-03-21 22:09:35,799 INFO mapreduce.Job:  map 100% reduce 0%
2024-03-21 22:09:59,168 INFO mapreduce.Job:  map 100% reduce 100%
2024-03-21 22:10:06,290 INFO mapreduce.Job: Job job_1711038909329_0001 completed successfully
2024-03-21 22:10:06,473 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=153
                FILE: Number of bytes written=555293
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=205
                HDFS: Number of bytes written=99
                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
```



```
                Reduce input records=12
                Reduce output records=12
                Spilled Records=24
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=110
                CPU time spent (ms)=1968
                Physical memory (bytes) snapshot=494399488
                Virtual memory (bytes) snapshot=633229312
                Total committed heap usage (bytes)=324534272
                Peak Map Physical memory (bytes)=289202176
                Peak Map Virtual memory (bytes)=359006208
                Peak Reduce Physical memory (bytes)=205197312
                Peak Reduce Virtual memory (bytes)=274345984
        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=106
        File Output Format Counters
                Bytes Written=99

C:\Windows\system32>hadoop dfs -cat /opdir/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Twenty  1
and     2
away    1
earlier 1
just    1
light   1
morning 2
seemed  1
this    1
was     2
years   2
yesterday       2
```

RESULT:

The MapReduce program that implements WordCount was executed and verified successfully.

Ex.No: 5

Date:

INSTALLATION OF HIVE ALONG WITH PRACTICE EXAMPLES

AIM:

To install HIVE along with practice examples.

PREREQUISITES:

- Java Development Kit (JDK8) installed and the JAVA_HOME environment variable set.
- Hadoop(version 2.9.2) installed and configured on your Windows system.
- Apache Derby installed and DERBY_HOME environment variable set.

PROCEDURE:

1. Download HIVE:

Visit the Apache Hive website and download the stable version (version 3.1.2) of Hive.Official Apache Hive website: **https://hive.apache.org/**

2. Extract the Downloaded Hive Archive to a Directory on Your Windows Machine:

C:\hive.

3. Configure Hive:

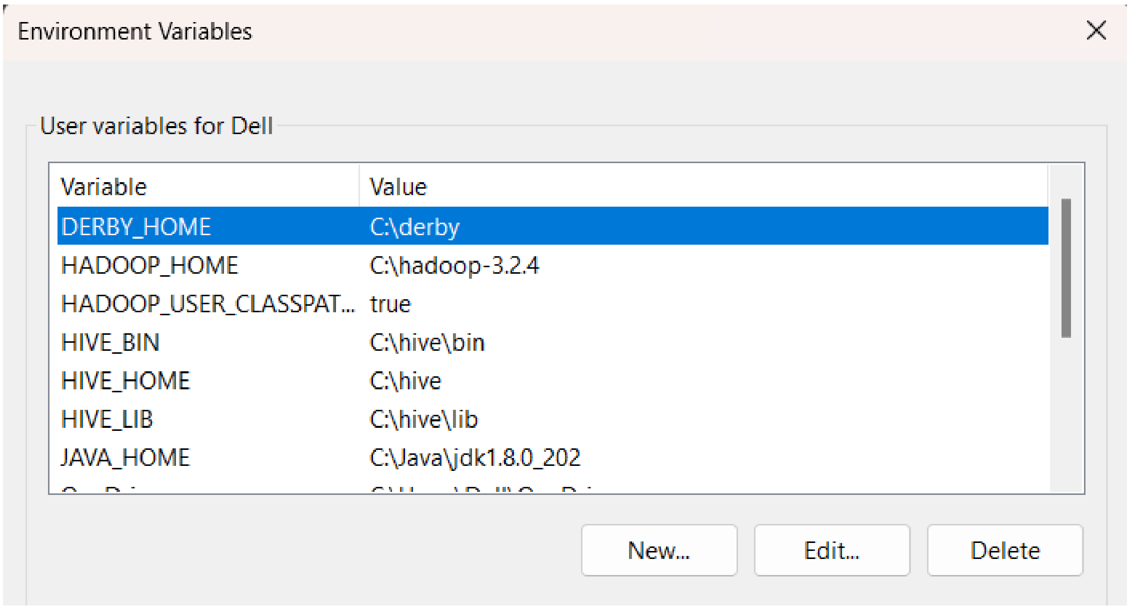
- Download a Hive configuration file (hive-site.xml) in the conf folder of the extracted hive.

Download link: https://drive.google.com/file/d/1tsBbHdvM1fFktmn9O0-u0pbG1vWWFoyE/view

- Copy the library files in the derby lib folder and paste them into the hive lib folder.
- Replace the bin folder in Hive with the fixed bin folder from this website: GitHub - HadiFadl/Hive-cmd: All cmd files needed to run Hive on windows (taken from https://svn.apache.org/repos/asf/hive/trunk/bin/)

4. Environment Variable Setup:

- Add the HIVE binary directory (C:\hive\bin) to your PATH environment variable.
- Set the HIVE_HOME, HIVE_LIB and HIVE_BIN environment variables.
- Add the Path variable HADOOP_USER_CLASSPATH_FIRST = true in both User and System variables.



Environment Variables

User variables for Dell

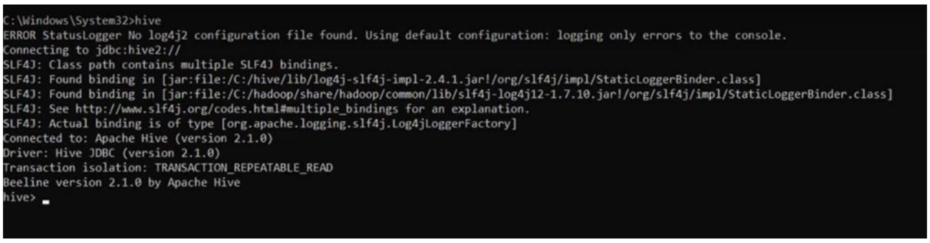
Variable	Value
DERBY_HOME	C:\derby
HADOOP_HOME	C:\hadoop-3.2.4
HADOOP_USER_CLASSPAT...	true
HIVE_BIN	C:\hive\bin
HIVE_HOME	C:\hive
HIVE_LIB	C:\hive\lib
JAVA_HOME	C:\Java\jdk1.8.0_202

6. Start Hive:

- Run Hadoop in the command prompt.
- Open a new command prompt and navigate to the Derby installation directory i.e., C:\derby\bin and execute the command:

  startNetworkServer -h 0.0.0.0
- Open a new command prompt and navigate to the Hive installation directory i.e., C:\hive\bin and execute the command:

  hive



