



ARUNAI ENGINEERING COLLEGE

(An Autonomous Institution)

Velu Nagar, Thiruvannamalai-606603

www.arunai.org



**DEPARTMENT OF ARTIFICIAL INTELLIGENCE
& DATA SCIENCE**

BACHELOR OF TECHNOLOGY

2024-2025

FIFTH SEMESTER

**CCS334 –BIG DATA ANALYTICS
LABORATORY**

ARUNAI ENGINEERING COLLEGE

TIRUVANNAMALAI – 606 603



DEPARTMENT OF ARTIFICIAL INTELLIGENCE &DATA SCIENCE CERTIFICATE

Certified that this is a bonafide record of work done by

Name : _____

University Reg.No : _____

Semester : _____

Branch : _____

Year : _____

Staff-in-Charge

Head of the Department

Submitted for the _____

Practical Examination held on _____

Internal Examiner

External Examiner

S.NO	DATE	List Of Experiments	Pg.No	Signature
1		INSTALLATION OF HADOOP		
2		HADOOP FILE MANAGEMENT TASKS		
3		MATRIX MULTIPLICATION		
4		WORD COUNT MAP REDUCE		
5		HIVE		
6.1		HBASE		
6.2		THRIFT		
7.1		CASSANDRA		
7.2		MongoDB		

EX.NO: 1

INSTALLATION OF HADOOP

DATE:

AIM:

To Download and install Hadoop; Understand different Hadoop modes. Startup scripts, Configuration files.

THEORY:

Hadoop is a Java-based programming framework that supports the processing and storage of extremely large datasets on a cluster of inexpensive machines. It was the first major open-source project in the big data playing field and is sponsored by the Apache Software Foundation.

Hadoop-2.8.0 is comprised of four main layers:

- **Hadoop Common** is the collection of utilities and libraries that support other Hadoop modules.
- **HDFS**, which stands for Hadoop Distributed File System, is responsible for persisting data to disk.
- **YARN**, short for Yet Another Resource Negotiator, is the "operating system" for HDFS.
- **MapReduce** is the original processing model for Hadoop clusters. It distributes work within the cluster or map, then organizes and reduces the results from the nodes into a response to a query. Many other processing models are available for the 2.x version of Hadoop.

Hadoop clusters are relatively complex to set up, so the project includes a stand-alone mode which is suitable for learning about Hadoop, performing simple operations, and debugging.

PREPARE:

These softwares should be prepared to install Hadoop 2.8.0 on window 10 64bit

1. Download Hadoop 2.8.0
2. Java JDK 1.8.0.zip

PROCEDURE:

Procedure to Run Hadoop

1. Install Apache Hadoop 2.8.0 in Microsoft Windows OS

If Apache Hadoop 2.8.0 is not already installed then follow the post Build, Install, Configure and Run Apache Hadoop 2.8.0 in Microsoft Windows OS.

2. Start HDFS (Namenode and Datanode) and YARN (Resource Manager and Node Manager)

Run following commands.

Command Prompt

C:\Users\> hdfs namenode –format

C:\hadoop\sbin>start-dfs

C:\hadoop\sbin>start-yarn

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

C:\hadoop\sbin>jps (used to check how many nodes are running in background of Hadoop)

Namenode, Datanode, Resource Manager and Node Manager will be started in few minutes and ready to execute Hadoop **MapReduce** job in the Single Node (pseudo-distributed mode) cluster.

PREREQUISITES:

Step1: Installing Java 8 version.

Openjdk version "1.8.0_91"

OpenJDK Runtime Environment (build 1.8.0_91-8u91-b14-3ubuntu1~16.04.1-b14)

OpenJDK 64-Bit Server VM (build 25.91-b14, mixed mode)

This output verifies that OpenJDK has been successfully installed.

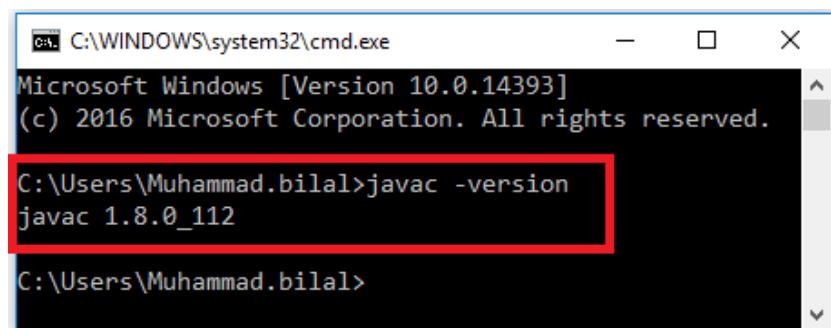
Note: To set the path for environment variables. i.e. JAVA_HOME

Step2: Installing Hadoop

With Java in place, we'll visit the Apache Hadoop Releases page to find the mostrecent stable release. Follow the binary for the current release:

Set up

1. Check either Java 1.8.0 is already installed on your system or not, use “**Javac -version**” to check



The screenshot shows a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The window displays the following text:
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Muhammad.bilal>javac -version
javac 1.8.0_112

C:\Users\Muhammad.bilal>

The command 'javac -version' is highlighted with a red rectangle.

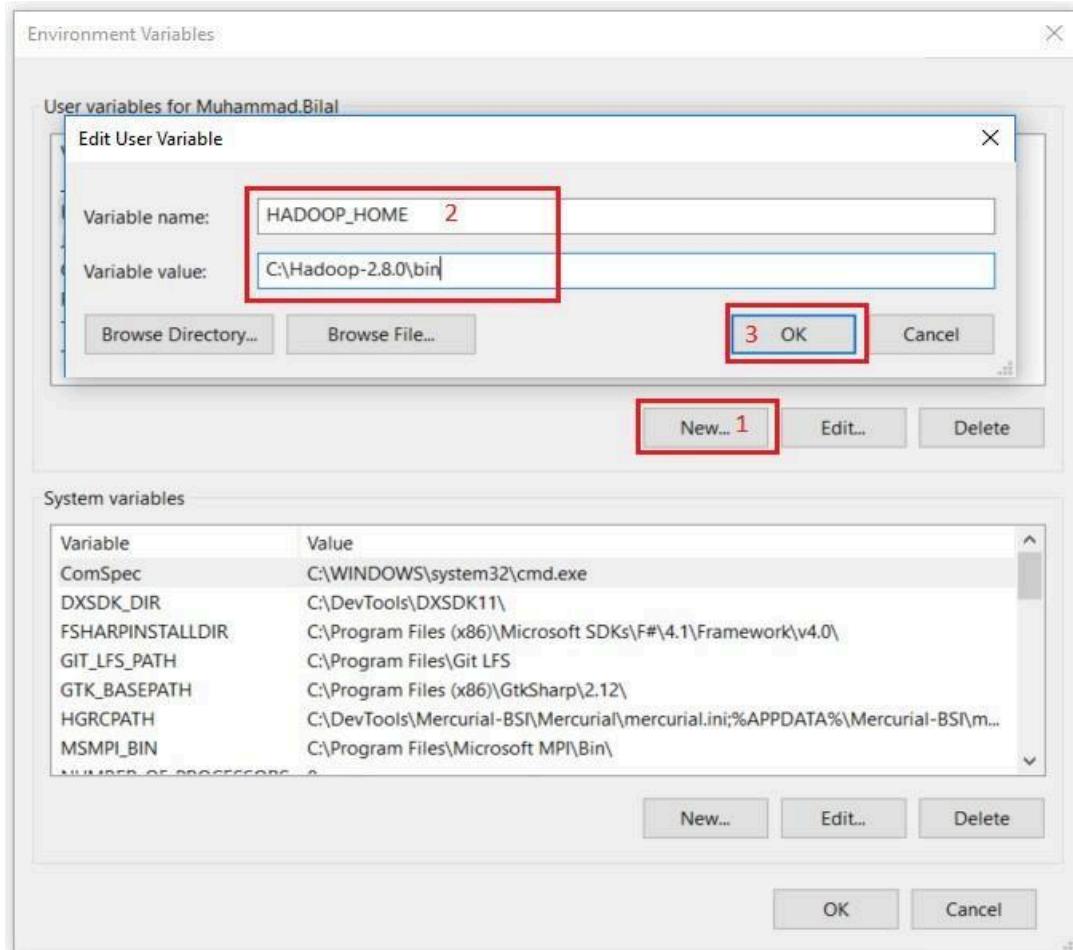
2. If Java is not installed on your system then first install java under **C:\JAVA**

Name	Date modified	Type
ATP	5/22/2017 3:19 PM	File folder
AzureTemp	7/18/2017 5:57 PM	File folder
cygwin64	7/18/2017 10:58 AM	File folder
DevTools	6/19/2017 12:39 PM	File folder
Hadoop-2.8.0	7/18/2017 12:43 PM	File folder
inetpub	5/8/2017 10:49 PM	File folder
Intel	4/25/2017 9:12 AM	File folder
ITSD	4/25/2017 9:20 AM	File folder
Java	7/18/2017 12:29 PM	File folder
PerfLogs	7/16/2016 4:47 PM	File folder
policies	5/18/2017 2:56 PM	File folder
Program Files	7/10/2017 1:06 PM	File folder
Program Files (x86)	7/12/2017 12:35 PM	File folder

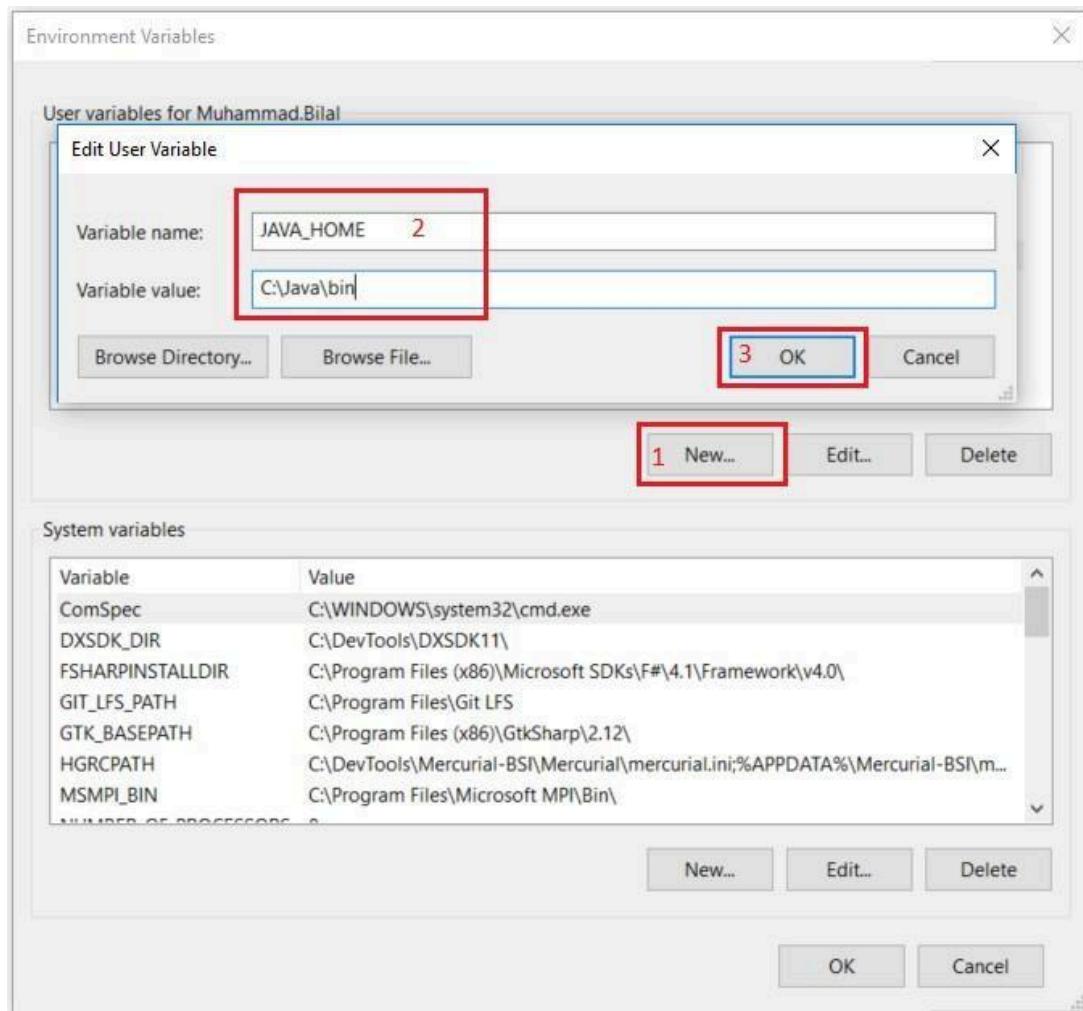
3. Extract file Hadoop 2.8.0.tar.gz or Hadoop-2.8.0.zip and place under "**C:\Hadoop-2.8.0**"

