

TABLE OF CONTENTS

S.NO	DATE	TITLE OF THE EXPERIMENTS	PAGE NO	MARK	SIGN
1.		Download ingand Installing Hadoop;Understanding Different Hadoop Modes. Startup Scripts, Configuration Files.	1		
2.		Hadoop Implementation of File Management Tasks, Such as Adding Files and Directories,Retrieving Files and Deleting Files	12		
3.		Implement of Matrix Multiplication with Hadoop Map Reduce	15		
4.		Run a Basic Word Count Map Reduce Program to Understand Map Reduce Paradigm.	19		
5.		Installation of Hive along with Practice Examples.	24		
6.		Installation of Hbase along with Practice Examples	27		
7.		Installation of Thrift.	30		
8.		Practice Importing and Exporting Data FromVarious Databases.	31		

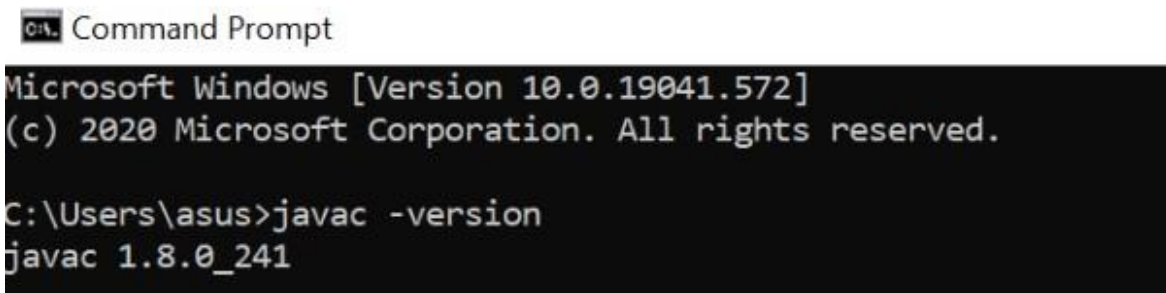
EX. No. 1: DOWNLOADING AND INSTALLING HADOOP; UNDERSTANDING DIFFERENT HADOOP MODES .START UP SCRIPTS, CONFIGURATION FILES.

PREREQUISITES TO INSTALL HADOOP ON WINDOWS

- **VIRTUALBOX** (For Linux): it is used for installing the operating system on it.
- **OPERATING SYSTEM**: You can install Hadoop on Windows or Linux based operating systems. Ubuntu and CentOS are very commonly used.
- **JAVA**: You need to install the Java8 package on your system.
- **HADOOP**: You require Hadoop latest version

1. Install Java

- Java JDK Link to download <https://www.oracle.com/java/technologies/javase-jdk8-downloads.html>
- Extract and install Java in C:\Java
- Open cmd and type -> javac -version



The screenshot shows a Windows Command Prompt window titled "C:\> Command Prompt". The text inside the window reads: "Microsoft Windows [Version 10.0.19041.572] (c) 2020 Microsoft Corporation. All rights reserved. C:\Users\asus>javac -version javac 1.8.0_241".

2. Download Hadoop

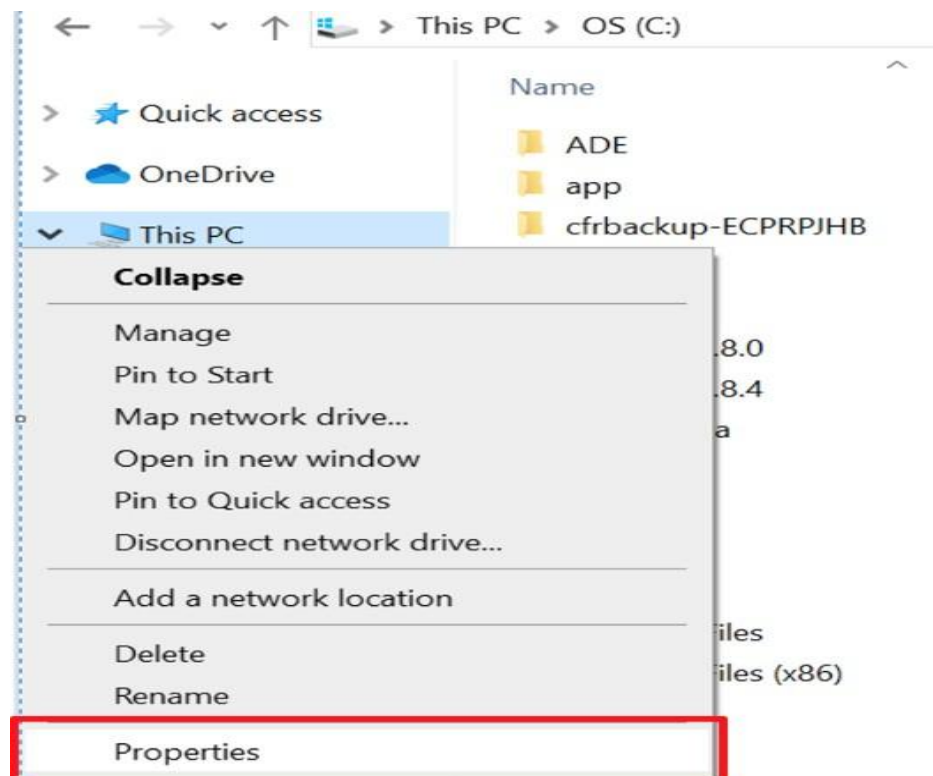
<https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz>

- extract to C:\Hadoop

ADE	1/26/2020 11:13 AM	File folder
app	1/26/2020 10:53 AM	File folder
cfrbackup-ECPRPJHB	4/18/2019 10:25 PM	File folder
eSupport	7/13/2017 5:22 AM	File folder
Games	8/20/2019 9:40 PM	File folder
hadoop	11/8/2020 3:15 PM	File folder
hadoop-2.8.0	12/10/2019 3:02 PM	File folder
hadoop-2.8.4	6/14/2019 9:36 PM	File folder
hadoop-3.3.0	11/8/2020 4:30 PM	File folder
Hortonwork	11/8/2020 2:40 PM	File folder
Informatica	1/28/2020 12:52 AM	File folder
Java	11/8/2020 3:25 PM	File folder
logs	3/27/2020 9:36 PM	File folder
oraclexe	1/29/2020 11:52 PM	File folder

3. SetthepathJAVA_HOME Environmentvariable

4. SetthepathHADOOP_HOMEEnvironmentvariable



Control Panel Home

- Device Manager
- Remote settings
- System protection
- Advanced system settings**

View basic information about your computer

Windows edition

Windows 10 Home Single Language

© 2019 Microsoft Corporation. All rights reserved.

System

Manufacturer:	ASUSTek Computer Inc.
Processor:	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz
Installed memory (RAM):	8.00 GB (7.89 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

ASUSTek Computer Inc. support

Website: [Online support](#)

Computer name, domain, and workgroup settings

Computer name:	DESKTOP-475FCII
Full computer name:	DESKTOP-475FCII
Computer description:	
Workgroup:	WORKGROUP

System Properties

Computer Name Hardware **Advanced** System Protection Remote

You must be logged on as an Administrator to make most of these changes.