```
C:\Windows\System32>hive
ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.
Connecting to jdbc:hive2://
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.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]
Connected to: Apache Hive (version 2.1.0)
Driver: Hive JDBC (version 2.1.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.1.0 by Apache Hive
hive> _
```

EXAMPLES:

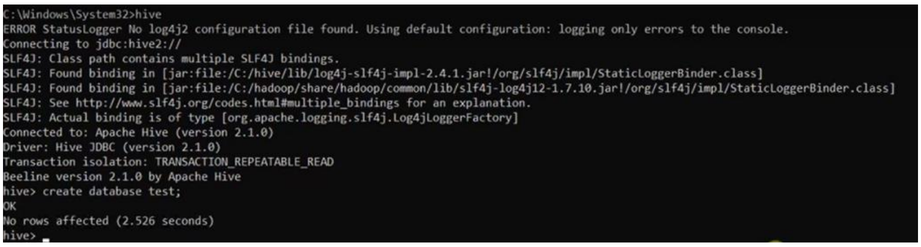
1. Create a Database:

To create a new database HIVE, use the following syntax:

CREATE DATABASE database_name

Example:

CREATE DATABASE test;

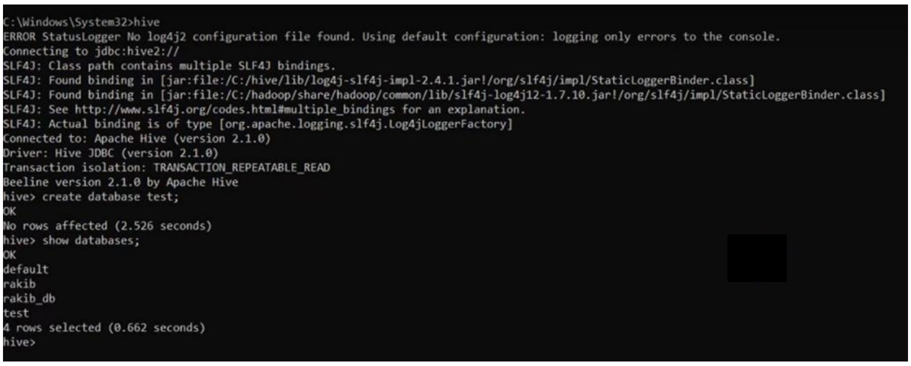


```
C:\Windows\System32>hive
ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.
Connecting to jdbc:hive2://
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.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]
Connected to: Apache Hive (version 2.1.0)
Driver: Hive JDBC (version 2.1.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.1.0 by Apache Hive
hive> create database test;
OK
No rows affected (2.526 seconds)
hive> _
```

2. Show Database:

To display a list of available databases in HIVE, use the following syntax:

SHOW DATABASES;



```
C:\Windows\System32>hive
ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.
Connecting to jdbc:hive2://
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.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]
Connected to: Apache Hive (version 2.1.0)
Driver: Hive JDBC (version 2.1.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.1.0 by Apache Hive
hive> create database test;
OK
No rows affected (2.526 seconds)
hive> show databases;
OK
default
rakib
rakib_db
test
4 rows selected (0.662 seconds)
hive>
```

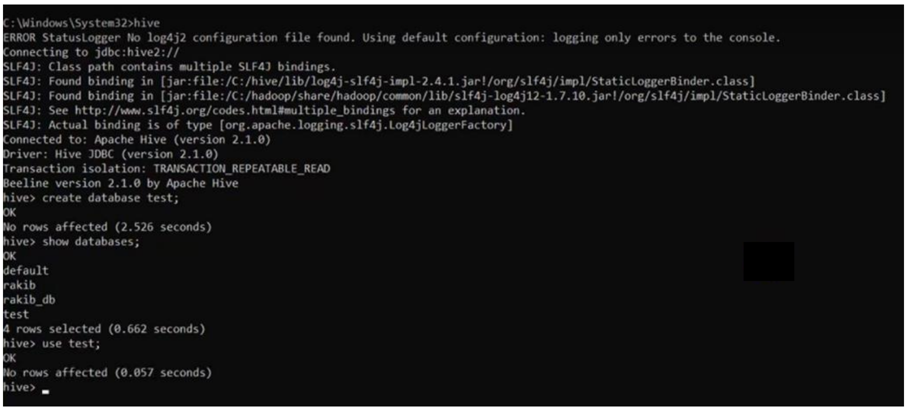
3. Use a Database:

To use a specific database in HIVE, use the following syntax:

USE database_name;

Example:

USE test;



```
C:\Windows\System32>hive
ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.
Connecting to jdbc:hive2://
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.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]
Connected to: Apache Hive (version 2.1.0)
Driver: Hive JDBC (version 2.1.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.1.0 by Apache Hive
hive> create database test;
OK
No rows affected (2.526 seconds)
hive> show databases;
OK
default
rakib
rakib_db
test
4 rows selected (0.662 seconds)
hive> use test;
OK
No rows affected (0.057 seconds)
hive> 
```

4. Create a Table:

To create a table in HIVE, use the following syntax{

CREATE TABLE table_name(column1 datatype, column 2 datatype, …);

Example:

CREATE TABLE student(id int, name string, age int);

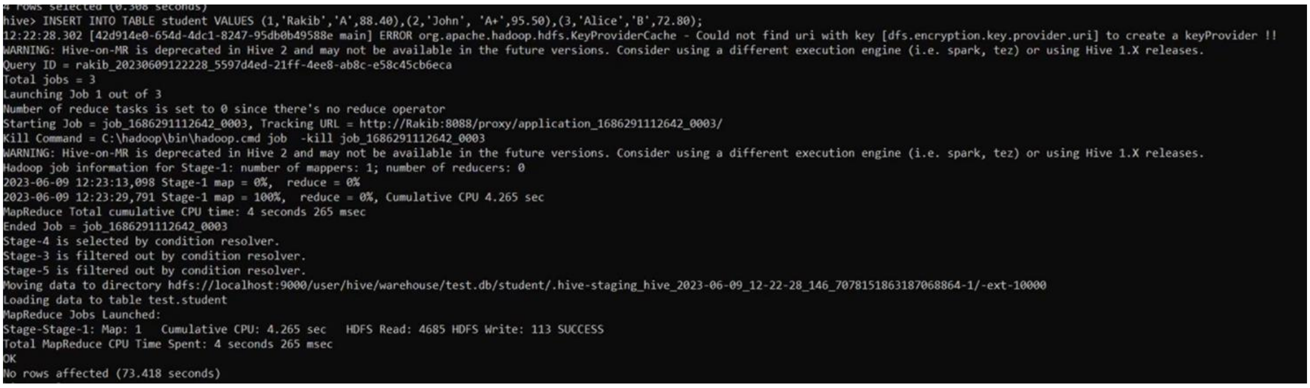
5. Insert data into a Table:

To insert data into a table in HIVE, use the following syntax:

INSERT INTO table_name (column1, column2, ..) VALUES(value1, value2, ..);

Example:

INSERT INTO student(id, name, age) VALUES(1, ‘John Doe’, 25);