Name	Date modified	Type
ATP	5/22/2017 3:19 PM	File folder
AzureTemp	7/18/2017 5:57 PM	File folder
cygwin64	7/18/2017 10:58 AM	File folder
DevTools	6/19/2017 12:39 PM	File folder
Hadoop-2.8.0	7/18/2017 12:43 PM	File folder
inetpub	5/8/2017 10:49 PM	File folder
Intel	4/25/2017 9:12 AM	File folder
ITSD	4/25/2017 9:20 AM	File folder
Java	7/18/2017 12:29 PM	File folder
PerfLogs	7/16/2016 4:47 PM	File folder
policies	5/18/2017 2:56 PM	File folder
Program Files	7/10/2017 1:06 PM	File folder
Program Files (x86)	7/12/2017 12:35 PM	File folder

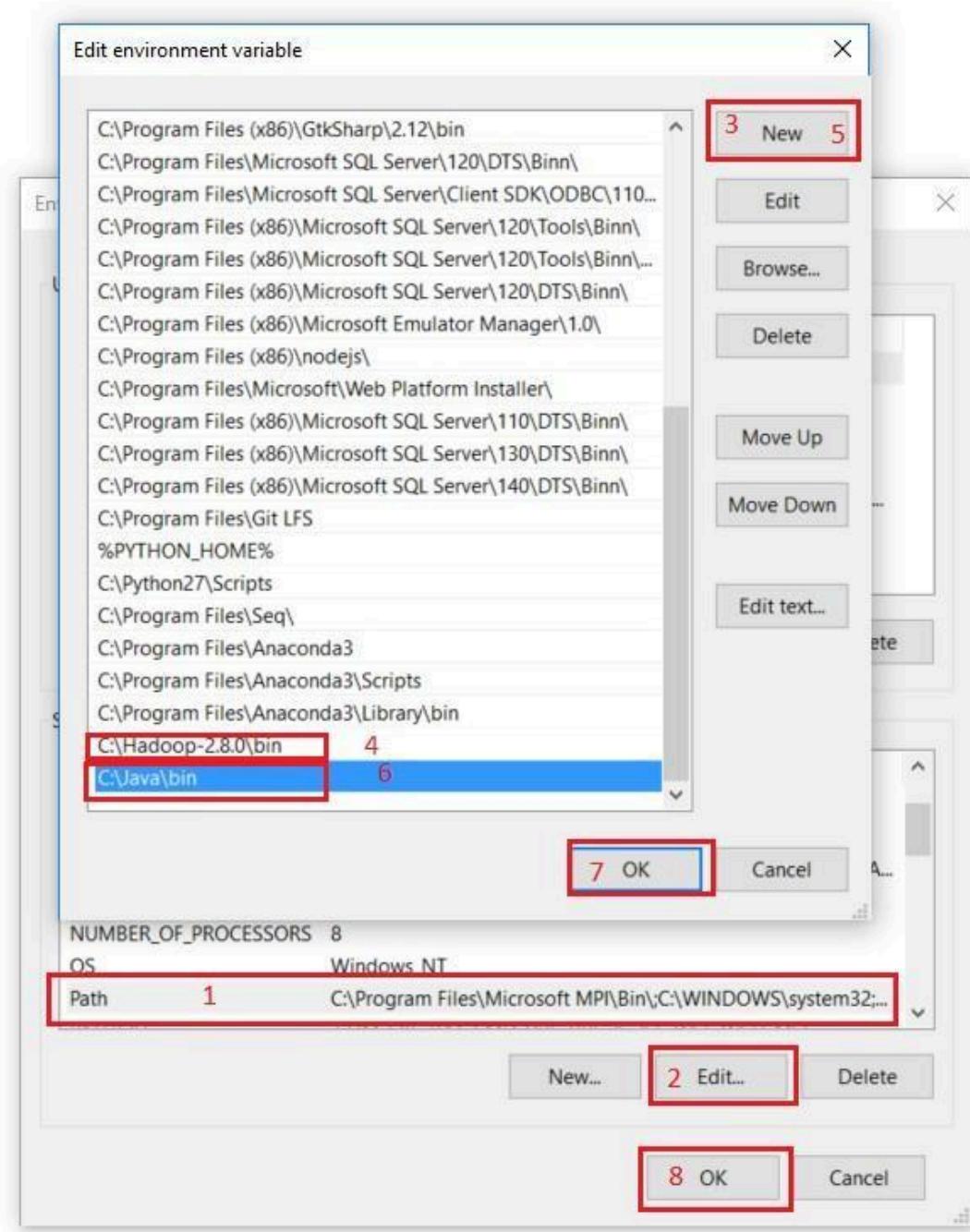
4. Set the path HADOOP_HOME Environment variable on windows 10(see Step 1,2,3 and 4 below)



5. Set the path JAVA_HOME Environment variable on windows 10(see Step 1,2,3 and 4 below)



6. Next, we set the Hadoop bin directory path and JAVA bin path



CONFIGURATION

1. Edit file **C:/Hadoop-2.8.0/etc/hadoop/core-site.xml**, paste below xml paragraph and save this file.

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

2. Rename “mapred-site.xml.template” to “mapred-site.xml” and edit this file

C:/Hadoop-2.8.0/etc/hadoop/mapred-site.xml, paste below xml paragraph and save this file.

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

3. Create folder “**data**” under “**C:\Hadoop-2.8.0**”

- Create folder “**datanode**” under “**C:\Hadoop-2.8.0\data**”
- Create folder “**namenode**” under “**C:\Hadoop-2.8.0\data**”

<input type="checkbox"/> Name	Date modified	Type	Size
bin	7/20/2017 2:14 PM	File folder	
<input checked="" type="checkbox"/> data	7/20/2017 2:47 PM	File folder	
etc	7/20/2017 2:14 PM	File folder	
include	7/20/2017 2:14 PM	File folder	
lib	7/20/2017 2:14 PM	File folder	
libexec	7/20/2017 2:14 PM	File folder	
sbin	7/20/2017 2:14 PM	File folder	
share	7/20/2017 2:20 PM	File folder	
LICENSE.txt	3/17/2017 10:31 AM	TXT File	97 KB
NOTICE.txt	3/17/2017 10:31 AM	TXT File	16 KB
README.txt	3/17/2017 10:31 AM	TXT File	2 KB

4. Edit file **C:/Hadoop-2.8.0/etc/hadoop/hdfs-site.xml**, paste below xml paragraph and save this file.

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

```

<value>1</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>/hadoop-2.8.0/data/namenode</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>/hadoop-2.8.0/data/datanode</value>
</property>
</configuration>
```

5. Edit file **C:/Hadoop-2.8.0/etc/hadoop/yarn-site.xml**, paste below xml paragraph 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>
```

6. Edit file **C:/Hadoop-2.8.0/etc/hadoop/hadoop-env.cmd** by closing the command

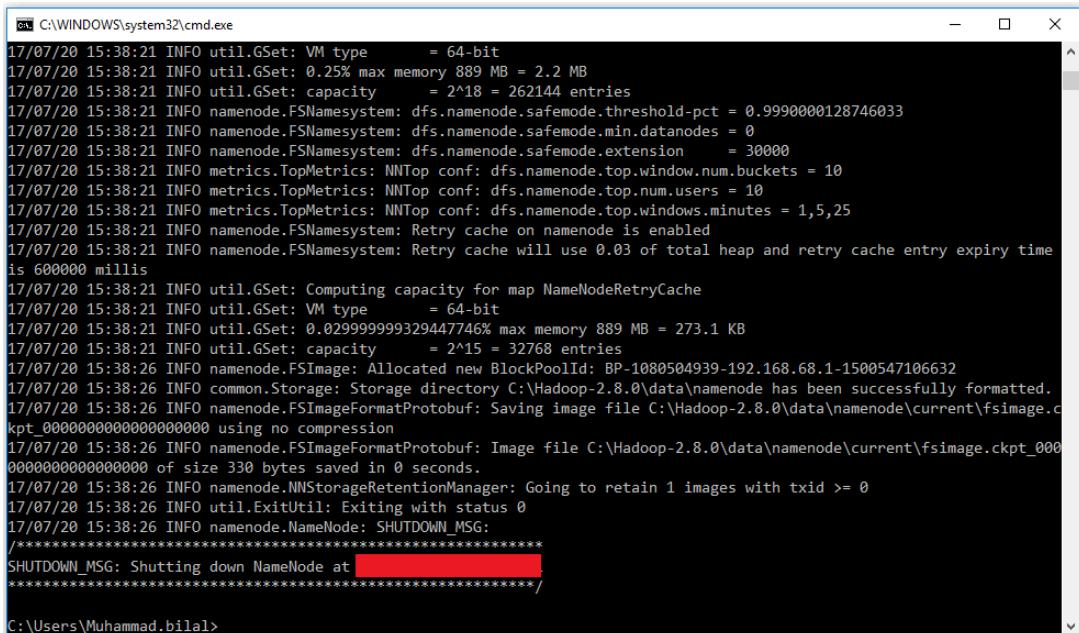
line “**JAVA_HOME=%JAVA_HOME%**” instead of set “**JAVA_HOME=C:\Java**” (On C:\java this is path to file jdk.18.0)

```

@rem The java implementation to use. Required.
@rem set JAVA_HOME=%JAVA_HOME%
set JAVA_HOME=C:\java|
```

Hadoop Configuration

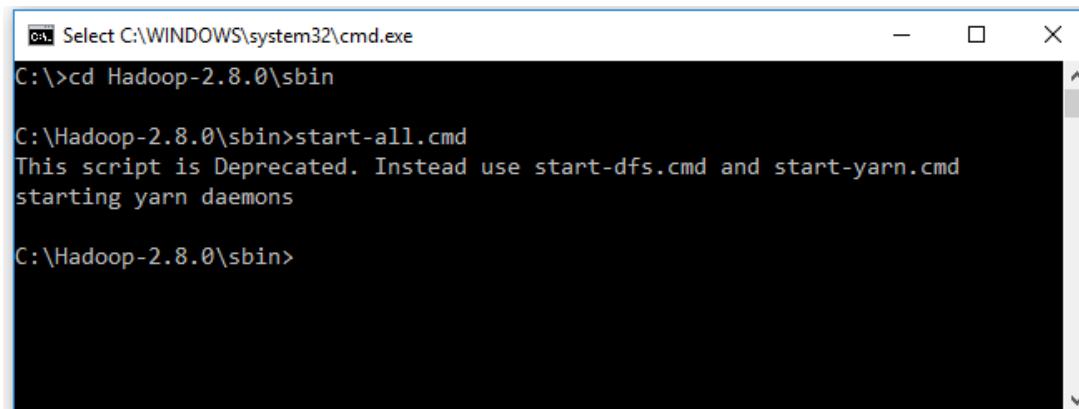
1. Dowload file [Hadoop Configuration.zip](#)
2. Delete file bin on C:\Hadoop-2.8.0\bin, replaced by file bin on file just download (from Hadoop Configuration.zip).
3. Open cmd and typing command “**hdfs namenode –format**” . You will see