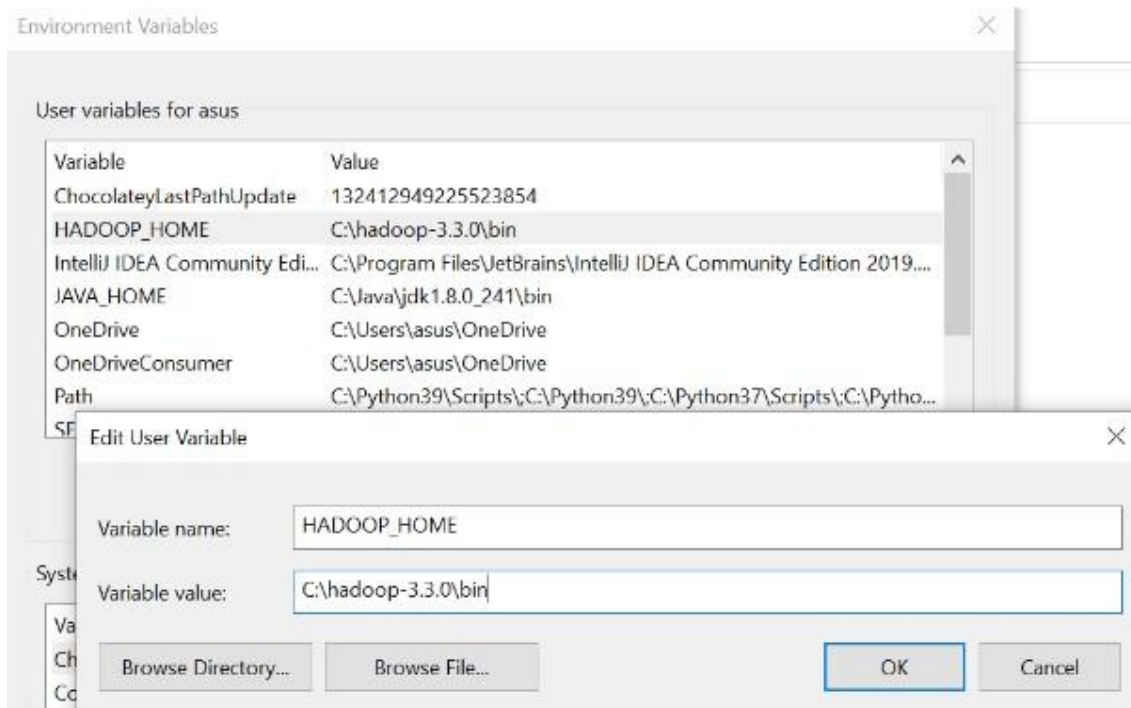
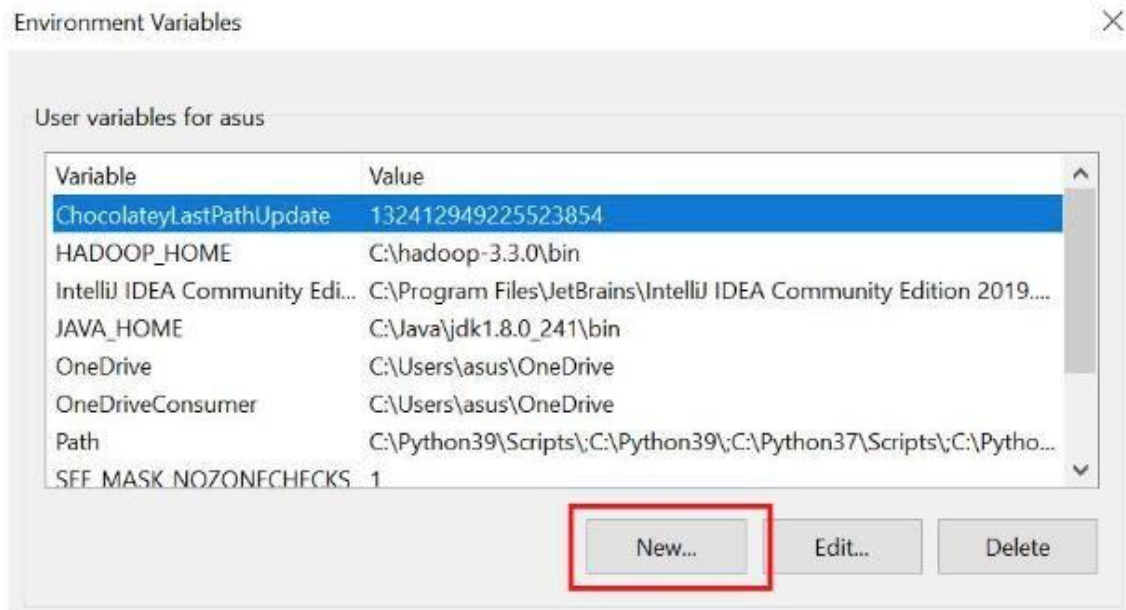
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
[Settings...](#)