```
4 rows selected (0.308 seconds)
hive> INSERT INTO TABLE student VALUES (1,'Rakib','A',88.40),(2,'John', 'A+',95.50),(3,'Alice','B',72.80);
12:22:28.302 [42d914e0-654d-4dc1-8247-95db0b49588e main] ERROR org.apache.hadoop.hdfs.KeyProviderCache - Could not find uri with key [dfs.encryption.key.provider.uri] to create a keyProvider !!
WARNING: 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.
Query ID = rakib_20230609122228_5597d4ed-21ff-4ee8-ab8c-e58c45cb6eca
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1686291112642_0003, Tracking URL = http://Rakib:8088/proxy/application_1686291112642_0003/
Kill Command = C:\hadoop\bin\hadoop.cmd job  -kill job_1686291112642_0003
WARNING: 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.
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-06-09 12:23:13,098 Stage-1 map = 0%,  reduce = 0%
2023-06-09 12:23:29,791 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.265 sec
MapReduce Total cumulative CPU time: 4 seconds 265 msec
Ended Job = job_1686291112642_0003
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/test.db/student/.hive-staging_hive_2023-06-09_12-22-28_146_7078151863187068864-1/-ext-10000
Loading data to table test.student
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 4.265 sec   HDFS Read: 4685 HDFS Write: 113 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 265 msec
OK
No rows affected (73.418 seconds)
```

6. Select Data from a Table:

To select data from a table in HIVE, use the following syntax:

SELECT * FROM table_name;

Example:

SELECT * FROM student;

RESULT:

Thus, the installation of HIVE along with practice examples was executed successfully.

Ex No: 6

DATE:

INSTALLATION OF HBASE, INSTALLING THRIFT WITH PRACTICE

AIM:

To install HBase and Thrift along with practice examples.

PROCEDURE:

Install Java:

- Open the terminal in Ubuntu and update the package list using the command

  sudo apt update
- install OpenJDK(Java Development Kit) using command

  sudo apt install openjdk-8-jdk
- Verify the Java installation:

  java –version

Installation of HBase:

Step 1: Download and extract HBase

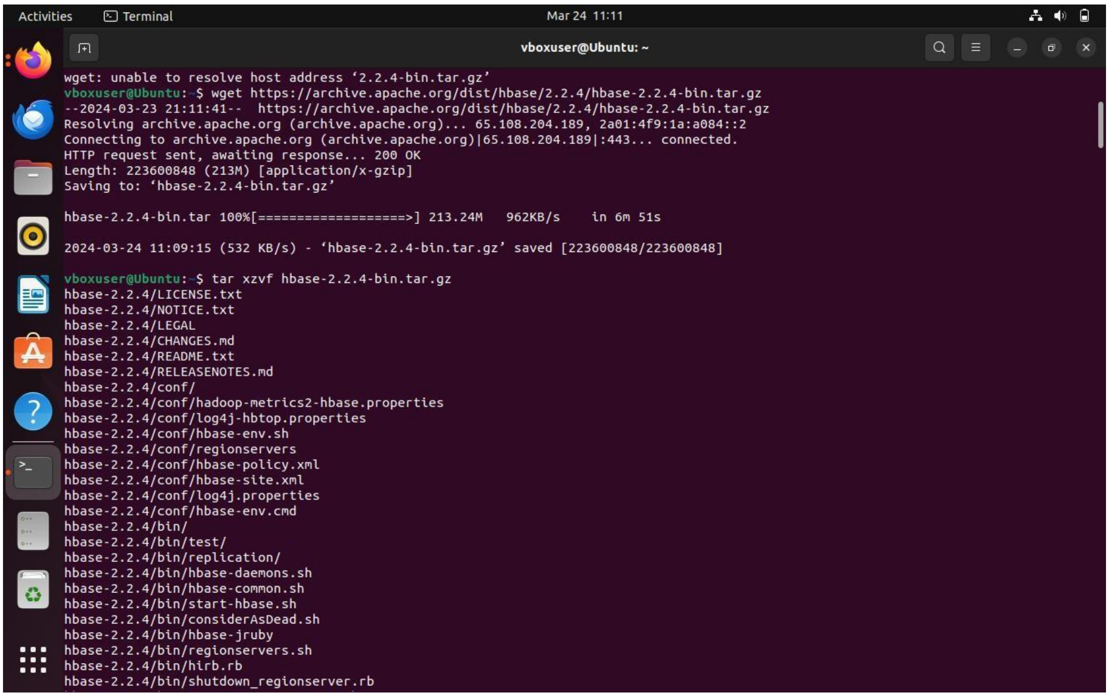
- Make a directory for our HBase data

  sudo mkdir -p /var/hbase
- Head over to the Apache download mirrors site and click the recommended mirror. Once there we want to find the latest HBase version and locate the version ending in -bin.tar.gz. Once we have that we can use wget to download it.

  Wget https://www.mirrorservice.org/sites/ftp.apache.org/hbase/2.2.4/hbase-2.2.4-bin.tar.gz
- Extract the downloaded archive

  tar xzvf hbase-2.2.4-bin.tar.gz
- And move into the extracted directory

  cd hbase-2.2.4



```
wget: unable to resolve host address ‘2.2.4-bin.tar.gz’
vboxuser@Ubuntu:~$ wget https://archive.apache.org/dist/hbase/2.2.4/hbase-2.2.4-bin.tar.gz
--2024-03-23 21:11:41--  https://archive.apache.org/dist/hbase/2.2.4/hbase-2.2.4-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 223600848 (213M) [application/x-gzip]
Saving to: ‘hbase-2.2.4-bin.tar.gz’

hbase-2.2.4-bin.tar 100%[===================>] 213.24M   962KB/s    in 6m 51s

2024-03-24 11:09:15 (532 KB/s) - ‘hbase-2.2.4-bin.tar.gz’ saved [223600848/223600848]

vboxuser@Ubuntu:~$ tar xzvf hbase-2.2.4-bin.tar.gz
hbase-2.2.4/LICENSE.txt
hbase-2.2.4/NOTICE.txt
hbase-2.2.4/LEGAL
hbase-2.2.4/CHANGES.md
hbase-2.2.4/README.txt
hbase-2.2.4/RELEASENOTES.md
hbase-2.2.4/conf/
hbase-2.2.4/conf/hadoop-metrics2-hbase.properties
hbase-2.2.4/conf/log4j-hbtop.properties
hbase-2.2.4/conf/hbase-env.sh
hbase-2.2.4/conf/regionservers
hbase-2.2.4/conf/hbase-policy.xml
hbase-2.2.4/conf/hbase-site.xml
hbase-2.2.4/conf/log4j.properties
hbase-2.2.4/conf/hbase-env.cmd
hbase-2.2.4/bin/
hbase-2.2.4/bin/test/
hbase-2.2.4/bin/replication/
hbase-2.2.4/bin/hbase-daemons.sh
hbase-2.2.4/bin/hbase-common.sh
hbase-2.2.4/bin/start-hbase.sh
hbase-2.2.4/bin/considerAsDead.sh
hbase-2.2.4/bin/hbase-jruby
hbase-2.2.4/bin/regionservers.sh
hbase-2.2.4/bin/hirb.rb
hbase-2.2.4/bin/shutdown_regionserver.rb
```

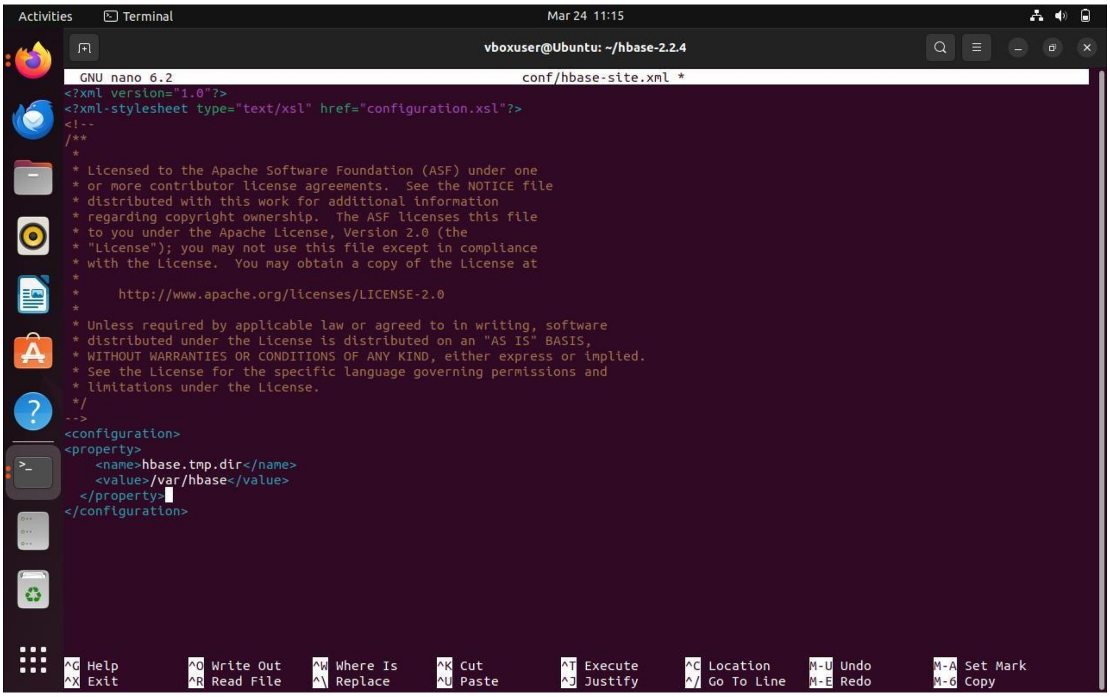
Step 3: Configure HBase

- Open the hbase-site.xml file in the conf folder for editing

sudo nano conf/hbase-site.xml

- Add the following in the .xml file