```
C:\WINDOWS\system32\cmd.exe
17/07/20 15:38:21 INFO util.GSet: VM type      = 64-bit
17/07/20 15:38:21 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
17/07/20 15:38:21 INFO util.GSet: capacity      = 2^18 = 262144 entries
17/07/20 15:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
17/07/20 15:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
17/07/20 15:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension = 30000
17/07/20 15:38:21 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
17/07/20 15:38:21 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
17/07/20 15:38:21 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
17/07/20 15:38:21 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
17/07/20 15:38:21 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time
is 600000 millis
17/07/20 15:38:21 INFO util.GSet: Computing capacity for map NameNodeRetryCache
17/07/20 15:38:21 INFO util.GSet: VM type      = 64-bit
17/07/20 15:38:21 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
17/07/20 15:38:21 INFO util.GSet: capacity      = 2^15 = 32768 entries
17/07/20 15:38:26 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1080504939-192.168.68.1-1500547106632
17/07/20 15:38:26 INFO common.Storage: Storage directory C:\Hadoop-2.8.0\data\namenode has been successfully formatted.
17/07/20 15:38:26 INFO namenode.FSImageFormatProtobuf: Saving image file C:\Hadoop-2.8.0\data\namenode\current\fimage.c
kpt_0000000000000000 using no compression
17/07/20 15:38:26 INFO namenode.FSImageFormatProtobuf: Image file C:\Hadoop-2.8.0\data\namenode\current\fimage.ckpt_000
0000000000000 of size 330 bytes saved in 0 seconds.
17/07/20 15:38:26 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
17/07/20 15:38:26 INFO util.ExitUtil: Exiting with status 0
17/07/20 15:38:26 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN_MSG: Shutting down NameNode at [REDACTED]
*****/
```

Testing

1. Open cmd and change directory to “C:\Hadoop-2.8.0\sbin” and type “**start-all.cmd**” to start Hadoop



```
Select C:\WINDOWS\system32\cmd.exe
C:>>cd Hadoop-2.8.0\sbin
C:\Hadoop-2.8.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Hadoop-2.8.0\sbin>
```

2. Make sure these apps are running

- o Hadoop Namenode
- o Hadoop datanode
- o YARN Resourc Manager
- o YARN Node Manager

```

17/e17/ Jun 17/07/20 15:50:09 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e2 oIN17/07/20 15:50:12 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e17/ Jun 17/07/20 15:50:15 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e168IN17/07/20 15:50:18 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
name17/ Jun 17/07/20 15:50:21 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
stor106IN17/07/20 15:50:24 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
stor17/wi17/07/20 15:50:27 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e17/ Jun 17/07/20 15:50:27 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e17/ Jun 17/07/20 15:50:30 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
lisadeoIN17/07/20 15:50:33 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e17/g117/07/20 15:50:36 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
d=8354dJu17/07/20 15:50:39 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
09fc17/IN17/07/20 15:50:42 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e17/e17/07/20 15:50:46 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e17/17/07/20 15:50:49 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
0.0.0f-1717/07/20 15:50:52 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e17/17/17/07/20 15:50:55 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
for54dty17/07/20 15:50:58 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e17/17/17/07/20 15:51:01 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
-ea483817/07/20 15:51:04 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/e17/is17/07/20 15:51:07 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
f-7sec 17/07/20 15:51:10 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
075_17/17/07/20 15:51:13 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
509st(s17/07/20 15:51:16 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
r Rtt17/07/20 15:51:19 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/17/07/20 15:51:22 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/20 15:51:25 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Co17/07/20 15:51:29 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/20 15:51:32 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/20 15:51:35 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7

```

3. OUTPUT:

Open:

<http://localhost:8088>

The screenshot shows the Apache Hadoop Resource Manager (YARN) UI at <http://localhost:8088/cluster>. The page title is "All Applications".

Cluster Metrics:

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	0	0	0	0 B	8 GB	0 B	0	8	0

Cluster Nodes Metrics:

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
1	0	0	0	0	0	0

Scheduler Metrics:

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Applications:

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU Vcores	Allocated Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No data available in table.																	

Showing 0 to 0 of 0 entries

First Previous Next Last

4. OUTPUT:

Open:

http://localhost:50070

① localhost:50070/dfshealth.html#tab-overview						
Hadoop	Overview	Datanodes	Datanode Volume Failures	Snapshot	Startup Progress	Utilities ▾

Overview 'localhost:9000' (active)

Started:	Thu Jul 20 15:44:11 +0500 2017
Version:	2.8.0, r91f2b7a13d1e97b [REDACTED] 7cc29ac0009
Compiled:	Fri Mar 17 09:12:00 +0500 2017 by jdu from branch-2.8.0
Cluster ID:	CID-098b09fc-fd [REDACTED] df7b674
Block Pool ID:	BP-10805049 [REDACTED] 47106632

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 36.53 MB of 311 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 40.68 MB of 41.53 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	475.24 GB
DFS Used:	321 B (0%)
Non DFS Used:	261.08 GB

RESULT:

Thus, a procedure to installation of Hadoop cluster was successfully executed.

EX.NO: 2

HADOOP FILE MANAGEMENT TASKS

DATE:

AIM:

Implement the following file management tasks in Hadoop:

- Adding files and directories
- Retrieving files
- Deleting files

DESCRIPTION:

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

- Adding files and directories to HDFS
- Retrieving files from HDFS to local filesystem
- Deleting files from HDFS

ALGORITHM:

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

Step-1: Adding Files and Directories to HDFS

Before you can run Hadoop programs on data stored in HDFS, you'll need to put the data into HDFS first. Let's create a directory in and put a file in it. HDFS has a default working directory of /user/\$USER, where \$USER is your login user name. This directory isn't automatically created for you, though, so let's create it with the mkdir command.

Note: input_file.txt is created in sbin with some contents

```
C:\hadoop-2.8.0\sbin>hadoop fs -mkdir /input_dir
```

```
C:\hadoop-2.8.0\sbin>hadoop fs -put input_file.txt /input_dir/input_file.txt
```

Step 2: List the contents of a directory.:

```
C:\hadoop-2.8.0\sbin>hadoop fs -ls /input_dir/
```

Step 3: Retrieving Files from HDFS

The Hadoop command get copies files from HDFS back to the local filesystem. To retrieve example.txt, we can run the following command:

```
C:\hadoop-2.8.0\sbin>Hadoop fs -cat /input_dir/input_file.txt
```

Output: Hello world hello hi (which is stored in input_file .txt)

Step 4: Download the file:

Command: hadoop fs -get: Copies/Downloads files to the local file system Example:

```
hadoop fs -get /user/saurzcode/dir3/Samplefile.txt /home/
```

Step 5: Copy a file from source to destination

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

Command: hadoop fs -cp

Example: hadoop fs -cp /user/saurzcode/dir1/abc.txt /user/saurzcode/ dir2 **Step**

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

Command: hadoop fs -copyFromLocal URI

Example: hadoop fs -copyFromLocal /home/saurzcode/abc.txt /user/ saurzcode/abc.txt

copyToLocal

Command: hadoop fs -copyToLocal [-ignorecrc] [-crc] URI

Step 7: Move file from source to destination

Note:- Moving files across filesystem is not permitted.

Command: hadoop fs -mv

Example: hadoop fs -mv /user/saurzcode/dir1/abc.txt /user/saurzcode/ dir2

Step 8: Deleting Files from HDFS

```
C:\hadoop-2.8.0\sbin>hadoop fs -rm input_file.txt /input_dir/input_file.txt
```

Recursive version of delete:

Command: hadoop fs -rmr

Example: hadoop fs -rmr /user/saurzcode/

Step 9: Display last few lines of a file

Similar to tail command in Unix.

Usage : hadoop fs -tail

Example: hadoop fs -tail /user/saurzcode/dir1/abc.txt

Step 10: Display the aggregate length of a file

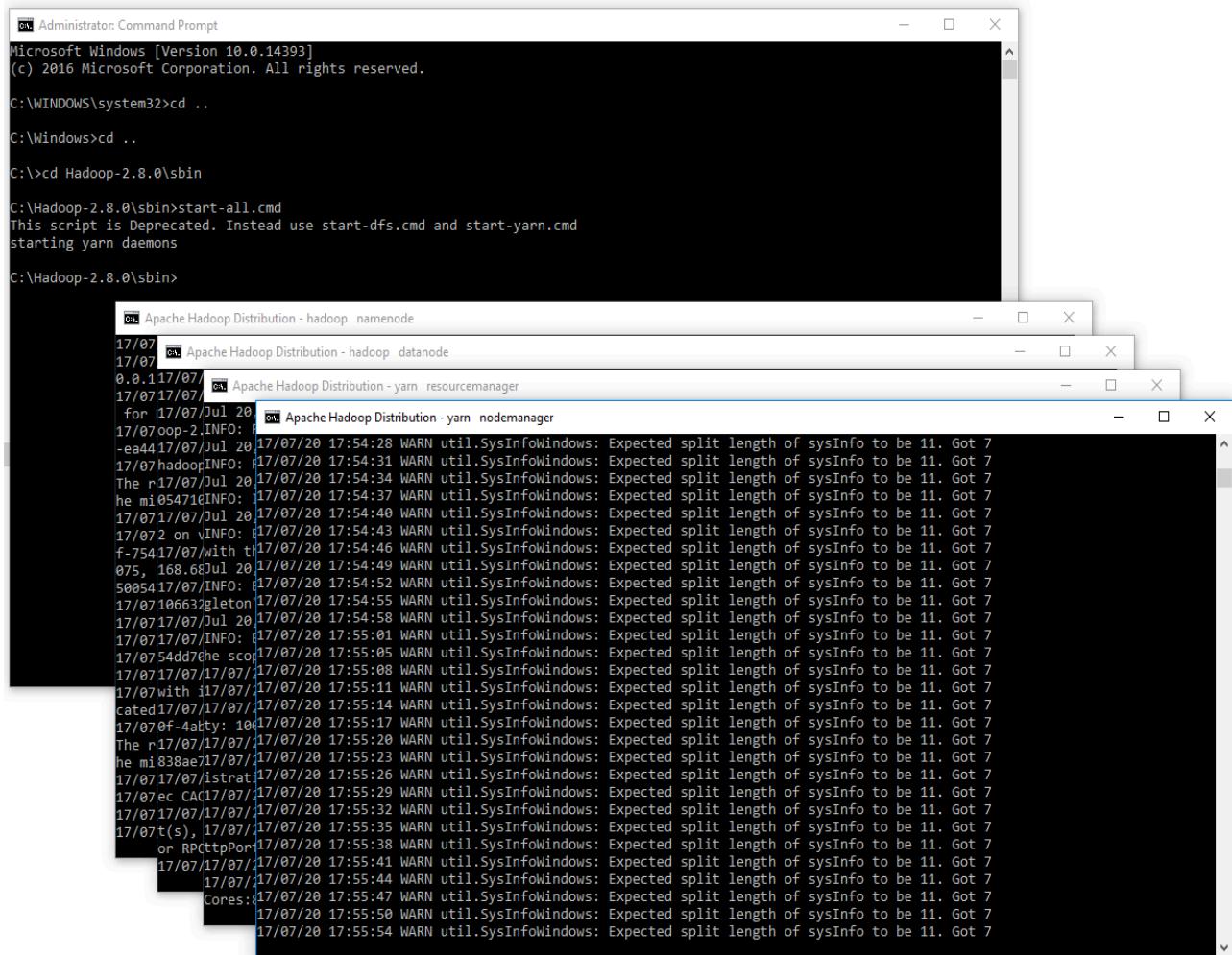
Command: hadoop fs -du

Example: hadoop fs -du /user/saurzcode/dir1/abc.txt

HADOOP OPERATION:

1. Open cmd in administrative mode and move to “C:/Hadoop-2.8.0/sbin” and start cluster

Start-all.cmd



The screenshot shows three overlapping command prompt windows. The top window is titled "Administrator: Command Prompt" and shows the command "start-all.cmd" being run. It outputs a warning message: "This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd starting yarn daemons". The bottom two windows are titled "Apache Hadoop Distribution - hadoop namenode" and "Apache Hadoop Distribution - yarn resourcemanager". Both show detailed log output from July 17, 2017, at 17:54:28, including numerous WARN and INFO messages related to sysInfoWindows splits.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd ..
C:\Windows>cd ..
C:\>cd Hadoop-2.8.0\sbin

C:\Hadoop-2.8.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Hadoop-2.8.0\sbin>

Apache Hadoop Distribution - hadoop namenode
17/07/17 17:54:28 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17 17:54:31 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
The r17/07/Jul 20 17:54:34 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
he mi05471@INFO: 17/07/20 17:54:37 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/Jul 20 17:54:40 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/20 17:54:43 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
f-75417/07/with t 17/07/20 17:54:46 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
075, 168.6@Jul 20 17/07/20 17:54:49 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
50054/17/07/INFO: 17/07/20 17:54:52 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/10663@gleton 17/07/20 17:54:55 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/Jul 20 17/07/20 17:54:58 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/INFO: 17/07/20 17:55:01 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/54dd7@hs sco 17/07/20 17:55:05 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/17/07/ 17/07/20 17:55:08 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/20 17:55:11 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
cated17/07/17/07/ 17/07/20 17:55:14 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/0f-4atty: 10 17/07/20 17:55:17 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
The r17/07/17/07/ 17/07/20 17:55:20 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
he mi838ae@17/07/ 17/07/20 17:55:23 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/istrat 17/07/20 17:55:26 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/20 17:55:29 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/17/07/ 17/07/20 17:55:32 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/0t(s), 17/07/ 17/07/20 17:55:35 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
or RPcttpPort 17/07/20 17:55:38 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/ 17/07/20 17:55:41 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/20 17:55:44 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
Cores: 17/07/20 17:55:47 SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/20 17:55:50 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/20 17:55:54 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
```