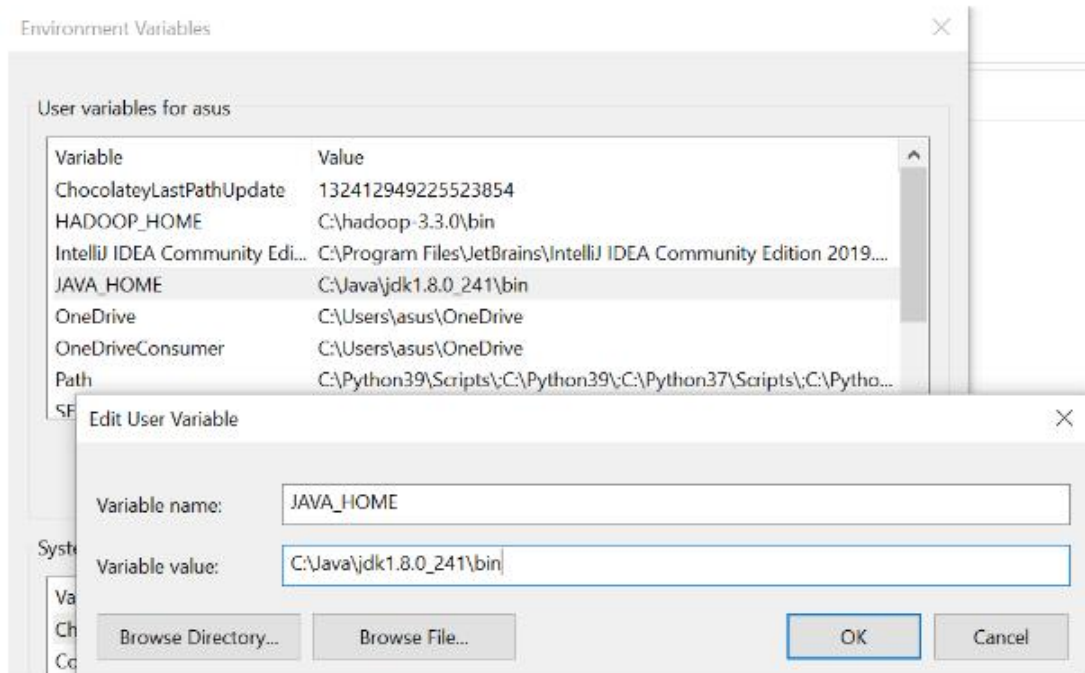
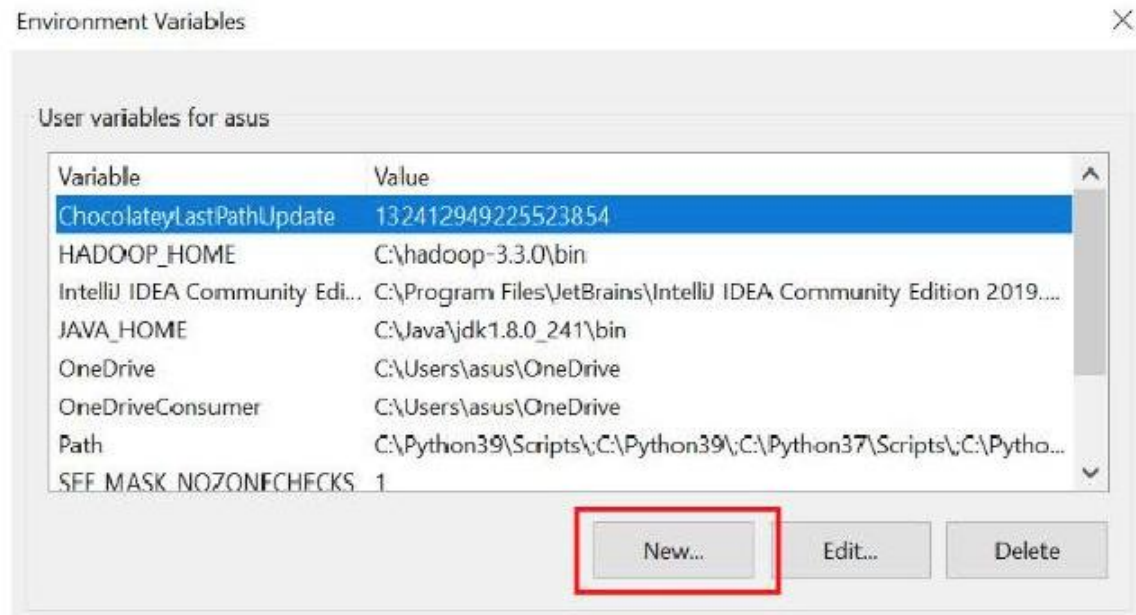
User Profiles
Desktop settings related to your sign-in
[Settings...](#)

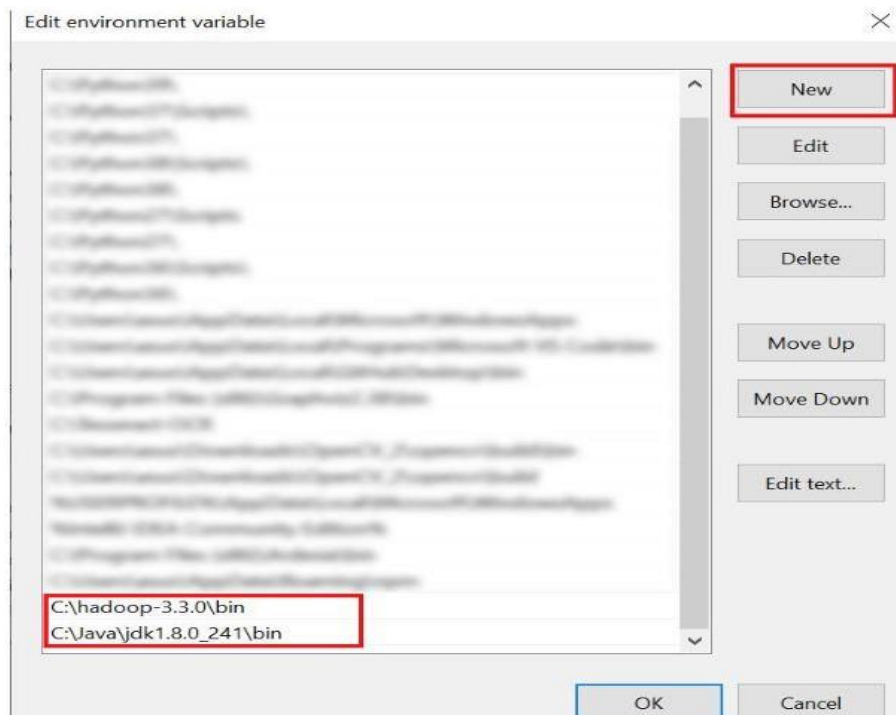
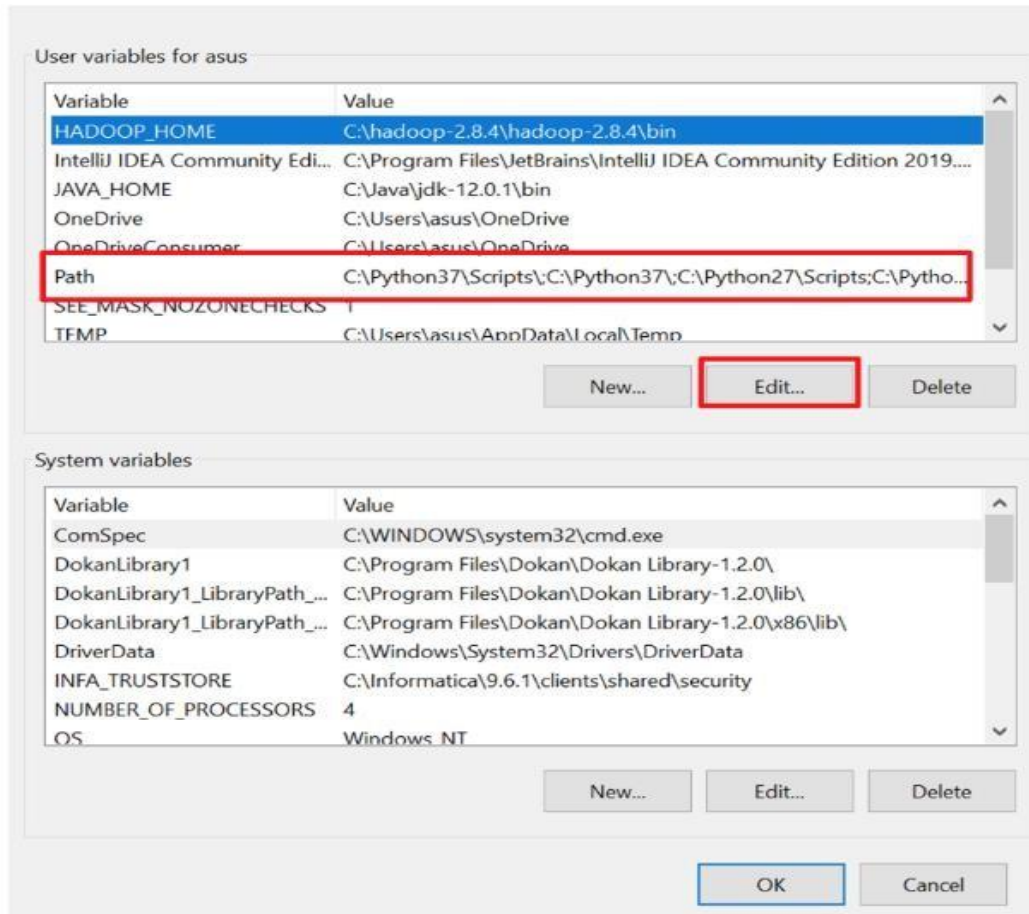
Startup and Recovery
System startup, system failure, and debugging information
[Settings...](#)