```
<configuration>
<property>
<name>hbase.tmp.dir</name>
<value>/var/hbase</value>
 </property>
 </configuration>
</property>
```



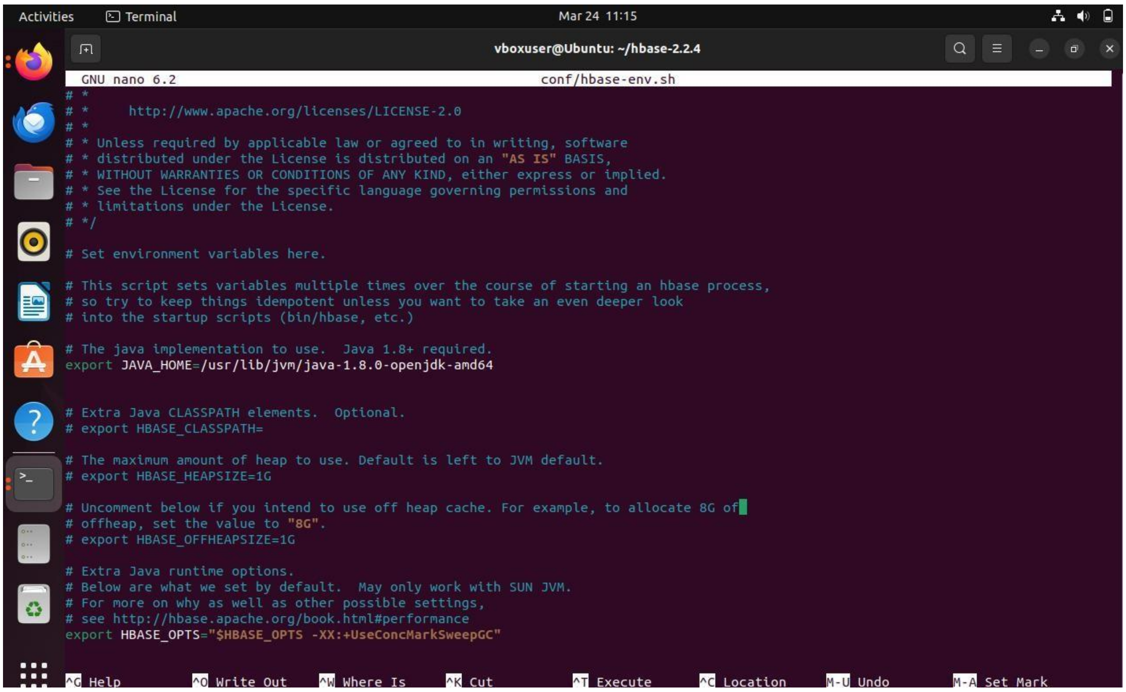
```
  GNU nano 6.2                                    conf/hbase-site.xml *
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
/**
 *
 * Licensed to the Apache Software Foundation (ASF) under one
 * or more contributor license agreements.  See the NOTICE file
 * distributed with this work for additional information
 * regarding copyright ownership.  The ASF licenses this file
 * to you 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.
 */
-->
<configuration>
<property>
    <name>hbase.tmp.dir</name>
    <value>/var/hbase</value>
  </property>
</configuration>
```

- Open the hbase-env file in the same folder and add the path to your jdk

sudo nano conf/hbase-env.sh

- Set JAVA_HOME to your Java installation directory and then save and exit

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre



```
  GNU nano 6.2                                    conf/hbase-env.sh
# *
# *     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.
# */

# Set environment variables here.

# This script sets variables multiple times over the course of starting an hbase process,
# so try to keep things idempotent unless you want to take an even deeper look
# into the startup scripts (bin/hbase, etc.)

# The java implementation to use.  Java 1.8+ required.
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64


# Extra Java CLASSPATH elements.  Optional.
# export HBASE_CLASSPATH=

# The maximum amount of heap to use. Default is left to JVM default.
# export HBASE_HEAPSIZE=1G

# Uncomment below if you intend to use off heap cache. For example, to allocate 8G of
# offheap, set the value to "8G".
# export HBASE_OFFHEAPSIZE=1G

# Extra Java runtime options.
# Below are what we set by default.  May only work with SUN JVM.
# For more on why as well as other possible settings,
# see http://hbase.apache.org/book.html#performance
export HBASE_OPTS="$HBASE_OPTS -XX:+UseConcMarkSweepGC"
```

Step 4: Start HBase

- Open terminal and navigate to your hbase folder

cd %HBASE_HOME%

- Start HBase using the command

```
sudo ./bin/start-hbase.sh
```

- Open HBase shell using the command

```
./bin/hbase shell
```

Practice examples:

Create a table

- In the HBase shell, you can create a table with column families
- For example, let’s create a table named “my_table” with a column family called “cf”

```
>> create ‘my_table’,’cf’
```

Insert Data

- Insert data into the table, you can use the put command.
- Here’s an example of inserting a row with a specific row key and values

```
>> put ‘my_table’,’row1’,’cf:column1’,’value1’

>>put ‘my_table’,’row1’,’cf:column2’,’value2’
```

Get Data

- You can retrieve data from the table using the get command.
- For example, to get the values of a specific row

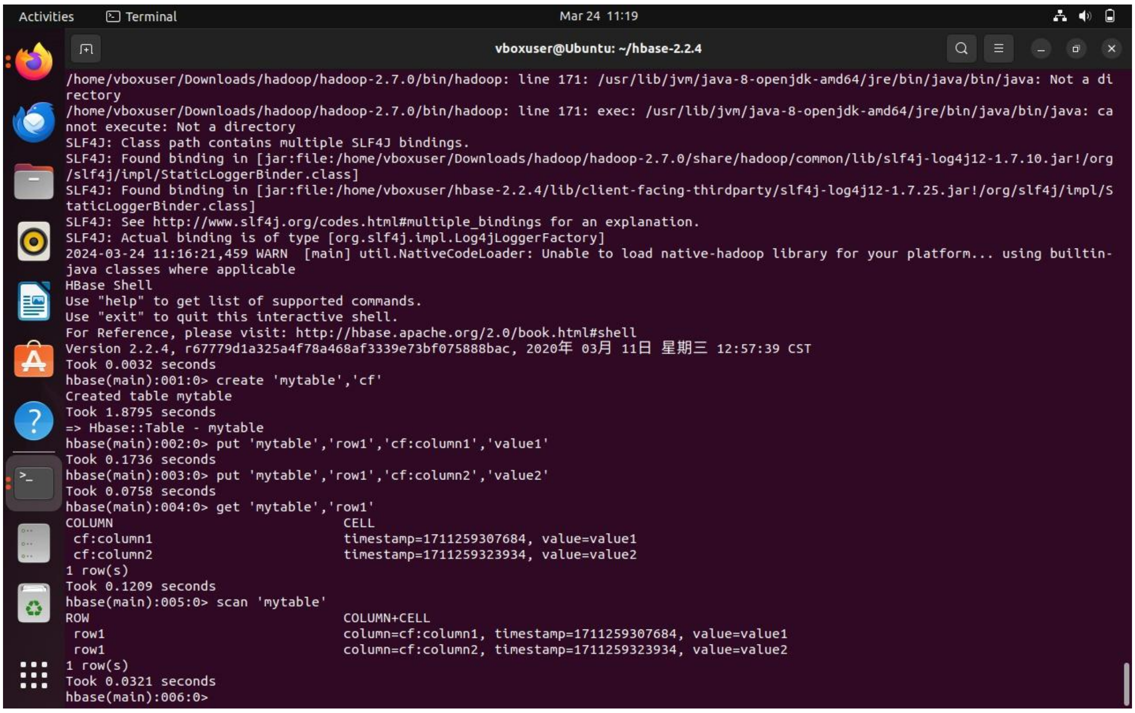
```
>> get ‘my_table’,’row1’
```

- This will display all the column family values for the specific row.

Disable and Drop Table

- If you want to remove the table entirely, you need to disable and drop it.
- Use the following commands