1. Create an input directory in

HDFS. hadoop fs -mkdir

/input_dir

2. Copy the input text file named input_file.txt in the input directory (input_dir) of HDFS.

```
hadoop fs -put C:/input_file.txt /input_dir
```

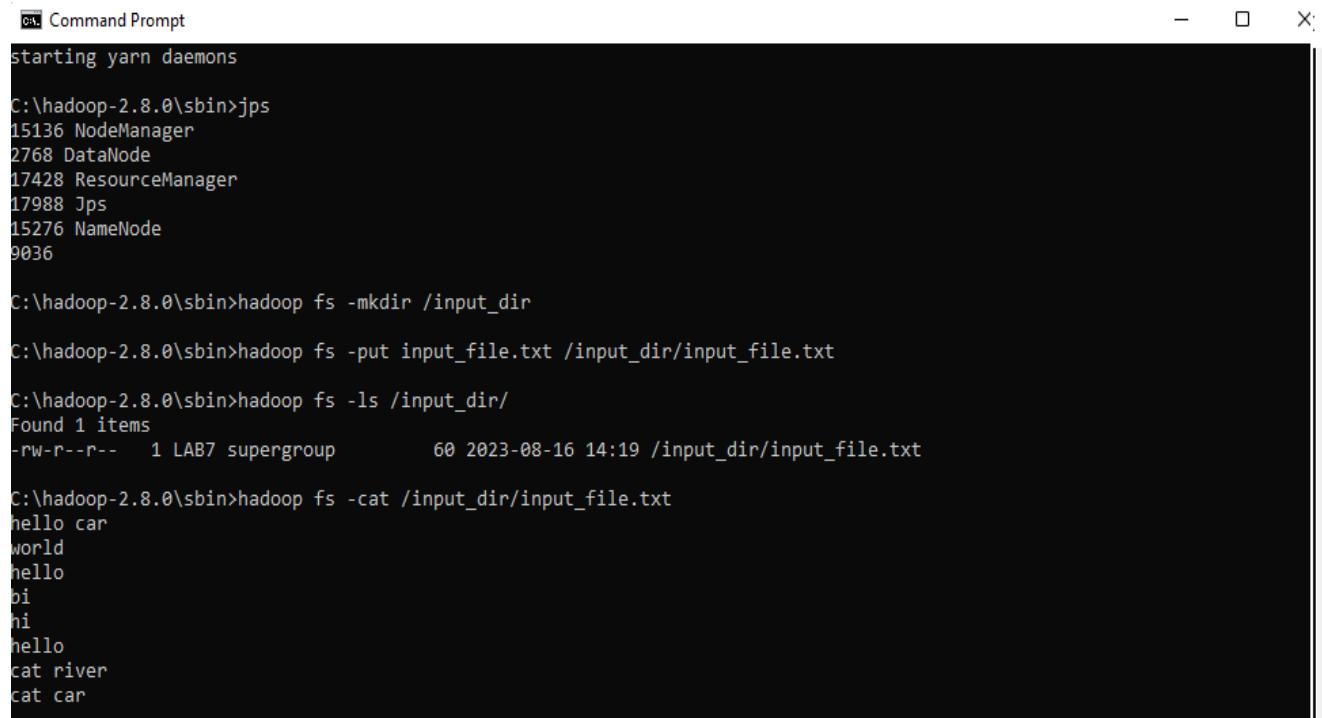
3. Verify input_file.txt available in HDFS input directory (input_dir).

```
hadoop fs -ls /input_dir/
```

Verify content of the copied file.

```
hadoop dfs -cat /input_dir/input_file.txt
```

OUTPUT:



The screenshot shows a Windows Command Prompt window with the title 'Command Prompt'. The window contains the following text:

```
starting yarn daemons
C:\hadoop-2.8.0\sbin>jps
15136 NodeManager
2768 DataNode
17428 ResourceManager
17988 Jps
15276 NameNode
9036

C:\hadoop-2.8.0\sbin>hadoop fs -mkdir /input_dir
C:\hadoop-2.8.0\sbin>hadoop fs -put input_file.txt /input_dir/input_file.txt
C:\hadoop-2.8.0\sbin>hadoop fs -ls /input_dir/
Found 1 items
-rw-r--r-- 1 LAB7 supergroup      60 2023-08-16 14:19 /input_dir/input_file.txt

C:\hadoop-2.8.0\sbin>hadoop fs -cat /input_dir/input_file.txt
hello car
world
hello
hi
hello
cat river
cat car
```

OTHER COMMANDS:

1. To leave Safe mode

```
hadoop dfsadmin --safemode leave
```

2. To delete file from HDFS directory

```
hadoop fs -rm -r /input_dir/input_file.txt
```

3. To delete directory from HDFS directory

```
hadoop fs -rm -r /input_dir
```

OUTPUT

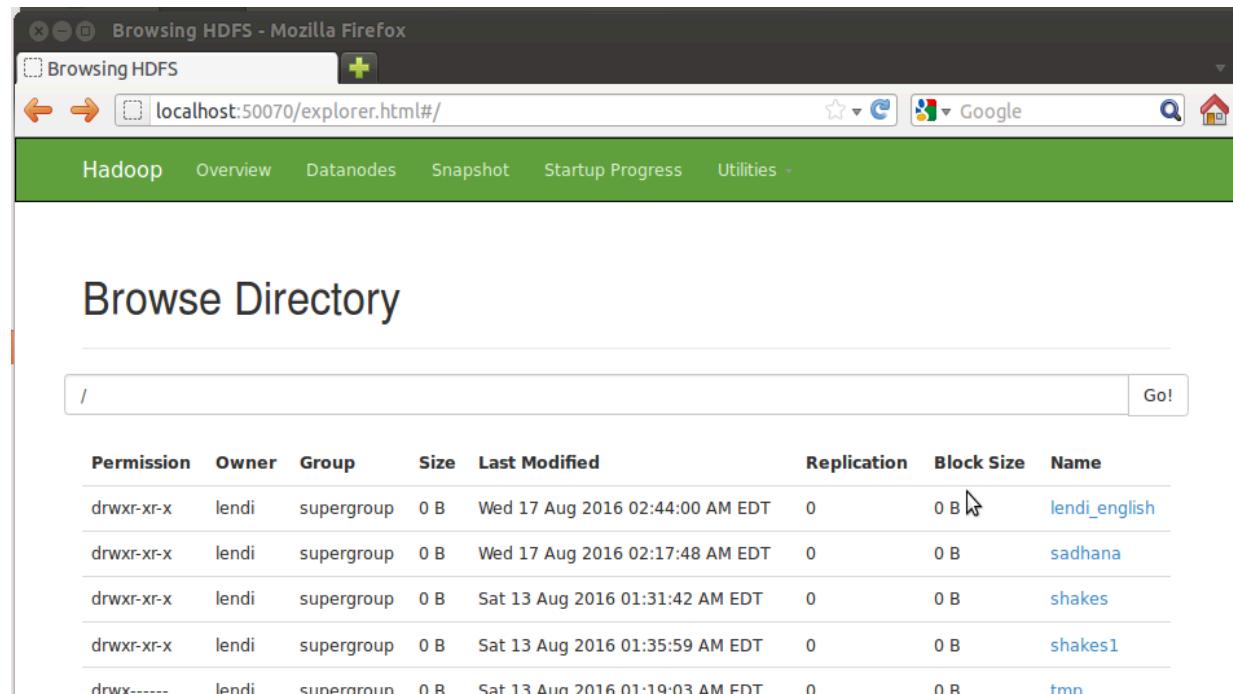
```
C:\>hadoop dfsadmin -safemode leave
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Safe mode is OFF

C:\>hadoop fs -rm -r /input_dir/input_file.txt
Deleted /input_dir/input_file.txt

C:\>hadoop fs -rm -r /input_dir
Deleted /input_dir

C:\>
```

OUTPUT:



The screenshot shows a Mozilla Firefox window titled "Browsing HDFS - Mozilla Firefox". The address bar displays "localhost:50070/explorer.html#/". The main content area is titled "Browse Directory" and shows a table of file and directory listings. The table has columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. The "Name" column contains links to files like "lendi_english", "sadhana", "shakes", "shakes1", and "tmp".

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	lendi	supergroup	0 B	Wed 17 Aug 2016 02:44:00 AM EDT	0	0 B	lendi_english
drwxr-xr-x	lendi	supergroup	0 B	Wed 17 Aug 2016 02:17:48 AM EDT	0	0 B	sadhana
drwxr-xr-x	lendi	supergroup	0 B	Sat 13 Aug 2016 01:31:42 AM EDT	0	0 B	shakes
drwxr-xr-x	lendi	supergroup	0 B	Sat 13 Aug 2016 01:35:59 AM EDT	0	0 B	shakes1
drwx-----	lendi	supergroup	0 B	Sat 13 Aug 2016 01:19:03 AM EDT	0	0 B	tmp

RESULT:

Thus, the implementation for file management in Hadoop was successfully executed.