Environment Variables...

OK Cancel Apply







5. Configurations

Edit file C:/Hadoop-3.3.0/etc/hadoop/core-site.xml,

paste the xml code in folder and save

```
<configuration>
```

```
  <property>
```

```
    <name>fs.defaultFS</name>
```

```
    <value>hdfs://localhost:9000</value>
```

```
  </property>
```

```
</configuration>
```

=====

Rename “mapred-site.xml.template” to “mapred-site.xml” and edit this file C:\Hadoop-3.3.0/etc/hadoop/mapred-site.xml, paste xml code and save this file.

```
<configuration>
```

```
  <property>
```

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

```
    <value>yarn</value>
```

```
  </property>
```

```
</configuration>
```

=====

Create folder “data” under “C:\Hadoop-3.3.0”

Create folder “datanode” under “C:\Hadoop-3.3.0\data”
Create folder “namenode” under “C:\Hadoop-3.3.0\data”

=====

Edit file C:\Hadoop-3.3.0/etc/hadoop/hdfs-site.xml, paste xml code and save this file.

```
<configuration>
```

```
  <property>
```

```
    <name>dfs.replication</name>
```

```
    <value>1</value>
```

```
  </property>
```

```
  <property>
```

```
    <name>dfs.namenode.name.dir</name>
```

```
    <value>/hadoop-3.3.0/data/namenode</value>
```

```
  </property>
```

```
  <property>
```

```
    <name>dfs.datanode.data.dir</name>
```



```

    <value>/hadoop-3.3.0/data/datanode</value>
  </property>
</configuration>

```

=====
 Edit file C:/Hadoop-3.3.0/etc/hadoop/yarn-site.xml, paste xml code and save this file.

```

<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>

```

=====
 Edit file C:/Hadoop-3.3.0/etc/hadoop/hadoop-env.cmd by closing the command line
 “JAVA_HOME=%JAVA_HOME%” instead of set “JAVA_HOME=C:\Java”

6. Hadoop Configurations

Download https://github.com/brainmentorspvtltd/BigData_RDE/blob/master/Hadoop%20Configuration.zip or (for hadoop 3)

<https://github.com/s911415/apache-hadoop-3.1.0-winutils>

- Copy folder bin and replace existing bin folder in C:\Hadoop-3.3.0\bin
- Format the NameNode
- Open cmd and type command “hdfs namenode -format”



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\hadoop-3.3.0\bin>hdfs namenode -format
```

7. Testing

- Open cmd and change directory to C:\Hadoop-3.3.0\sbin
- type start-all.cmd



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\hadoop-3.3.0\sbin>start-all.cmd
```

(Or you can start like this)

Start name node and data node with this command

- type start-dfs.cmd
- Start yarn through this command
- type start-yarn.cmd

Make sure these apps are running

- Hadoop Namenode
- Hadoop datanode
- YARN ResourceManager

- YARN Node Manager

```

Apache Hadoop Distribution - hadoop - namenode
11/28/2016 12:21:09 PM Apache Hadoop Distribution - hadoop - datanode
11/28/2016 12:21:09 PM Apache Hadoop Distribution - yarn - resourcemanager
11/28/2016 12:21:09 PM Apache Hadoop Distribution - yarn - nodemanager
INFO: the scope "Singleton"
Nov 08, 2016 12:21:09 AM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedComponentProvider with the scope "Singleton"
Nov 08, 2016 12:21:10 AM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManagedComponentProvider with the scope "Singleton"
Nov 08, 2016 12:21:10 AM WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Nov 08, 2016 12:21:10 AM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManagedComponentProvider with the scope "Singleton"
Nov 08, 2016 12:21:10 AM INFO morthbay.log: Started HttpServer2$SelectChannelConnectorWithSafeStartup@0.0.0.0:8042
Nov 08, 2016 12:21:10 AM INFO webapp.WebApps: Web app node started at 8042
Nov 08, 2016 12:21:10 AM INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is : DESKTOP-475FCII:61797
Nov 08, 2016 12:21:10 AM INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8031
Nov 08, 2016 12:21:10 AM INFO nodemanager.NodeStatusUpdaterImpl: Sending out 0 NM container statuses: []
Nov 08, 2016 12:21:10 AM INFO nodemanager.NodeStatusUpdaterImpl: Registering with RM using containers: []
Nov 08, 2016 12:21:10 AM INFO security.NMContainerTokenSecretManager: Rolling master-key for container-tokens, got key with id 528277285
Nov 08, 2016 12:21:10 AM INFO security.NMTokenSecretManagerInNM: Rolling master-key for container-tokens, got key with id 2833623966
Nov 08, 2016 12:21:10 AM INFO nodemanager.NodeStatusUpdaterImpl: Registered with ResourceManager as DESKTOP-475FCII:61797 with total resource of <memory:8192, vCores:8>
Nov 08, 2016 12:21:10 AM INFO nodemanager.NodeStatusUpdaterImpl: Notifying ContainerManager to unblock new container-requests
Nov 08, 2016 12:21:13 AM WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Nov 08, 2016 12:21:16 AM WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Nov 08, 2016 12:21:19 AM WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Nov 08, 2016 12:21:22 AM WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Nov 08, 2016 12:21:25 AM WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Nov 08, 2016 12:21:28 AM WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Nov 08, 2016 12:21:31 AM WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Nov 08, 2016 12:21:34 AM WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7

```

Open:<http://localhost:8088>

The screenshot shows the Hadoop YARN web interface at localhost:8088. The page title is "All Applications". On the left, there is a sidebar with a "Cluster" section containing links for "About", "Nodes", "Node Labels", and "Applications". The "Applications" link is selected. Below the sidebar, there are several tables and sections:

- Cluster Metrics:** A table showing various metrics like Apps Submitted, Apps Pending, Apps Running, etc.
- Cluster Nodes Metrics:** A table showing metrics for Active Nodes, Decommissioning Nodes, Decommissioned Nodes, Lost Nodes, Unhealthy Nodes, Rejected Nodes, and Shutdown Nodes.
- Scheduler Metrics:** A table showing Scheduler Type (Capacity Scheduler), Scheduling Resource Type (MEMORY), and other scheduler-related information.
- Applications Table:** A table with columns for ID, User, Name, Application Type, Queue, Application Priority, StartTime, FinTime, State, PreStatus, Running Containers, Allocated CPU V-Cores, Allocated Memory MB, % of Queue, % of Cluster, Progress %, Tracking UI, and Backlisted Nodes. The table is currently empty, showing "No data available in table".