```
>> disable ‘my_table’

>>drop ‘my_table’
```



```
/home/vboxuser/Downloads/hadoop/hadoop-2.7.0/bin/hadoop: line 171: /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java/bin/java: Not a di
rectory
/home/vboxuser/Downloads/hadoop/hadoop-2.7.0/bin/hadoop: line 171: exec: /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java/bin/java: ca
nnot execute: Not a directory
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/vboxuser/Downloads/hadoop/hadoop-2.7.0/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org
/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/vboxuser/hbase-2.2.4/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/S
taticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
2024-03-24 11:16:21,459 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-
java classes where applicable
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 2.2.4, r67779d1a325a4f78a468af3339e73bf075888bac, 2020年 03月 11日 星期三 12:57:39 CST
Took 0.0032 seconds
hbase(main):001:0> create 'mytable','cf'
Created table mytable
Took 1.8795 seconds
=> Hbase::Table - mytable
hbase(main):002:0> put 'mytable','row1','cf:column1','value1'
Took 0.1736 seconds
hbase(main):003:0> put 'mytable','row1','cf:column2','value2'
Took 0.0758 seconds
hbase(main):004:0> get 'mytable','row1'
COLUMN                                   CELL
 cf:column1                              timestamp=1711259307684, value=value1
 cf:column2                              timestamp=1711259323934, value=value2
1 row(s)
Took 0.1209 seconds
hbase(main):005:0> scan 'mytable'
ROW                                      COLUMN+CELL
 row1                                    column=cf:column1, timestamp=1711259307684, value=value1
 row1                                    column=cf:column2, timestamp=1711259323934, value=value2
1 row(s)
Took 0.0321 seconds
hbase(main):006:0>
```

Installation of Thrift:

Step 1: Install dependencies

- To use Java you will need to install Apache Ant

sudo apt-get install ant

- Installing required tools and libraries

sudo apt-get install libboost-dev libboost-test-dev libboost-program-options-dev libboost-filesystem-dev libboost-thread-dev libevent-dev automake libtool flex bison pkg-config g++ libssl-dev

- You can check for specific requirements for each language you wish to use here: http://thrift.apache.org/docs/install/

Step 2: Download Thrift

- Download Thrift: http://thrift.apache.org/download
- Copy the downloaded file into the desired directory and untar the file

tar -xvf thrift-0.9.3.tar.gz

Step 3: Configure Thrift

- For an Ubuntu linux distribution you just need to go to the thrift directory and type

./bootstrap.sh

./configure

- At the end of the output you should be able to see a list of all the libraries that are currently built in your system and ready to use with your desired programming languages. If a component is missing you should download the missing language and repeat the above step.
- Here http://thrift.apache.org/docs/install/debian/ you can find all the packages you might need to support your desired language in case some of them are missing.

Step 4: Build and Run Thrift

- On the same directory run make to build Thrift

sudo make

- (Optional) Run the test suite if you want

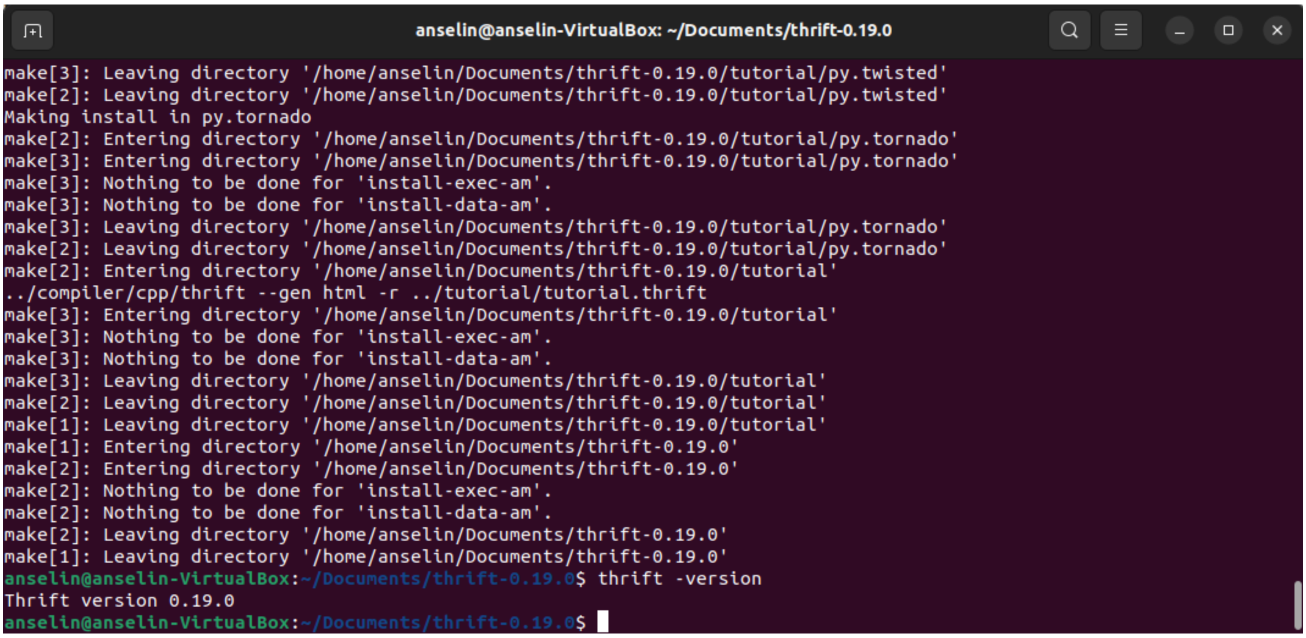
sudo make check

- And finally you are ready to install Thrift by running

sudo make install

- To verify that you have succesfully installed Thrift just type

thrift –version