EX.NO: 3

MATRIX MULTIPLICATION

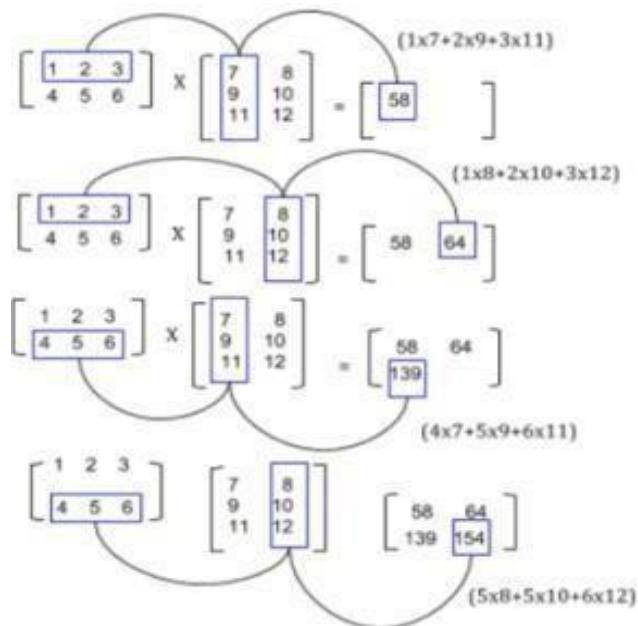
DATE:

AIM:

To Implement of Matrix Multiplication with Hadoop Map Reduce.

THEORY:

In mathematics, matrix multiplication or the matrix product is a binary operation that produces a matrix from two matrices. In more detail, if A is an $n \times m$ matrix and B is an $m \times p$ matrix, their matrix product AB is an $n \times p$ matrix, in which the m entries across a row of A are multiplied with the m entries down a column of B and summed to produce an entry of AB. When two linear transformations are represented by matrices, then the matrix product represents the composition of the two transformations.



ALGORITHM FOR MAP FUNCTION:

for each element m_{ij} of M do

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

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

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

ALGORITHM FOR REDUCE FUNCTION:

for each key (i,k) do
sort values begin with M by j in listM sort values begin with N by j in listN
multiply mij and njk for jth value of each listsum
up mij x njk return (i,k), $\Sigma_{j=1}^n mij \times njk$

HADOOP OPERATION:

Make sure that Hadoop is installed on your system with java idk Steps to follow

Step 1: Open Eclipse> File > New > Java Project > (Name it – MRProgramsDemo) > Finish

Step 2: Right Click > New > Package (Name as **com.mapreduce.wc**) > Finish

Step 3: Right Click on Package > New > Class (Name it - Matrixmultiply) **Step 4:**

Add Following Reference Libraries –

Right Click on Project > Build Path> Add External Archivals

1. C:/Hadoop/share/Hadoop->common/lib -> add all jar
2. C:/Hadoop/share/Hadoop-> client -> add all jar
3. C:/Hadoop/share/Hadoop-> mapreduce -> add all jar
4. C:/Hadoop/share/Hadoop-> yarn -> add all jar
4. C:/lib/hadoop-2.8.0/lib/Commons-cli-1.2.jar
5. C:/lib/hadoop-2.8.0/hadoop-core.jar

By Downloading the hadoop jar files with these links.

- Download Hadoop Common Jar files: <https://goo.gl/G4MyHp>
hadoop-common-2.2.0.jar
- Download Hadoop Mapreduce Jar File: <https://goo.gl/KT8yfB>
hadoop-mapreduce-client-core-2.7.1.jar

PROGRAM:

Creating Map file for Matrix Multiplication.

```
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);
    }
}
}
}
}

```

Creating Reduce file for Matrix Multiplication.

```

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)));
}
}

```

Creating MatrixMultiply.java file for

```

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();
    // M is an m-by-n matrix;
    N is an n-by-p matrix. 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);
}
}

```

Step 5: Uploading the M, N file which contains the matrix multiplication data to HDFS.

Create M.txt in sbin

M,0,0,1

M,0,1,2

M,1,0,3

M,1,1,4

Create N.txt in sbin

N,0,0,5 1,6 N,1,0,7

N,1,1,8

Run following commands in cmd prompt:

```

$ hadoop fs -mkdir /input_matrix/
$ hadoop fs -put M.txt / input_matrix
$ hadoop fs -put N.txt / input_matrix
$ hadoop fs -cat N.txt / input_matrix / M.txt
$ hadoop fs -cat N.txt / input_matrix / N.txt

```

Step 6: Export jar and run following hadoop command.

Step 1: Open Eclipse> open -> (MRProgramsDemo)project -> Right Click -> Export->java->JAR file

-> Next -> name it as matrix.jar Step 2: Open Command prompt

Sbin >hadoop jar C:/MRProgramsDemo\matrix com.mapreduce.wc/matrixmultiply /input_matrix/* /output_matrix

OUTPUT:

```
G:\hadoop\sbin>hadoop fs -put G:\Hadoop_Experiments\MatrixMultiply_M.txt /input_matrix
G:\hadoop\sbin>hadoop fs -ls /input_matrix
Found 1 items
-rw-r--r-- 1 hp supergroup          34 2021-04-06 00:07 /input_matrix/MatrixMultiply_M.txt

G:\hadoop\sbin>hadoop dfs -cat /input_matrix/MatrixMultiply_M.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
M,0,0,1
M,0,1,2
M,1,0,3
M,1,1,4
G:\hadoop\sbin>hadoop fs -put G:\Hadoop_Experiments\MatrixMultiply_N.txt /input_matrix
G:\hadoop\sbin>hadoop dfs -cat /input_matrix/MatrixMultiply_N.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
N,0,0,5
N,0,1,6
N,1,0,7
N,1,1,8
G:\hadoop\sbin>hadoop jar G:\Hadoop_Experiments\MatrixMultiply.jar com.mapreduce.wc/MatrixMultiply /input_matrix/* /output_matrix
```

```
G:\hadoop\sbin>
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=68
File Output Format Counters
  Bytes Written=36

G:\hadoop\sbin>hadoop dfs -cat /output_matrix/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
0,0,19.0
0,1,22.0
1,0,43.0
1,1,50.0

G:\hadoop\sbin>
```

RESULT:

Thus, a Implementation of Matrix Multiplication with Hadoop Map Reduce was executed successfully.

EX.NO: 4

WORD COUNT MAP REDUCE

DATE:

AIM:

To Run a basic Word Count MapReduce program to understand Map Reduce Paradigm.

THEORY:

MapReduce is a programming model used for efficient processing in parallel over large data-sets in a distributed manner. The data is first split and then combined to produce the final result. The libraries for MapReduce is written in so many programming languages with various different-different optimizations.

Workflow of MapReduce consists of 5 steps:

1. Splitting – The splitting parameter can be anything, e.g. splitting by space, comma, semicolon, or even by a new line ('\n').
2. Mapping – It takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (Key-Value pair).
3. Intermediate splitting – the entire process in parallel on different clusters. In order to group them in “Reduce Phase” the similar KEY data should be on same cluster.
4. Reduce – it is nothing but mostly group by phase
5. Combining – The last phase where all the data (individual result set from each cluster) is combined together to form a Result.

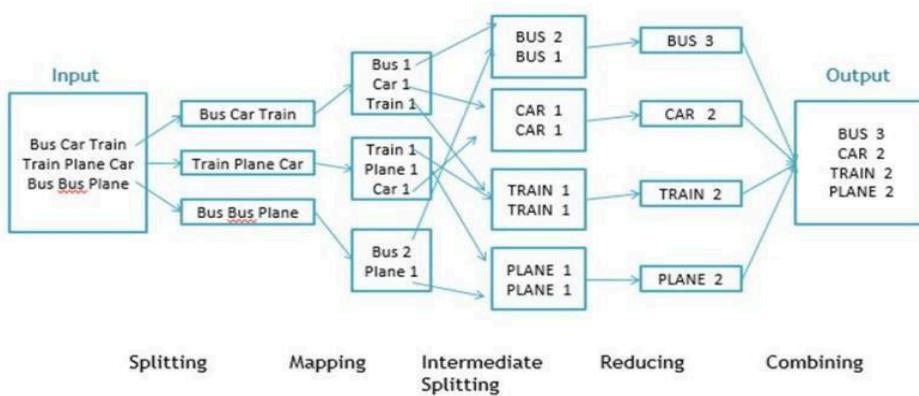


Fig. WorkFlow of MapReducing

PREPARE:

1. Download MapReduceClient.jar

(Link: <https://github.com/MuhammadBilalYar/HADOOP- INSTALLATION-ON-WINDOW-10/blob/master/MapReduceClient.jar>)

2. Download Input_file.txt

(Link: https://github.com/MuhammadBilalYar/HADOOP- INSTALLATION-ON-WINDOW-10/blob/master/input_file.txt)

Place both files in "C:/"

Make sure that Hadoop is installed on your system with java jdk Steps to follow

Step 1: Open Eclipse> File > New > Java Project > (Name it – MRProgramsDemo) >Finish

Step 2: Right Click > New > Package (Name it - PackageDemo) > Finish **Step**

3: Right Click on Package > New > Class (Name it - WordCount) **Step 4:** Add

Following Reference Libraries –

Right Click on Project > Build Path> Add External Archivals

- /usr/lib/hadoop-0.20/hadoop-core.jar
- Usr/lib/hadoop-0.20/lib/Commons-cli-1.2.jar

PROGRAM:

```
package PackageDemo;
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class WordCount {
    public static void main(String [] args) throws Exception
    {
        Configuration c=new Configuration();
        String[] files=new GenericOptionsParser(c,args).getRemainingArgs();
        Path input=new Path(files[0]);
        Path output=new Path(files[1]);
```

```

Job j=new Job(c,"wordcount");
j.setJarByClass(WordCount.class);
j.setMapperClass(MapForWordCount.class);
j.setReducerClass(ReduceForWordCount.class);
j.setOutputKeyClass(Text.class);
j.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(j, input);
FileOutputFormat.setOutputPath(j, output);
System.exit(j.waitForCompletion(true)?0:1);
}
public static class MapForWordCount extends Mapper<LongWritable, Text, Text, IntWritable>
{
public void map(LongWritable key, Text value, Context con) throws IOException, InterruptedException
{
String line = value.toString();
String[] words=line.split(",");for(String word: words )
{
Text outputKey = new Text(word.toUpperCase().trim());
IntWritable outputValue = new IntWritable(1);
con.write(outputKey, outputValue);
}
}
}
}
public static class ReduceForWordCount extends Reducer<Text, IntWritable, Text, IntWritable>
{
public void reduce(Text word, Iterable<IntWritable> values, Context con) throws IOException, InterruptedException
{
int sum = 0;
for(IntWritable value : values)
{
sum += value.get();
}
con.write(word, new IntWritable(sum));
}
}
}
}

```

Make Jar File

Right Click on Project > Export > Select export destination as Jar File > next > Finish

To Move this into Hadoop directly,

open the terminal and enter the following commands:

```
$ hadoop fs -put wordcountFile /input_dir
```

Run Jar file

Syntax: Hadoop jar jarfilename.jar packageName.ClassName PathToInputTextFile

PathToOutputDirectry

sbin> Hadoop jar MRProgramsDemo.jar PackageDemo.WordCount /input_dir /out

OUTPUT:

```
$ hadoop fs -ls out
1Found 1 item
-rw-r--r-- 1 training supergroup
20 2016-02-23 03:36 /user/training/MRDir1/part-r-00000
$ hadoop fs -cat out/*
CAR 4
TRAIN 6
```

HADOOP OPERATION:

1. Open cmd in Administrative mode and move to "C:/Hadoop-2.8.0/sbin" and start cluster

Start-all.cmd

The screenshot shows three command prompt windows. The top window is titled 'Administrator: Command Prompt' and contains the command 'start-all.cmd'. It outputs: 'This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd starting yarn daemons'. The bottom-left window is titled 'Apache Hadoop Distribution - hadoop namenode' and shows log entries from 17/07/2016 at 17:54:28. The bottom-right window is titled 'Apache Hadoop Distribution - yarn nodemanager' and also shows log entries from 17/07/2016 at 17:54:28.

```
C:\Windows\system32>d ..
C:\Windows>cd ..

C:\>cd Hadoop-2.8.0\sbin

C:\Hadoop-2.8.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Hadoop-2.8.0\sbin>

Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>d ..

Apache Hadoop Distribution - hadoop namenode
17/07 17/07/Jul 2016 17:54:28 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07 hadoopINFO: 17/07/2016 17:54:31 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
The r17/07/Jul 2016 17:54:34 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
he mi054716INFO: 17/07/2016 17:54:37 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/Jul 2016 17:54:40 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2 on \INFO: 17/07/2016 17:54:43 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
f-75417/07/with 17/07/2016 17:54:46 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
075, 168.63Jul 2016 17/07/2016 17:54:49 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
5085417/07/INFO: 17/07/2016 17:54:52 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/Jul 2016 17/07/2016 17:54:55 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/Jul 2016 17/07/2016 17:54:58 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/INFO: 17/07/2016 17:55:01 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/54dd7d\hadoop\sco 17/07/2016 17:55:05 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/17/07/ 17/07/2016 17:55:08 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/with 17/07/17/07/2016 17:55:11 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
cated17/07/17/07/ 17/07/2016 17:55:14 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/05-4atty: 104 17/07/2016 17:55:17 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
The r17/07/17/07/ 17/07/2016 17:55:20 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
he mi1838ae17/07/ 17/07/2016 17:55:23 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/17/07/ 17/07/2016 17:55:26 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/tec Cdc17/07/ 17/07/2016 17:55:29 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/17/07/ 17/07/2016 17:55:32 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/17/07/ 17/07/2016 17:55:35 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/17/07/ 17/07/2016 17:55:38 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
or R0tcpPort 17/07/2016 17:55:41 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/17/07/17/07/ 17/07/2016 17:55:44 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
cores 17/07/2016 17:55:47 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2016 17:55:50 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2016 17:55:54 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
```

2. Create an input directory in HDFS.

```
hadoop fs -mkdir /input_dir
```

3. Copy the input text file named input_file.txt in the input directory (input_dir) of HDFS.

```
hadoop fs -put C:/input_file.txt /input_dir/input_file.txt
```

4. Verify input_file.txt available in HDFS input directory .

```
hadoop fs -ls /input_dir/
```

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The window displays the following command-line session:

```
C:\> Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd/
C:\>cd Hadoop-2.8.0\sbin\
C:\Hadoop-2.8.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Hadoop-2.8.0\sbin>cd/
C:\>hadoop dfsadmin -safemode leave
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Safe mode is OFF

C:\>hadoop fs -mkdir /input_dir
C:\>hadoop fs -put C:/input_file.txt /input_dir
C:\>hadoop fs -ls /input_dir/
Found 1 items
-rw-r--r-- 1 Muhammad.Bilal supergroup      1888 2017-07-20 18:31 /input_dir/input_file.txt
C:\>
```

5. Verify content of the copied file.

```
hadoop dfs -cat /input_dir/input_file.txt
```

6. Run MapReduceClient.jar and also provide input and out directories.

```
>hadoop jar C:/hadoop-2.8.0/share/Hadoop/mapreduce/hadoop-mapreduce-examples-2.8.0.jar
```

```
>hadoop jar C:/hadoop-2.8.0/share/Hadoop/mapreduce/hadoop-mapreduce-examples-2.8.0.jar
wordcount /input_dir /out
```

7. Verify content for generated output file.

```
hadoop dfs -cat /out/*
```

OUTPUT

```
cmd Command Prompt
starting yarn daemons

C:\hadoop-2.8.0\sbin>jps
15136 NodeManager
2768 DataNode
17428 ResourceManager
17988 Jps
15276 NameNode
9036

C:\hadoop-2.8.0\sbin>hadoop fs -mkdir /input_dir
C:\hadoop-2.8.0\sbin>hadoop fs -put input_file.txt /input_dir/input_file.txt
C:\hadoop-2.8.0\sbin>hadoop fs -ls /input_dir/
Found 1 items
-rw-r--r-- 1 LAB7 supergroup      60 2023-08-16 14:19 /input_dir/input_file.txt

C:\hadoop-2.8.0\sbin>hadoop fs -cat /input_dir/input_file.txt
hello car
world
hello
hi
hi
hello
cat river
cat car
```

```
cmd Command Prompt
C:\hadoop-2.8.0\sbin>hadoop jar C:/hadoop-2.8.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.0.jar wordcount /in
put /out
23/08/16 16:16:20 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
23/08/16 16:16:20 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
23/08/16 16:16:20 INFO input.FileInputFormat: Total input files to process : 1
23/08/16 16:16:20 INFO mapreduce.JobSubmitter: number of splits:1
23/08/16 16:16:20 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local2017574452_0001
23/08/16 16:16:20 INFO mapreduce.Job: The url to track the job: http://localhost:8880/
23/08/16 16:16:20 INFO mapreduce.Job: Running job: job_local2017574452_0001
23/08/16 16:16:20 INFO mapred.LocalJobRunner: OutputCommitter set in config null
23/08/16 16:16:20 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
23/08/16 16:16:20 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output dire
ctory:false, ignore cleanup failures: false
23/08/16 16:16:20 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommit
ter
23/08/16 16:16:20 INFO mapred.LocalJobRunner: Waiting for map tasks
23/08/16 16:16:20 INFO mapred.LocalJobRunner: Starting task: attempt_local2017574452_0001_m_000000_0
23/08/16 16:16:20 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
23/08/16 16:16:20 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output dire
ctory:false, ignore cleanup failures: false
23/08/16 16:16:20 INFO util.ProcfsBasedProcessTree: ProcfsBasedProcessTree currently is supported only on Linux.
23/08/16 16:16:20 INFO mapred.Task: Using ResourceCalculatorProcessTree : org.apache.hadoop.yarn.util.WindowsBasedProce
ssTree@184b07db
23/08/16 16:16:20 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/input/input_file.txt:0+60
23/08/16 16:16:20 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
23/08/16 16:16:20 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
23/08/16 16:16:20 INFO mapred.MapTask: soft limit at 83886080
23/08/16 16:16:20 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
23/08/16 16:16:20 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
```

```
 Command Prompt
Reduce input records=7
Reduce output records=7
Spilled Records=14
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=0
Total committed heap usage (bytes)=524288000
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=60
File Output Format Counters
Bytes Written=46

C:\hadoop-2.8.0\sbin>hadoop fs -cat /out/*
bi      1
car     2
cat     2
hello   3
hi      1
river   1
world   1
```

RESULT:

Thus, the implementation of Wordcount with Hadoop Map Reduce was executed successfully.

EX.NO: 5

HIVE

DATE:

AIM:

To Installation of Hive along with practice examples.

THEORY:

Apache Hive is a data warehouse and an ETL tool which provides an SQL-like interface between the user and the Hadoop distributed file system (HDFS) which integrates Hadoop. It is built on top of Hadoop. It is a software project that provides data query and analysis. It facilitates reading, writing and handling wide datasets that stored in distributed storage and queried by Structure Query Language (SQL) syntax.

PREPARE:

These softwares should be prepared to install Hadoop 2.8.0 on window 10 64bit

1. Download Hadoop 2.8.0
2. Java JDK 1.8.0.zip
3. **Download Hive 2.1.0 :** <https://archive.apache.org/dist/hive/hive-2.1.0/>
4. **Download Derby Metastore 10.12.1.1:** <https://archive.apache.org/dist/db/derby/db-derby-10.12.1.1/>
5. **Download hive-site.xml :**

<https://drive.google.com/file/d/1qqAo7ROfr5Q6O-GTom6Rji3TdufP81zd/view?usp=sharing>

PROCEDURE:

STEP - 1: Download and Extract the Hive file:

[1] Extract file apache-hive-2.1.0-bin.tar.gz and place under "D:\Hive", you can use any preferred location

This PC > DATA (D:) > Hive > apache-hive-2.1.0-bin.tar > apache-hive-2.1.0-bin		
Name	Date modified	Type
apache-hive-2.1.0-bin	8/29/2018 6:07 PM	File folder

[2] Copy the leaf folder "apache-hive-2.1.0-bin" and move to the root folder "D:\Hive".

STEP - 2: Extract the Derby file

Similar to Hive, extract file db-derby-10.12.1.1-bin.tar.gz and place under "D:\Derby", you can use any preferred location:

This PC > DATA (D:) > Derby > db-derby-10.12.1.1-bin	
Name	Date modified
bin	8/29/2018 6:15 PM
demo	8/29/2018 6:15 PM
docs	8/29/2018 6:15 PM
javadoc	8/29/2018 6:15 PM
lib	8/29/2018 6:15 PM
test	8/29/2018 6:15 PM
index.html	9/14/2015 1:28 AM
KEYS	9/20/2015 7:31 PM
LICENSE	9/20/2015 7:31 PM
NOTICE	9/20/2015 7:31 PM
RELEASE-NOTES.html	9/20/2015 7:31 PM

STEP - 3: Moving hive-site.xml file

Drop the downloaded file “hive-site.xml” to hive configuration location “D:\Hive\apache-hive-2.1.0-bin\conf”.

This PC > DATA (D:) > Hive > apache-hive-2.1.0-bin > conf	
Name	Date modified
beeline-log4j2.properties.template	6/3/2016 4:13 PM
hive-default.xml.template	6/17/2016 5:33 AM
hive-env.sh.template	6/3/2016 4:13 PM
hive-exec-log4j2.properties.template	6/3/2016 4:13 PM
hive-log4j2.properties.template	6/3/2016 4:13 PM
hive-site.xml	8/28/2018 6:01 PM
ivysettings.xml	6/10/2016 2:30 PM
llap-cli-log4j2.properties.template	6/3/2016 4:13 PM
llap-daemon-log4j2.properties.template	6/3/2016 4:13 PM
parquet-logging.properties	6/9/2016 12:17 AM

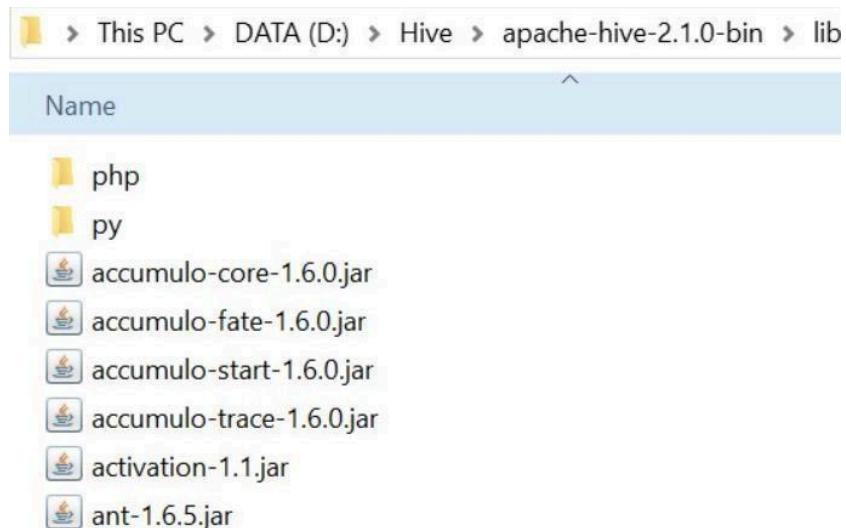
STEP - 4: Moving Derby libraries

Next, need to drop all derby library to hive library location :

- [1] Move to library folder under derby location D:\Derby\db-derby-10.12.1.1-bin\lib.



2] Select all , copy and paste all libraries from derby to hive location D:\Hive\apache-hive-2.1.0-bin\lib.