Open:<http://localhost:9870>

←

→

↺

🏠

localhost:9870/dfshealth.html#tab-overview

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities ▾

Overview 'localhost:9000' (✓active)

Started:	Sun Nov 08 16:53:46 +0530 2020
Version:	3.3.0 [REDACTED] 9af
Compiled:	Tue Jul 07 00:14:00 +0530 2020 by brahma from branch-3.3.0
Cluster ID:	C [REDACTED]
Block Pool ID:	B [REDACTED] 44

Summary

Hadoop installed Successfully.....

EX. No : 2 HADOOP IMPLEMENTATION OF FILE MANAGEMENT TASKS, SUCH AS ADDING FILES AND DIRECTORIES, RETRIEVING FILES AND DELETING FILES

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

Usage:

hadoopfs-mkdir<paths>Example:

hadoopfs-mkdir/user/saurzcode/dir1/user/saurzcode/dir2

2. List the contents of a directory.

Usage:

hadoopfs-

ls<args>Example:

hadoopfs-ls/user/saurzcode

3. Upload and download a file in

HDFS.Upload:hadoop fs -put:

Copy single src file, or multiple src files from local file system to the Hadoop data file systemUsage:

hadoopfs-put<localsrc>...<HDFS_dest_Path> Example:

hadoopfs-put/home/saurzcode/Samplefile.txt/user/saurzcode/dir3/

Download:

hadoopfs-get:

Copies/Downloads files to the local file

systemUsage:

hadoopfs-get<hdfs_src><localdst>Example:

hadoopfs-get/user/saurzcode/dir3/Samplefile.txt/home/

4. See contents of a

fileSame as unix cat

command:Usage:

hadoopfs-cat<path[filename]>

Example:

hadoopfs-cat/user/saurzcode/dir1/abc.txt

5. Copy a file from source to destination

This command allows multiple sources as well in which case the destination must be a directory. Usage:

hadoopfs-cp <source><dest>

Example:

```
hadoop      fs      -cp
/user/saurzcode/dir1/abc.txt
/user/saurzcode/dir2
```

6. Copy a file from/To Local file system to**HDFS copyFromLocal**

Usage:

hadoopfs-copyFromLocal<localsrc>

URI Example:

hadoop fs -copyFromLocal /home/saurzcode/abc.txt /user/
saurzcode/abc.txt Similar to put command, except that the source is restricted to a
local file reference.

copyToLocal

Usage:

hadoopfs-copyToLocal [-ignorecrc] [-crc] URI<localdst>

Similar to get command, except that the destination is restricted to a local file reference.

7. Move file from source to destination.

Note:- Moving files across filesystem is not permitted. Usage:

hadoopfs-mv <src><dest> Example:

hadoopfs-mv /user/saurzcode/dir1/abc.txt /user/saurzcode/dir2

8. Remove a file or directory in HDFS.

Remove files specified as argument. Deletes directory only when it is empty
Usage:

`hadoop fs -rm <arg>` Example:

`hadoop fs -rm /user/saurzcode/dir1/abc.txt`

Recursive version of delete.

Usage:

`hadoop fs -rmr <arg>`

Example: `hadoop fs -`

`rmr /user/saurzcode/`

9. Display last few lines of a file. Similar to tail command in

Unix. Usage:

`hadoop fs -tail <path[filename]>`

Example: `hadoop fs -`

`tail /user/saurzcode/dir1/abc.txt`

10. Display the aggregate length of a file.

Usage:

`hadoop fs -du <path>` Example:

`hadoop fs -du /user/saurzcode/dir1/abc.txt`

EX.No:3 IMPLEMENT OF MATRIX MULTIPLICATION WITH HADOOP MAPREDUCE

AIM:-

To write a Map Reduce Program that implements Matrix Multiplication.

PROCEDURE:

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).

STEP1: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.

STEP2:REDUCER

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

STEP3:MAINDRIVER

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

STEP4: 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.

PROGRAM:

```
import
java.io.IOException;import java.
util.StringTokenizer;
import
org.apache.hadoop.io.IntWritable;import
org.apache.hadoop.io.LongWritable;import
org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
```



```

import
org.apache.hadoop.mapreduce.Reducer;import
org.apache.hadoop.conf.Configuration;import
org.apache.hadoop.mapreduce.Job;
import
org.apache.hadoop.mapreduce.lib.input.TextInputFormat;import
org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;import
org.apache.hadoop.mapreduce.lib.input.FileInputFormat;import
org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;import
org.apache.hadoop.fs.Path;
publicclassMatrixMultiplicationMapperextendsMapper<LongWritable,Text,Text,Text>
{
protectedvoidmap(LongWritablekey,Textvalue,Contextcontext)throwsIOException,InterruptedException{
    // Parse the input line to get row, column, and value of each element in the input
    matricesString[] elements = value.toString().split(",");
    int row =
    Integer.parseInt(elements[0]);int col =
    Integer.parseInt(elements[1]);intval=Integer.parseInt(elements[2]);

    //Emitkey-value pairswherekeyis(row, column) indexoftheresultelement
    // and value is the corresponding element
    valuecontext.write(newText(row+","+ col),new
    Text(val));
}
}

public class MatrixMultiplicationReducer extends Reducer<Text, Text, Text, IntWritable>
{protectedvoidreduce(Textkey,Iterable<Text>values,Contextcontext)throwsIOException,InterruptedException{
    intresult =0;
    for(Text value: values) {
        // Accumulate the partial sum for the result
        elementresult+= Integer.parseInt(value.toString());
    }
}
}