```
make[3]: Leaving directory '/home/anselin/Documents/thrift-0.19.0/tutorial/py.twisted'
make[2]: Leaving directory '/home/anselin/Documents/thrift-0.19.0/tutorial/py.twisted'
Making install in py.tornado
make[2]: Entering directory '/home/anselin/Documents/thrift-0.19.0/tutorial/py.tornado'
make[3]: Entering directory '/home/anselin/Documents/thrift-0.19.0/tutorial/py.tornado'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/anselin/Documents/thrift-0.19.0/tutorial/py.tornado'
make[2]: Leaving directory '/home/anselin/Documents/thrift-0.19.0/tutorial/py.tornado'
make[2]: Entering directory '/home/anselin/Documents/thrift-0.19.0/tutorial'
../compiler/cpp/thrift --gen html -r ../tutorial/tutorial.thrift
make[3]: Entering directory '/home/anselin/Documents/thrift-0.19.0/tutorial'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/home/anselin/Documents/thrift-0.19.0/tutorial'
make[2]: Leaving directory '/home/anselin/Documents/thrift-0.19.0/tutorial'
make[1]: Leaving directory '/home/anselin/Documents/thrift-0.19.0/tutorial'
make[1]: Entering directory '/home/anselin/Documents/thrift-0.19.0'
make[2]: Entering directory '/home/anselin/Documents/thrift-0.19.0'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/anselin/Documents/thrift-0.19.0'
make[1]: Leaving directory '/home/anselin/Documents/thrift-0.19.0'
anselin@anselin-VirtualBox:~/Documents/thrift-0.19.0$ thrift -version
Thrift version 0.19.0
anselin@anselin-VirtualBox:~/Documents/thrift-0.19.0$
```

RESULT:

Thus, the installation of HBase with some practice examples and the installation of Thrift was executed successfully.

Ex No: 7

DATE:

PERFORM IMPORTING AND EXPORTING DATA FROMVARIOUS DATABASES

AIM:

To perform importing and exporting data from various databases.

PRE-REQUISITE:

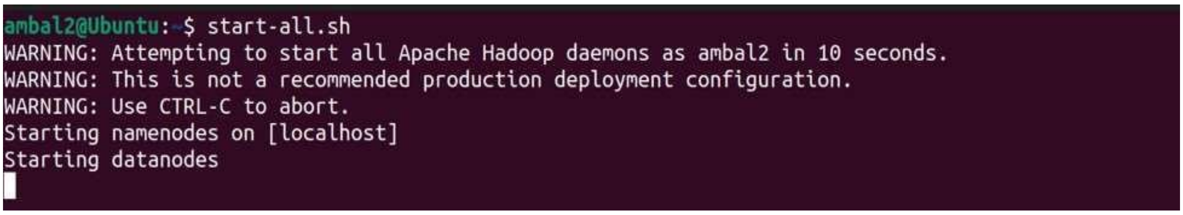
Hadoop and JavaHive

MySQL

Sqoop

PROCEDURE:

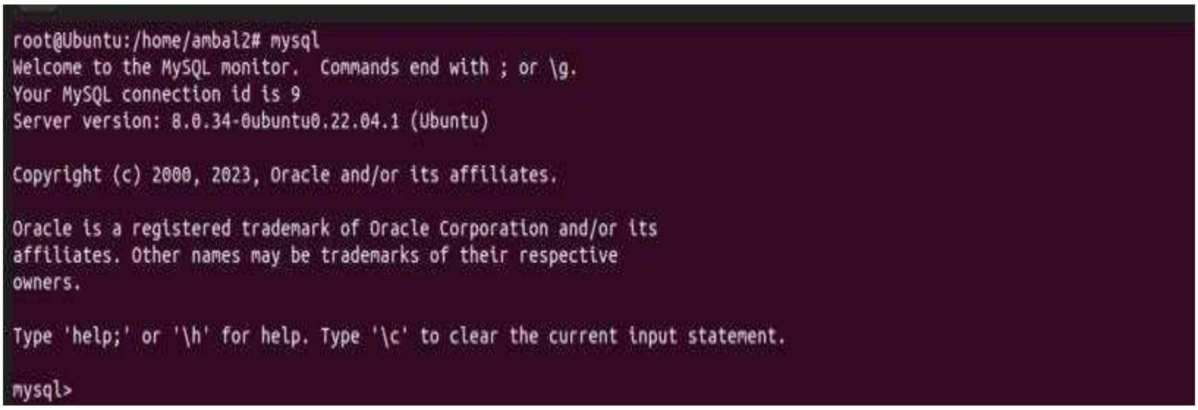
Step 1: To start hdfs



```
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 Command to install MySQL server

sudo apt 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>
```

COMMAND:

~$sudo su

After this enter your linux user password, then the root mode will be open here we don’tneed any authentication for mysql.

~root$ mysql

Creating user profile and grant permission:

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 operationsin 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 tables and insert data

The following command is used to create database table.

CREATE DATABASE dell;

CREATE TABLE dell (name var char(10) , id int);

Now the dell table is created and insert the values in that data.

INSERT VALUES INTO dell(‘inspiron’,3505);

INSERT VALUES INTO dell(‘alienware’,5005);

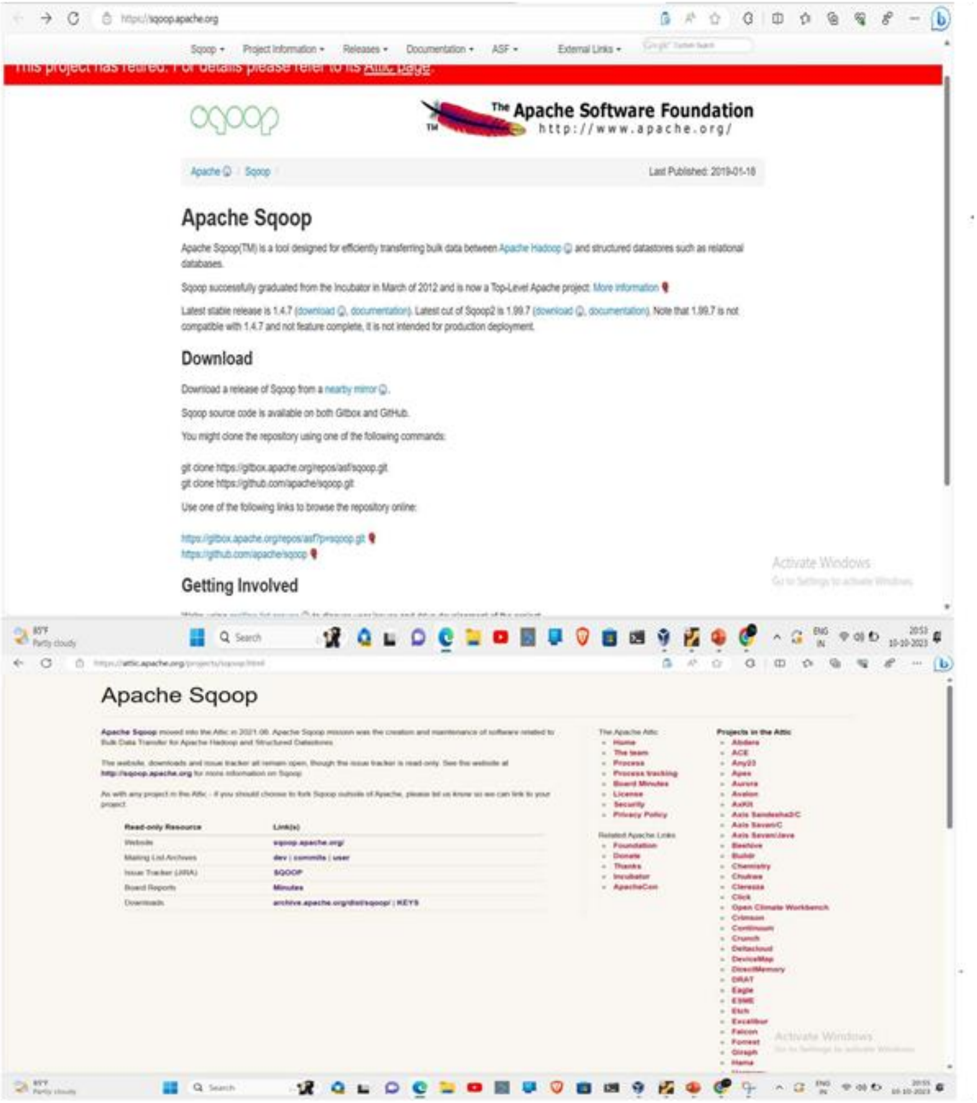
INSERT VALUES INTO dell(‘inspiron’,3550);

Now three rows are inserted into the dell table.

Step 4: SQOOP installation

First you need to install Sqoop on your Hadoop cluster or machine.

Download the latest version of Sqoop from the Apache Sqoop website(http://sqoop.apache.org/)and follow the installation instructions provided in the documentation.



https://sqoop.apache.org
Sqoop ▾ Project Information ▾ Releases ▾ Documentation ▾ ASF ▾ External Links ▾
sqoop
The Apache Software Foundation
http://www.apache.org/
Apache / Sqoop
Activate Windows
Go to Settings to activate Windows.
85°F Partly cloudy
Search
ENG IN
20:53 10-10-2023
https://attic.apache.org/projects/sqoop.html

This project has retired. For details please refer to its Attic page.

Last Published: 2019-01-18

Apache Sqoop

Apache Sqoop(TM) is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.

Sqoop successfully graduated from the Incubator in March of 2012 and is now a Top-Level Apache project: More information

Latest stable release is 1.4.7 (download, documentation). Latest cut of Sqoop2 is 1.99.7 (download, documentation). Note that 1.99.7 is not compatible with 1.4.7 and not feature complete, it is not intended for production deployment.

Download

Download a release of Sqoop from a nearby mirror.

Sqoop source code is available on both Gitbox and GitHub.

You might clone the repository using one of the following commands:

```
git clone https://gitbox.apache.org/repos/asf/sqoop.git
git clone https://github.com/apache/sqoop.git
```

Use one of the following links to browse the repository online:

https://gitbox.apache.org/repos/asf?p=sqoop.git
https://github.com/apache/sqoop

Getting Involved

Apache Sqoop

Apache Sqoop moved into the Attic in 2021-06. Apache Sqoop mission was the creation and maintenance of software related to Bulk Data Transfer for Apache Hadoop and Structured Datastores.

The website, downloads and issue tracker all remain open, though the issue tracker is read-only. See the website at http://sqoop.apache.org for more information on Sqoop.

As with any project in the Attic - if you should choose to fork Sqoop outside of Apache, please let us know so we can link to your project.

Read-only Resource	Link(s)
Website	sqoop.apache.org/
Mailing List Archives	dev | commits | user
Issue Tracker (JIRA)	SQOOP
Board Reports	Minutes
Downloads	archive.apache.org/dist/sqoop/ | KEYS

The Apache Attic
- Home
- The team
- Process
- Process tracking
- Board Minutes
- License
- Security
- Privacy Policy

Related Apache Links
- Foundation
- Donate
- Thanks
- Incubator
- ApacheCon

Projects in the Attic
- Abdera
- ACE
- Any23
- Apex
- Aurora
- Avalon
- AxKit
- Axis Sandesha2/C
- Axis Savan/C
- Axis Savan/Java
- Beehive
- Buildr
- Chemistry
- Chukwa
- Clerezza
- Click
- Open Climate Workbench
- Crimson
- Continuum
- Crunch
- Deltacloud
- DeviceMap
- DirectMemory
- DRAT
- Eagle
- ESME
- Etch
- Excalibur
- Falcon
- Forrest
- Giraph
- Hama

After downloading the Sqoop, go to the directory where we downloaded the Sqoop and then extractit 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/sqoopThen

exit: $ exit

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

export SQOOP_HOME=/usr/lib/sqoop

export PATH=$ PATH:$SQOOP_HOME/bin

$ source ~/.bashrc

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

```
$ cd $SQOOP_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
```

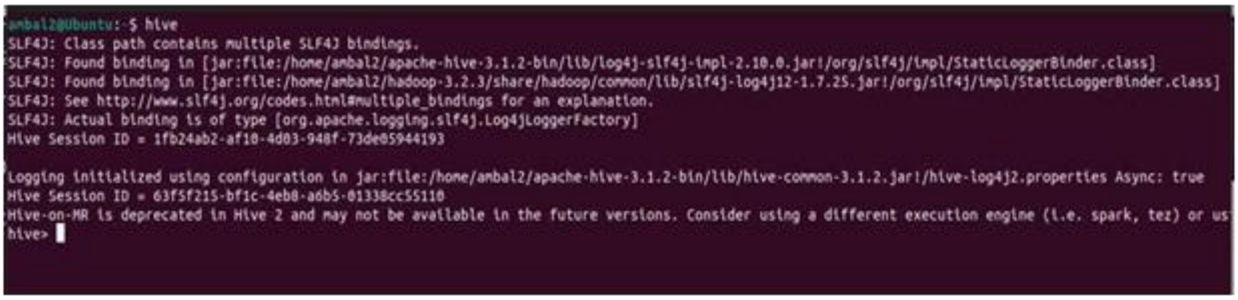
Note : This is library file is very important don’t skip this step because it contains the librariesto connect the mysql databases to jdbc.

Verify sqoop: sqoop-version

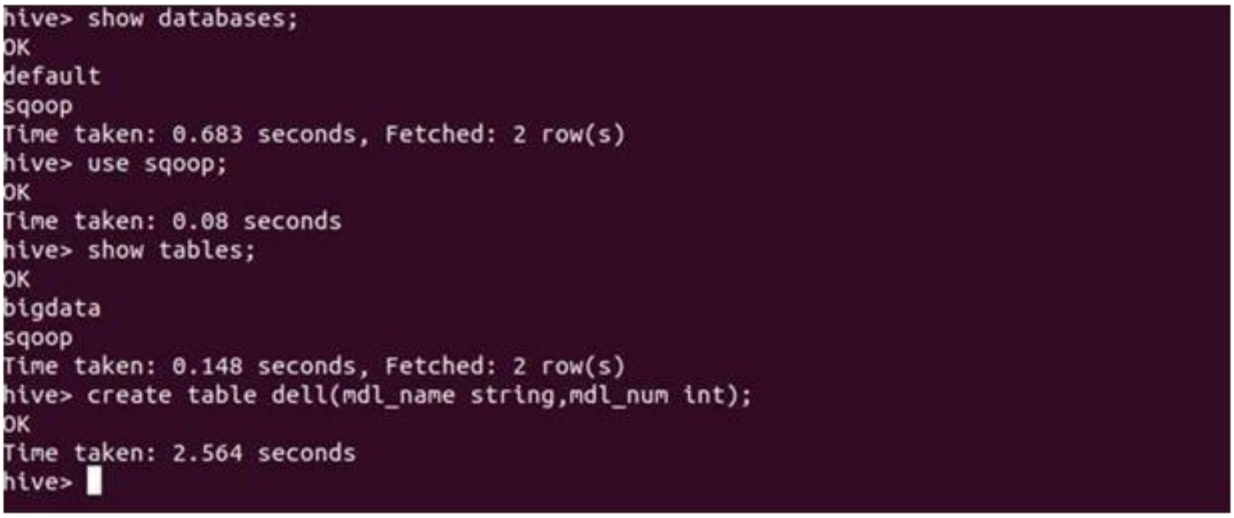
Step 6: Hive database creation

```
hive> create database sqoop_example;hive>

use sqoop_example;

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

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

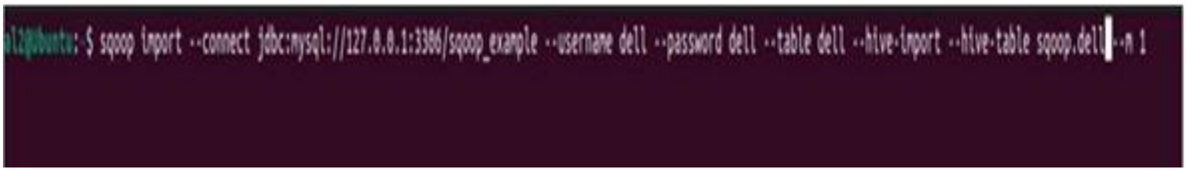
Step 7: Importing data from MySQL to hive:

Sqoop importing –connect\

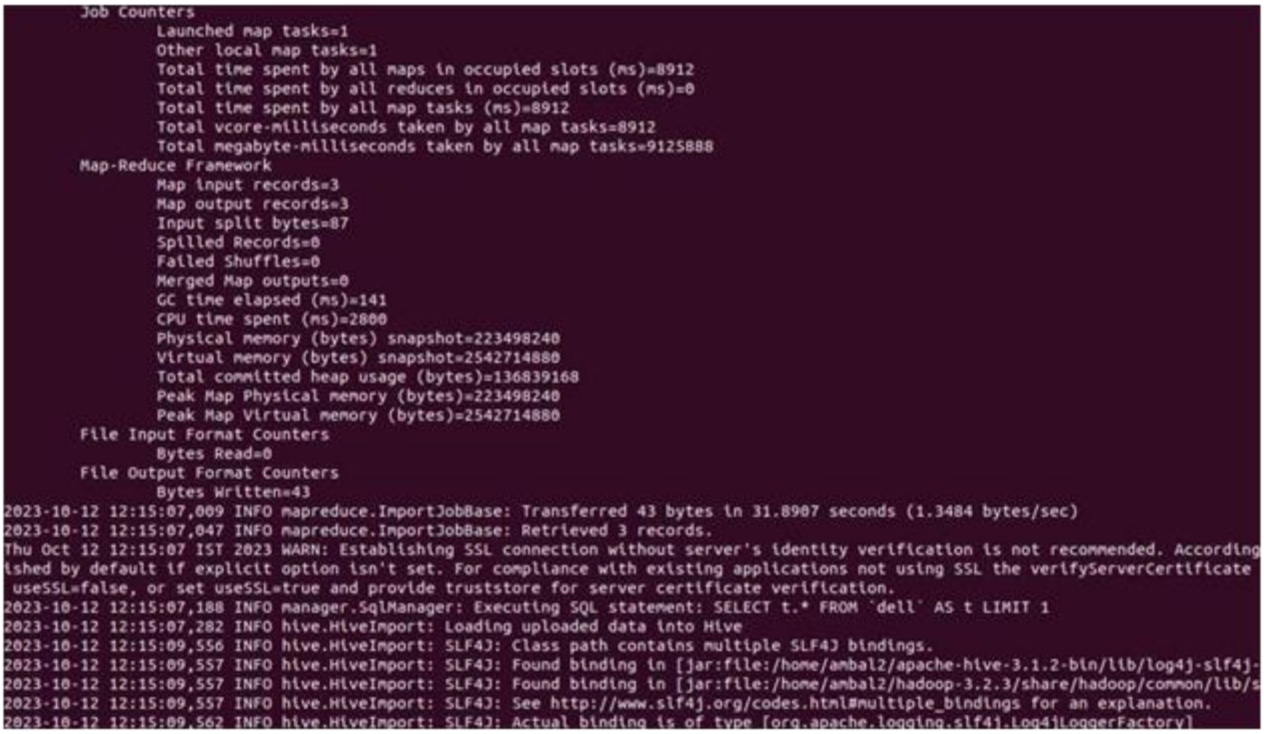
jdbc : mysql://127.0.0.1:3306/database_name_in_hive \

--username root –password cloudera \

--table table_name_in_mysql \

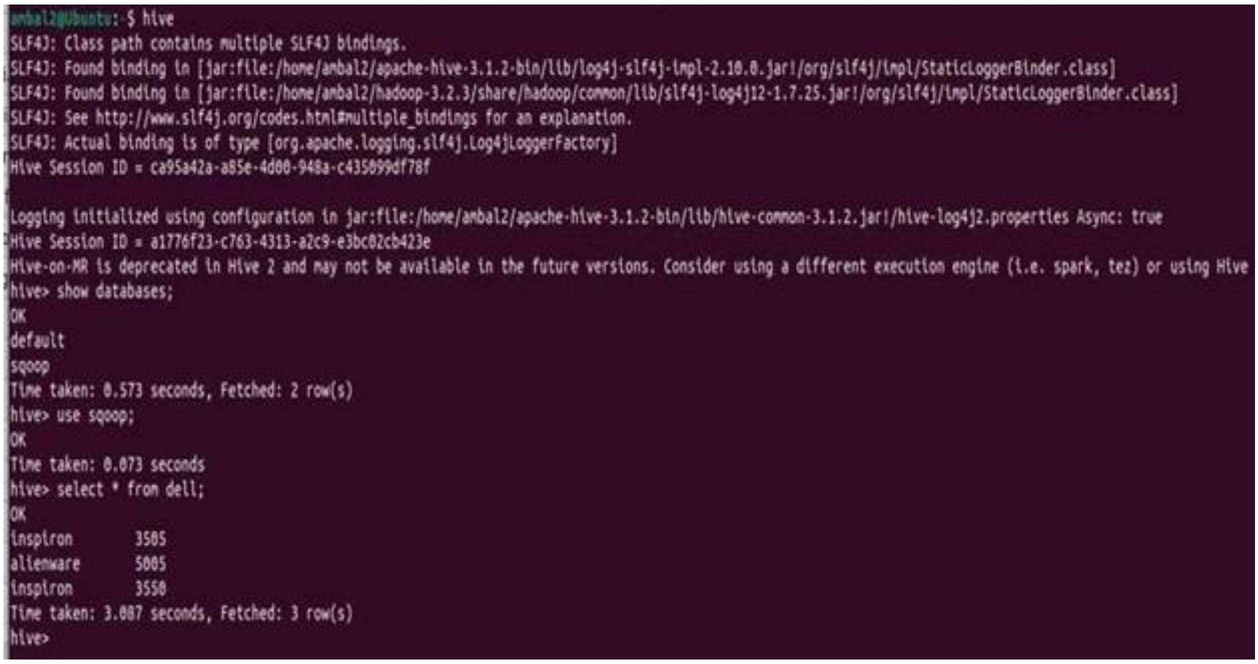


```
al2@Ubuntu:~$ sqoop import --connect jdbc:mysql://127.0.0.1:3306/sqoop_example --username dell --password dell --table dell --hive-import --hive-table sqoop.dell --m 1
```



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

Step 8: Exporting data to a database



```
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, to import and export data from mysql to hive using Sqoop was executed successfully.