```

```

    }
    // Emit the final result for the result
    elementcontext.write(key,newIntWritable(r
    esult));
}

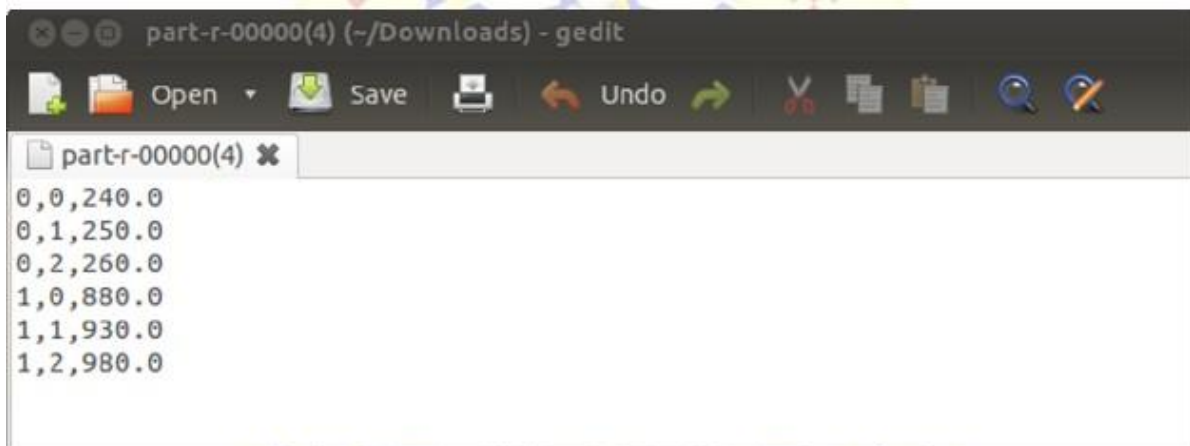
publicclassMatrixMultiplicationDriver{
    public static void main(String[] args) throws Exception
    { Configurationconf=new Configuration();
    Job job = Job.getInstance(conf, "Matrix
    Multiplication");job.setJarByClass(MatrixMultiplication
    Driver.class);job.setMapperClass(MatrixMultiplicationM
    apper.class);job.setReducerClass(MatrixMultiplicationRe
    ducer.class);job.setOutputKeyClass(Text.class);job.setOu
    tputValueClass(Text.class);FileInputFormat.addInputPat
    h(job, new
    Path(args[0]));FileOutputFormat.setOutputPath(job, new
    Path(args[1]));System.exit(job.waitForCompletion(true)?
    0: 1);
    }
}
}

```

Run the program

Hadoop jar matrix multiplication.jar Matrix Multiplication Driver input_path output_path

```
lendi@ubuntu:~/Desktop$ hadoop jar MatrixMultiplication.jar /matrix_data/ /matrix_output_new
```



RESULT:

Thus the Map Reduce Program that implements Matrix Multiplication was executed and verified successfully.

EX.NO: 4 RUN A BASIC WORD COUNT MAP REDUCE PROGRAM TO UNDERSTAND MAPREDUCE PARADIGM.

AIM:-

To write a Basic Word Count program to understand Map Reduce Paradigm.

PROCEDURE:

The entire Map Reduce program can be fundamentally divided into three parts:

- MapperPhaseCode
- ReducerPhaseCode
- DriverCode

STEP1:MAPPERCODE:

We have created a class Map that extends the class Mapper which is already defined in the MapReduce Framework.

- We define the data types of input and output key/value pair after the class declaration using angle brackets.
- Both the input and output of the Mapper is a key/value pair.

Input:

- The key is nothing but the offset of each line in the text file: LongWritable
- The value is each individual line : Text

Output:

- The key is the tokenized words: Text
- We have the hardcoded value in our case which is 1: IntWritable
- **Example**—Dear 1, Bear 1, etc.

We have written a java code where we have tokenized each word and assigned them a hardcoded value equal to 1.

STEP2 :REDUCER CODE:

- We have created a class Reduce which extends class Reducer like that of Mapper.
- We define the data types of input and output key/value pair after the class declaration using angle brackets as done for Mapper.
- Both the input and the output of the Reducer is a key value pair.

Input:

- The key nothing but those unique words which have been generated after the sorting and shuffling phase: Text
- The value is a list of integers corresponding to each key: IntWritable
- Example—Bear, [1,1], etc.

Output:

- The key is all the unique words present in the input text file: Text
- The value is the number of occurrences of each of the unique words: IntWritable
- Example—Bear,2;Car,3, etc.
- We have aggregated the values present in each of the list corresponding to each key and produced the final answer.
- In general, a single reducer is created for each of the unique words, but, you can specify the number of reducer in mapred-site.xml.

STEP3:DRIVER CODE:

- In the driver class, we set the configuration of our MapReduce job to run in Hadoop.
 - We specify the name of the job, the data type of input/output of the mapper and reducer.
 - We also specify the names of the mapper and reducer classes.
 - The path of the input and output folder is also specified.
 - The method setInputFormatClass () is used for specifying that how a Mapper will read the input data or what will be the unit of work. Here, we have chosen TextInputFormat so that single line is read by the mapper at a time from the input text file.
- The main
- () method is the entry point for the driver. In this method, we instantiate a new Configuration object for the job.

PROGRAM:

```
import
java.io.IOException;import java.
util.StringTokenizer;
import
org.apache.hadoop.io.IntWritable;import
org.apache.hadoop.io.LongWritable;import
org.apache.hadoop.io.Text;
```

```

import
org.apache.hadoop.mapreduce.Mapper;import
org.apache.hadoop.mapreduce.Reducer;

import
org.apache.hadoop.conf.Configuration;import
org.apache.hadoop.mapreduce.Job;

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

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

public static class Reduce extends Reducer<Text,IntWritable,Text,IntWritable>
{public void reduce(Text key, Iterable<IntWritable> values,Context
context)throwsIOException,InterruptedException {
intsum=0;

for(IntWritablax:values)
{
sum+=x.get();
}
}
}
}

```

```

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

public static void main(String[] args) throws Exception
{
    Configuration conf = new Configuration()

    Job job = new Job(conf, "My Word Count
    Program");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(Map.class);
    job.setReducerClass(Reducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

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

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

    //deleting the output path automatically from hdfs so that we don't have
    to delete it explicitly
    outputPath.getFileSystem(conf).delete(outputPath);

    //exiting the job only if the flag value becomes
    false
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}

```

Run the MapReduce code:

The command for running a MapReduce code is:

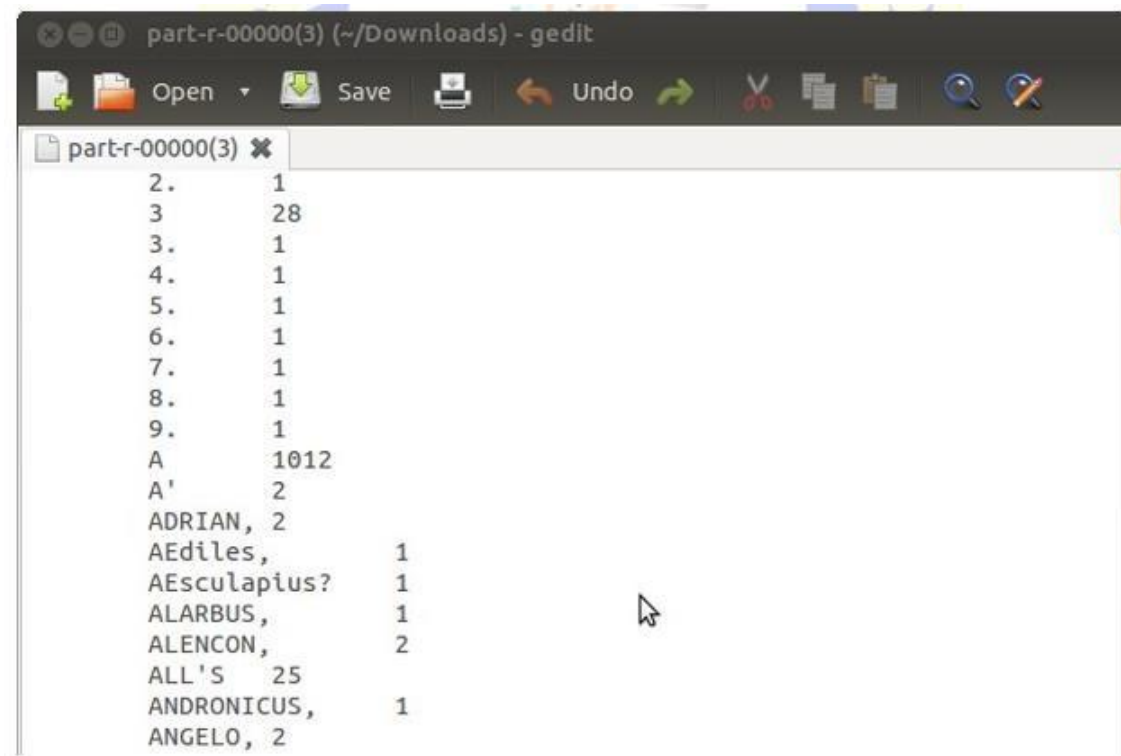
`hadoop jar hadoop-mapreduce-example.jar WordCount/sample/input/sample/output`

OUTPUT:

```

lendl@ubuntu: ~/Desktop
16/08/17 01:17:45 INFO impl.YarnClientImpl: Submitted application application_1471410736896_0001
16/08/17 01:17:45 INFO mapreduce.Job: The url to track the job: http://ubuntu.ubuntu-domain:8088/proxy/application_1471410736896_0001/
16/08/17 01:17:45 INFO mapreduce.Job: Running job: job_1471410736896_0001
16/08/17 01:17:52 INFO mapreduce.Job: Job job_1471410736896_0001 running in uber mode : false
16/08/17 01:17:52 INFO mapreduce.Job:  map 0% reduce 0%
16/08/17 01:17:59 INFO mapreduce.Job:  map 100% reduce 0%
16/08/17 01:18:06 INFO mapreduce.Job:  map 100% reduce 100%
16/08/17 01:18:06 INFO mapreduce.Job: Job job_1471410736896_0001 completed successfully
16/08/17 01:18:06 INFO mapreduce.Job: Counters: 49
      File System Counters
        FILE: Number of bytes read=3772644
        FILE: Number of bytes written=7775215
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=1714718
        HDFS: Number of bytes written=510970
        HDFS: Number of read operations=6
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2

```

The screenshot shows a gedit text editor window titled "part-r-00000(3) (~/Downloads) - gedit". The window contains a list of words and their corresponding counts, formatted as follows:

2.	1
3	28
3.	1
4.	1
5.	1
6.	1
7.	1
8.	1
9.	1
A	1012
A'	2
ADRIAN,	2
AEdiles,	1
AEsculapius?	1
ALARBUS,	1
ALENCON,	2
ALL 'S	25
ANDRONICUS,	1
ANGELO,	2

RESULT:

Thus the Map Reduce Program that implements word count was executed and verified successfully.

EX.NO : 5 INSTALLATION OF HIVE ALONG WITH PRACTICE EXAMPLES.

PREREQUISITES:

- JavaDevelopmentKit(JDK)installedandtheJAVA_HOMEenvironmentvariableset.
- HadoopinstalledandconfiguredonyourWindowssystem.

STEP-BY-STEPINSTALLATION:

1. DownloadHIVE:

Visit the Apache Hive website and download the latest stable version of Hive.OfficialApacheHivewebsite:<https://hive.apache.org/>

2. Extractthe DownloadedHiveArchivetoaDirectoryonYourWindowsMachine, e.g.,C:\hive.

3. ConfigureHive:

- Open the Hive configuration file (hive-site.xml) located in the conf folder of theextractedHive directory.
- Setthe necessary configurations,suchasHiveMetastore connectionsettingsandHadoop configurations. Make sure to adjust paths accordingly for Windows. Here's anexampleof someconfigurations:

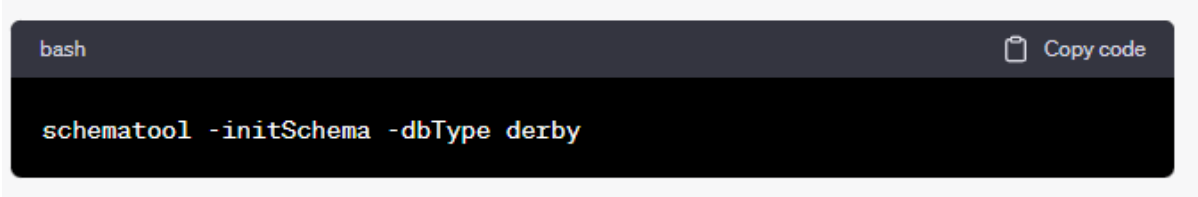
```
<configuration>
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:derby::databaseName=/path/to/metastore_db;create=true</value>
  <description>JDBCconnectstringforaJDBC metastore.</description>
</property>
<!--OtherHiveconfigurations-->
</configuration>
```

4. EnvironmentVariablesSetup:

- AddtheHivebinarydirectory(C:\hive\bininthisexample)toyourPATHenvironmentvariable.
- SettheHIVE_HOMEenvironmentvariabletopointtotheHiveinstallationdirectory(C:\hivein this example).

5. Start the Hive Metastore service:

To start the Hive Metastore service, you can use the schematool script:



```
bash
schematool -initSchema -dbType derby
```

6. Start Hive:

- Open a command prompt or terminal and navigate to the Hive installation directory.
- Execute the hive command to start the Hive shell.

EXAMPLES:

1. Create a Database:

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

```
CREATE DATABASE database_name;
```

Example:

```
CREATE DATABASE mydatabase;
```

2. Use a Database:

To use a specific database in HIVE, use the following syntax: `USE database_name;`

Example:

```
USE mydatabase;
```

3. Show Databases:

To display a list of available databases in HIVE, use the following syntax: `SHOW DATABASES;`

4. Create a Table:

To create a table in HIVE, use the following

```
syntax: CREATE TABLE table_name (  
    column1  
    datatype, column2  
    datatype,  
    ... );
```

Example:

```
CREATE TABLE mytable
(idINT,
name
STRING,age
INT
);
```

5. ShowTables:

To display a list of tables in the current database, use the following syntax: **SHOW TABLES;**

6. Describe a Table:

To view the schema and details of a specific table, use the following syntax: **DESCRIBE table_name;**

Example:

```
DESCRIBE mytable;
```

7. 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 mytable(id,name,age)VALUES (1,'JohnDoe',25);
```

8. Select Data from a Table:

```
SELECT column1,column2,...FROM table_name WHERE condition;
```

Example:

```
SELECT* FROM mytable WHERE age>20;
```

RESULT:

Thus the Installation of HIVE was done successfully.

EX.NO : 6 INSTALLATION OF HBASE INSTALLING ALONG WITH PRACTICE EXAMPLES

AIM:

To install HBASE using Virtual Machine and perform some operations in HBASE.

PROCEDURE:

Step1:Install a VirtualMachine

- DownloadandinstallavirtualmachinesoftwaresuchasVirtualBox(<https://www.virtualbox.org/>)or VMware(<https://www.vmware.com/>).
- CreateanewvirtualmachineandinstallaUnix-basedoperatingsystemlikeUbuntuorCentOS. YoucandownloadtheISOimageofyourdesiredLinuxdistributionfromtheirofficialwebsites.

Step2:SetuptheVirtualMachine

- LaunchthevirtualmachineandinstalltheUnix-basedoperatingsystemfollowingtheinstallationwizard.
- Makesurethevirtualmachinehasnetworkconnectivitytodownloadsoftwarepackages.

Step3:InstallJava

- Opentheterminalorcommandlinein thevirtualmachine.
- Updatethepackagelist
sudoaptupdate
- InstallOpenJDK(JavaDevelopmentKit):
sudoaptinstalldefault-jdk
- Verifythe Javainstallation:
java-version

Step4:DownloadandInstallHBase

- Inthevirtualmachine,navigatetothedirectorywhereyouwanttoinstallHBase.
- Download the HBase binarydistributionfromtheApacheHBasewebsite(<https://hbase.apache.org/>). Look for thelateststable version.
- Extractthedownloadedarchive
tar-xvf<hbase_archive_name>.tar.gz
- Replace<hbase_archive_name>withtheactual nameofthe HBasearchivefile.

- Move the extracted HBase directory to a desired location:
sudo mv <hbase_extracted_directory> /opt/hbase
- Replace <hbase_extracted_directory> with the actual name of the extracted HBase directory.

Step 5: Configure HBase

- Open the HBase configuration file for editing:
sudo nano /opt/hbase/conf/hbase-site.xml
- Add the following properties to the configuration file:

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>file:///var/lib/hbase</value>
  </property>
  <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/var/lib/zookeeper</value>
  </property>
</configuration>
```

- Save the file and exit the text editor.

Step 6: Start HBase

- Start the HBase server:
sudo /opt/hbase/bin/start-

hbase.sh HBASE PRACTICE EXAMPLE

S:

Step 1: Start HBase

- Make sure HBase is installed and running on your Windows system.

Step 2: Open HBase Shell

- Open a command prompt or terminal window and navigate to the directory where the HBase installation is located. Run the following command to start the HBase shell:

>> hbase shell

Step3:CreateaTable

- Inthe 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'
```

Step4:InsertData

- To 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'
```

Step5:GetData

- 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 specified row.

Step6:ScanData

- To scan and retrieve multiple rows or the entire table, use the scan command.
- For instance, to scan all rows in the table:

```
>>scan'my_table'
```

- This will display all rows and their corresponding column family values.

Step7:DeleteData

- To delete a specific row or a particular cell value, you can use the delete command.
- Here's an example of deleting a specific row:

```
>>delete'my_table','row1'
```

Step8:Disableand DropTable

- 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'
```

RESULT:

Thus the installation of HBase using Virtual Machine was done successfully.

EX.NO : 7 INSTALLATION OF THRIFT

AIM:

To install Apache thrift on Windows OS.

PROCEDURE:

Step1:DownloadApacheThrift:

- VisittheApacheThriftwebsite:<https://thrift.apache.org/>
- Gotothe "Downloads"section andfindthelatestversion ofThrift.
- DownloadtheWindowsbinarydistribution (ZIPfile) for thedesired version.

Step2:Extract theZIPfile:

- LocatethedownloadedZIPfile andextractitscontentsto adirectoryofyourchoice.
- Thisdirectorywillbereferred toas <THRIFT_DIR>inthe followingsteps.

Step3:Setupenvironmentvariables:

- OpentheStartmenuandsearchfor"EnvironmentVariables"andselect"Editthesystemenvironment variables."
- Clickthe"EnvironmentVariables"buttonatthebottomrightofthe"SystemProperties"window.
- Underthe"Systemvariables"section,findthe "Path"variableandclick"Edit."
- Addthefollowingentries tothe"Variablevalue"field(replace<THRIFT_DIR>withthe actual directorypath):

<THRIFT_DIR>\bin
<THRIFT_DIR>\lib
- Click"OK"tosavethechanges.

Step4:Verifytheinstallation:

- Openanew Command Prompt window.
- Runthe followingcommand to verifythat Thrift is installed andaccessible:

thrift-version
- Ifeverythingissetupcorrectly,youshouldseetheversionnumberofThriftprintedonthe screen.

RESULT:

Thus the installation of Thrift on windows OS was done successfully.

EX.NO :8 PRACTICE IMPORTING AND EXPORTING DATA FROM VARIOUS DATABASES.

AIM:

To import and export data from various Databases using SGOOP.

PROCEDURE:

Step1:InstallSGOOP.

- 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

Step2:Importing data from a database:

- To import data from a database into Hadoop, use the following Sqoop command:

```
Sqoop import –  
connect jdbc:<DB_TYPE>://<DB_HOST>:<DB_PORT>/<DB_N  
AME>\  
    --username<DB_USERNAME> \  
    --password<DB_PASSWORD>\  
    --table<TABLE_NAME>\  
    --target-dir<HDFS_TARGET_DIR> \  

```

- Replace the placeholders
 - (<DB_TYPE>,<DB_HOST>,<DB_PORT>,<DB_NAME>,<DB_USERNAME>,<DB_PASSWORD>,<TABLE_NAME>,<HDFS_TARGET_DIR>,<NUMBER_OF_MAP_TASKS>) with the appropriate values for your database and Hadoop environment.

Step3:Exporting data to a database:

To export data from Hadoop to a database, use the following Sqoop command:

```
sqoop export –  
connect jdbc:<DB_TYPE>://<DB_HOST>:<DB_PORT>/<DB_N  
AME>\
```

```
--password <DB_PASSWORD>\
--table<TABLE_NAME>\
--export-dir<HDFS_EXPORT_DIR> \
--input-fields-terminated-by'<DELIMITER>'
```

- Replace the placeholders
 - (<DB_TYPE>,<DB_HOST>,<DB_PORT>,<DB_NAME>,<DB_USERNAME>,<DB_PASSWORD>,<TABLE_NAME>,<HDFS_EXPORT_DIR>, and <DELIMITER>) with the appropriate values for your database and Hadoop environment.

RESULT:

Thus the implementation export data from various Databases using SQOOP was done successfully.