STEP - 5: Configure Environment variables

Set the path for the following Environment variables (User Variables) on windows 10 –

- HIVE_HOME - D:\Hive\apache-hive-2.1.0-bin
- HIVE_BIN - D:\Hive\apache-hive-2.1.0-bin\bin
- HIVE_LIB - D:\Hive\apache-hive-2.1.0-bin\lib
- DERBY_HOME - D:\Derby\db-derby-10.12.1.1-bin
- HADOOP_USER_CLASSPATH_FIRST - true

This PC -> Right Click -> Properties -> Advanced System Settings -> Advanced -> Environment Variables

Variable name:

Variable value:

STEP - 6: Configure System variables

Next onward need to set System variables, including Hive bin directory path: HADOOP_USER_CLASSPATH_FIRST
- true

Variable: Path

Value:

1. D:\Hive\apache-hive-2.1.0-bin\bin
2. D:\Derby\db-derby-10.12.1.1-bin\bin

D:\Hadoop\hadoop-2.8.0\share\hadoop\mapreduce\lib*
D:\Hadoop\hadoop-2.8.0\share\hadoop\mapreduce*
D:\Hadoop\hadoop-2.8.0\share\hadoop\common\lib*
D:\Java\jdk1.8.0_171\bin
D:\Hive\apache-hive-2.1.0-bin\bin
D:\Derby\db-derby-10.12.1.1-bin\bin

STEP - 7: Working with hive-site.xml

Download hive-site.xml and paste it in D:/Hive/apache-hive-2.1.0-bin/conf/hive-site.xml

- hive-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration><property> <name>javax.jdo.option.ConnectionURL</name>
<value>jdbc:derby://localhost:1527/metastore_db;create=true</value>
<description>JDBC connect string for a JDBC metastore</description>
</property><property>
<name>javax.jdo.option.ConnectionDriverName</name>
<value>org.apache.derby.jdbc.ClientDriver</value>
<description>Driver class name for a JDBC metastore</description>
</property>
<property>
<name>hive.server2.enable.impersonation</name>
<description>Enable user impersonation for HiveServer2</description>
<value>true</value>
</property>
<property>
```

```

<name>hive.server2.authentication</name>
<value>NONE</value>
<description> Client authentication types. NONE: no authentication check LDAP: LDAP/AD based
authentication KERBEROS: Kerberos/GSSAPI authentication CUSTOM: Custom authentication provider
(Use with property hive.server2.custom.authentication.class) </description>
</property>
<property>
<name>datanucleus.autoCreateTables</name>
<value>True</value>
</property>
</configuration>

```

STEP - 8: Start the Hadoop

Here need to start Hadoop first -

Open command prompt and change directory to "D:\Hadoop\hadoop-2.8.0\sbin" and type "**start-all.cmd**" to start apache.

```
D:\Hadoop\hadoop-2.8.0\sbin>start-all.cmd
```

It will open four instances of cmd for following tasks –

- Hadoop Datanaode
- Hadoop Namenode
- Yarn Nodemanager
- Yarn Resourcemanager



It can be verified via browser also as –

- Namenode (hdfs) - http://localhost:50070
- Datanode - http://localhost:50075
- All Applications (cluster) - http://localhost:8088 etc.

The screenshot shows the Hadoop YARN ResourceManager UI at <http://localhost:8088/cluster>. The main title is "All Applications". On the left, there's a sidebar with a tree view:

- Cluster
 - About
 - Nodes
 - Node Labels
 - Applications
 - NEW
 - NEW_SAVING
 - SUBMITTED
 - ACCEPTED
 - RUNNING
 - FINISHED
 - FAILED
 - KILLED
 - Scheduler
- + Tools

The main content area displays "Cluster Metrics" and "Cluster Nodes Metrics" tables. Below them is a "Scheduler Metrics" table. At the bottom, there's a table header for "Applications" with columns: ID, User, Name, Application Type, Queue, Application Priority, StartTime, FinishTime, State, FinalStatus, Running Containers, Allocated CPU VCores, Allocated Memory MB, and % Qu. A message "No data available in table" is shown above the table body, which says "Showing 0 to 0 of 0 entries".

Since the ‘start-all.cmd’ command has been deprecated so you can use below command in order wise -

- “start-dfs.cmd” and
- “start-yarn.cmd”

STEP - 9: Start Derby server

Post successful execution of Hadoop, change directory to

```
>cd "D:\Derby\db-derby-10.12.1.1-bin\bin"
```

and type “**startNetworkServer -h 0.0.0.0**” to start derby server.

```
D:\Derby\db-derby-10.12.1.1-bin\bin>startNetworkServer -h 0.0.0.0
```

OUTPUT:

```
D:\Derby\db-derby-10.12.1.1-bin\bin>startNetworkServer -h 0.0.0.0
Thu Aug 30 10:40:27 IST 2018 : Security manager installed using the Basic server security policy.

Thu Aug 30 10:40:28 IST 2018 : Apache Derby Network Server - 10.12.1.1 - (1704137) started and ready to accept connections on port 1527
```

STEP - 10: Start the Hive

Derby server has been started and ready to accept connection so open a new command prompt under administrator privileges and move to hive directory as “

```
> cd D:\Hive\apache-hive-2.1.0-bin\bin" –
```

[1] Type “jps -m” to check NetworkServerControl

```
D:\Hive\apache-hive-2.1.0-bin\bin>jps -m
384 NameNode
22100 NetworkServerControl start -h 0.0.0.0
17960 Jps -m
19944 NodeManager
4776 ResourceManager
23468 DataNode
```

[2] Type “hive” to execute hive server.

```
D:\Hive\apache-hive-2.1.0-bin\bin>hive
```

OUTPUT:

```
D:\Hive\apache-hive-2.1.0-bin\bin>hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/Hive/apache-hive-2.1.0-bin/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/D:/Hadoop/hadoop-2.8.0/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]
ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.
Connecting to jdbc:hive2://
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>
```

Hive installed Successfully!

PROGRAM:

HIVE QUARIES AND OUTPUT:

[1] Create Database in Hive -

```
hive>CREATE DATABASE IF NOT EXISTS TRAINING;
```

```
hive> CREATE DATABASE IF NOT EXISTS TRAINING;
OK
No rows affected (1.603 seconds)
hive>
```

[2] Show Database -

```
hive>SHOW DATABASES;
```

```
hive> SHOW DATABASES;
OK
default
training
2 rows selected (0.382 seconds)
hive>
```

[3] Creating Hive Tables -

```
hive>CREATE TABLE IF NOT EXISTS testhive(col1 char(10), col2 char(20));
```

```
hive> USE TRAINING;
OK
No rows affected (0.047 seconds)
hive> CREATE TABLE IF NOT EXISTS testhive(col1 char(10),col2 char(20));
OK
No rows affected (0.669 seconds)
hive>
```

```
hive> CREATE TABLE IF NOT EXISTS students(
. . > id BIGINT COMMENT 'unique id for each student',
. . > name STRING COMMENT 'student name',
. . > age INT COMMENT 'student age',
. . > fee DOUBLE COMMENT 'college fee',
. . > city STRING COMMENT 'cities to which students belongs',
. . > state STRING COMMENT 'student home address state',
. . > zip BIGINT COMMENT 'student address zip code')
. . > COMMENT 'this table holds the demography info for each student'
. . > ROW FORMAT DELIMITED
. . > FIELDS TERMINATED BY '|'
. . > LINES TERMINATED BY '\n'
. . > STORED AS TEXTFILE
. . > LOCATION '/user/hive/warehouse/training.db/students';
OK
No rows affected (0.327 seconds)
hive>
```

[4] DESCRIBE Table Command in Hive -

hive>describe students

```
hive> describe students;
OK
id bigint unique id for each student
name string student name
age int student age
fee double college fee
city string cities to which students belongs
state string student home address state
zip bigint student address zip code
7 rows selected (0.135 seconds)
hive>
```

[5] LOAD Command for Inserting Data Into Hive Tables

Create a sample text file using ‘|’ delimiter :



students.txt - Notepad

File Edit Format View Help

ID	name	age	fee	city	state	zip
1	Kendall	22	25874	Kulti-Barakar	WB	451333
2	Mikayla	25	35367	Jalgaon	Maharastra	710179
3	Raven	20	49103	Rewa	Madhya Pradesh	392423
4	Carla	19	27121	Pilibhit	UP	769853
5	Edward	21	32053	Tuticorin	Tamil Nadu	368262

hive>LOAD DATA LOCAL INPATH “D:/students.txt” OVERWRITE INTO TABLE STUDENTS;

```
hive> LOAD DATA LOCAL INPATH 'D:/students.txt' OVERWRITE INTO TABLE students;
Loading data to table training.students
OK
No rows affected (1.798 seconds)
hive>
```

[6] Retrieve Data from Table -

```
hive>SELECT * FROM STUDENTS;
```

```
hive> SELECT * FROM students;
OK
  name    city   state
  1 Kendall 22 25874.0 Kulti-Barakar WB 451333
  2 Mikayla 25 35367.0 Jalgaon Maharastra 710179
  3 Raven 20 49103.0 Rewa Madhya Pradesh 392423
  4 Carla 19 27121.0 Pilibhit UP 769853
  5 Edward 21 32053.0 Tuticorin Tamil Nadu 368262
  6 Wynter 21 43956.0 Surendranagar GJ 457441
  7 Patrick 19 19050.0 Mumbai MH 580220
  8 Hayfa 18 15590.0 Amroha UP 470705
  9 Raven 16 37836.0 Cuddalore TN 787001
19 rows selected (0.351 seconds)
hive>
```

[7] Create another Table -

The following query creates a table named **employee** :

```
hive> CREATE TABLE IF NOT EXISTS employee ( eid int, name String,salary String, destination String)
```

```
COMMENT 'Employee details'
```

```
ROW FORMAT DELIMITED
```

```
FIELDS TERMINATED BY '\t'
```

```
LINES TERMINATED BY '\n'
```

```
STORED AS TEXTFILE; OUTPUT:
```

```
OK
```

```
Time taken: 5.905 seconds
```

```
hive>
```

[8] Alter

Table - Syntax:

The statement takes any of the following syntaxes based on what attributes we wish to modify in a table.

```
hive> ALTER TABLE name RENAME TO new_name
```

```
hive> ALTER TABLE name ADD COLUMNS (col_spec[, col_spec ...])
```

```
hive> ALTER TABLE name DROP [COLUMN] column_name
```

```
hive> ALTER TABLE name CHANGE column_name new_name new_type
```

```
hive> ALTER TABLE name REPLACE COLUMNS (col_spec[, col_spec ...])
```

Example:

The following query renames the table from employee to emp:

```
hive> ALTER TABLE employee RENAME TO emp;
```

The following queries rename the column name and column data type using the above data:

```
hive> ALTER TABLE employee CHANGE name ename String;
```

```
hive> ALTER TABLE employee CHANGE salary salary Double;
```

The following query adds a column named dept to the employee table:

```
hive> ALTER TABLE employee ADD COLUMNS (dept STRING COMMENT 'Department name');
```

[9] Drop Table-

The syntax is as follows:

```
hive>DROP TABLE [IF EXISTS] table_name;
```

The following query drops a table named employee:

```
hive> DROP TABLE IF EXISTS employee;
```

On successful execution of the query, you get to see the following response:

OUTPUT:

OK

Time taken: 5.3 seconds

```
hive>
```

The following query is used to verify the list of tables after deleting employee table:

```
hive> SHOW TABLES;
```

OUTPUT:

emp

ok

Time taken: 2.1 seconds

```
hive>
```

RESULT:

Thus, a procedure to installation of HIVE and commands are executed successfully.

EX NO: 6.1

HBASE

DATE:

AIM:

To Installation of HBase along with Practice examples.

THEORY:

Apache **HBase** is an open source non-relational (NoSQL) distributed column-oriented database that runs on top of HDFS and real-time read/write access to those large data-sets. Initially, it was Google Big Table, afterwards it was re-named as HBase and is primarily written in Java, designed to provide quick random access to huge amounts of the data-set.

In brief, the HBase can store massive amounts of data from terabytes to petabytes and allows fast random reads and writes that cannot be handled by the Hadoop. Even relational databases (RDBMS) cannot handle a variety of data that is growing exponentially.

HBase can be installed in three modes. The features of these modes are mentioned below.

[1] **Standalone mode installation** (No dependency on Hadoop system)

- This is default mode of HBase
- It runs against local file system
- It doesn't use Hadoop HDFS
- Only HMaster daemon can run
- Not recommended for production environment
- Runs in single JVM

[2] **Pseudo-Distributed mode installation** (Single node Hadoop system + HBase installation)

- It runs on Hadoop HDFS
- All Daemons run in single node
- Recommend for production environment

[3] **Fully Distributed mode installation** (Multi node Hadoop environment + HBase installation)

- It runs on Hadoop HDFS
- All daemons going to run across all nodes present in the cluster
- Highly recommended for production environment

PREPARE:

These softwares should be prepared to install Hadoop 2.8.0 on window 10 64bit

1. Download Hadoop 2.8.0

2. Java JDK 1.8.0.zip

3. Download HBase 1.4.7

<http://www.apache.org/dyn/closer.lua/hbase/>

PROCEDURE:

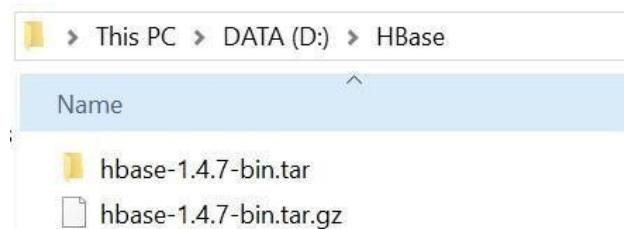
Hbase - Standalone mode installation

Here, we will go through the Standalone mode installation with Hbase on Windows 10.

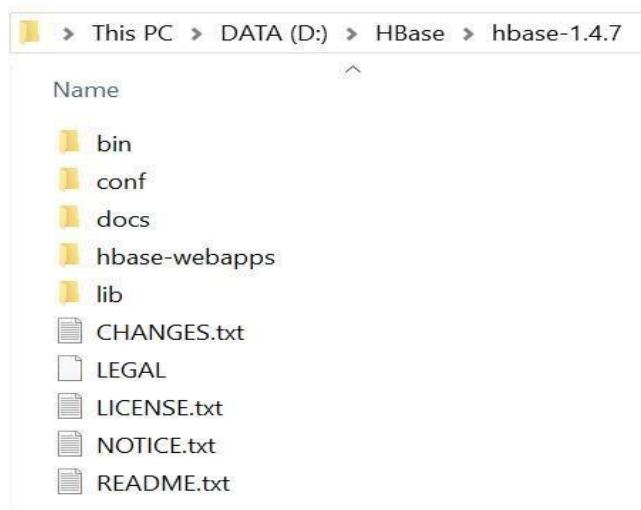
STEP - 1: Extract the HBase file

Extract file hbase-1.4.7-bin.tar.gz and place under "D:\HBase", you can use any preferred location:

[1] You will get again a tar file post extraction –



[2] Go inside of hbase-1.4.7-bin.tar folder and extract again. Then Copy the leaf folder “hbase-1.4.7” and move to the root folder "D:\HBase" folders:

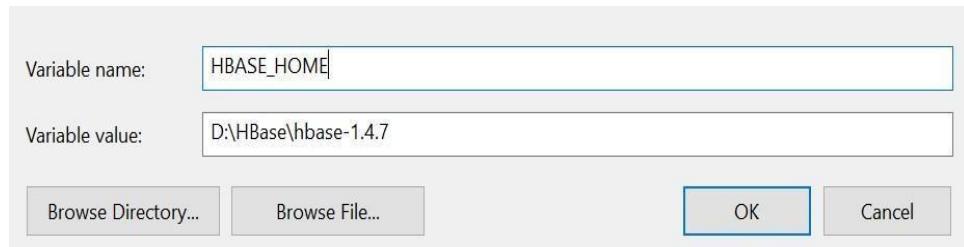


STEP - 2: Configure Environment variable

Set the path for the following Environment variable (User Variables) on windows 10 –

- **HBASE_HOME - D:\HBase\hbase-1.4.7**

This PC -> Right Click -> Properties -> Advanced System Settings -> Advanced -> Environment Variables

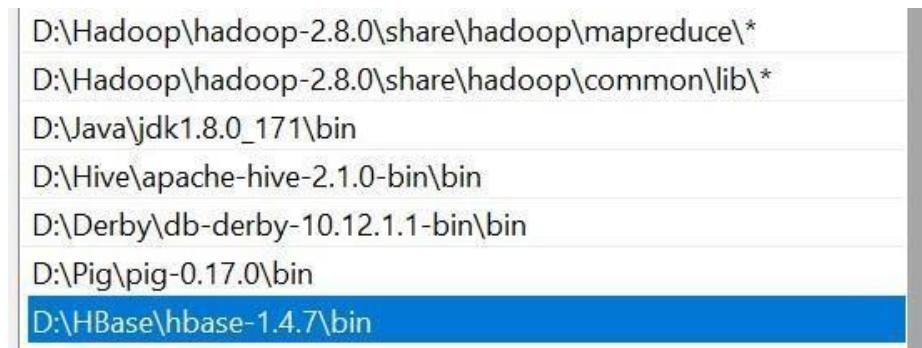


STEP - 3: Configure System variable

Next onward need to set System variable, including Hive bin directory path: Variable:

Path

Value: >D:\HBase\hbase-1.4.7\bin

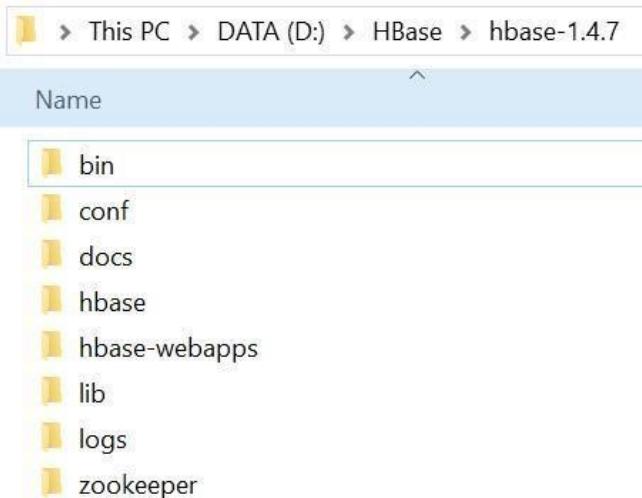


STEP - 4: Create required folders

Create some dedicated folders -

1. Create folder "hbase" under "D:\HBase\hbase-1.4.7".
2. Create folder "zookeeper" under "D:\HBase\hbase-1.4.7".

For example -



STEP - 5: Configured required files

Next, essential to configure two key files with minimal required details –

- hbase-env.cmd
- hbase-site.xml

[1] Edit file D:/HBase/hbase-1.4.7/conf/hbase-env.cmd, mention JAVA_HOME path in the location and save this file.

```
@rem set JAVA_HOME=c:\apps\java
set JAVA_HOME=%JAVA_HOME%
@rem set JAVA_HOME=c:\apps\java
set JAVA_HOME=%JAVA_HOME%
```

[2] Edit file D:/HBase/hbase-1.4.7/conf/hbase-site.xml, paste below xml paragraph and save this file.

```
<configuration>
<property>
<name>hbase.rootdir</name>
<value>file:///D:/HBase/hbase-1.4.7/hbase</value>
</property>
<property>
<name>hbase.zookeeper.property.dataDir</name>
<value>/D:/HBase/hbase-1.4.7/zookeeper</value>
</property>
```

```

<property>
<name> hbase.zookeeper.quorum</name>
<value>127.0.0.1</value>
</property>
</configuration>

```

All HMaster and ZooKeeper activities point out to this hbase-site.xml.

[3] Edit file hosts (C:/Windows/System32/drivers/etc/hosts), mention localhost IP and save this file.

127.0.0.1 localhost

```

# For example:
#
#      102.54.94.97      rhino.acme.com      # source server
#      38.25.63.10      x.acme.com          # x client host

# localhost name resolution is handled within DNS itself.
# 127.0.0.1      localhost
# ::1            localhost

127.0.0.1      localhost

```

STEP - 6: Start HBase

Here need to start HBase first :

Open command prompt and change directory to "D:\HBase\hbase-1.4.7\bin" and type "start-hbase.cmd" to start HBase.

D:\HBase\hbase-1.4.7\bin>start-hbase.cmd

It will open a separate instances of cmd for following tasks –

- HBase Master

```

2018-09-07 16:23:24,230 INFO [AM.ZK.Worker-pool5-t9] master.RegionStates: Transition state=OPENING, ts=1536317604053, server=g1ml22524.mindtree.com,55627,1536317586482} to state=OPEN, ts=1536317604230, server=g1ml22524.mindtree.com,55627,1536317586482}
2018-09-07 16:23:24,235 INFO [PostOpenDeployTasks:80168775cc41e25ffd701cac380c7422] b w test,,1536315609619.80168775cc41e25ffd701cac380c7422. with server=g1ml22524.mindtree
2018-09-07 16:23:24,243 INFO [AM.ZK.Worker-pool5-t11] master.RegionStates: Offlined 6 om g1ml22524.mindtree.com,55295,1536315760228
2018-09-07 16:23:24,249 INFO [AM.ZK.Worker-pool5-t12] master.RegionStates: Offlined 6 om g1ml22524.mindtree.com,55295,1536315760228
2018-09-07 16:23:24,254 INFO [AM.ZK.Worker-pool5-t13] master.RegionStates: Transition state=OPENING, ts=1536317604033, server=g1ml22524.mindtree.com,55627,1536317586482} to state=OPEN, ts=1536317604254, server=g1ml22524.mindtree.com,55627,1536317586482}
2018-09-07 16:23:24,260 INFO [AM.ZK.Worker-pool5-t15] master.RegionStates: Offlined 6 om g1ml22524.mindtree.com,55295,1536315760228

```

STEP - 7: Validate HBase

Post successful execution of HBase, verify the installation using following commands –

- hbase –version
- jps

```
D:\HBase\hbase-1.4.7\bin>hbase -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
```

```
D:\HBase\hbase-1.4.7\bin>jps
20768 HMaster
14040 Jps
```

If we can see HMaster is in running mode, then our installation is okay.

STEP - 8: Execute HBase Shell

The standalone mode does not require Hadoop daemons to start. HBase can run independently. HBase shell can start by using "hbase shell" and it will enter into interactive shell mode –

```
D:\HBase\hbase-1.4.7\bin>hbase shell
```

```
D:\HBase\hbase-1.4.7\bin>hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/D:/HBase/hbase-1.4.7/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLogger
Binder.class]
SLF4J: Found binding in [jar:file:/D:/Hadoop/hadoop-2.8.0/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/s
lf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 1.4.7, r763f27f583cf8fd7acf79fb6f3af57f1615dbf9b, Tue Aug 28 14:40:11 PDT 2018
hbase(main):001:0>
```

HBase installed !!

PROGRAM:

QUERIES AND OUTPUT:

[1] Create a simple table-

hBase>create 'student', 'bigdata'

```
hbase(main):001:0> create 'student', 'bigdata'
0 row(s) in 1.7430 seconds

=> Hbase::Table - student
hbase(main):002:0>
```

[2] List the table has been created-

hBase>list

```
hbase(main):002:0> list
TABLE

student

test

test_table

3 row(s) in 0.0180 seconds

=> ["student", "test", "test_table"]
hbase(main):003:0>
```

[3] Insert some data to above created table-

hBase>put 'tablename', 'rowname', 'columnvalue', 'value'

hBase>put 'student', 'row1', 'bigdata:hadoop', 'hadoop cause'

```
hbase(main):006:0> put 'student', 'row1', 'bigdata:hadoop', 'hadoop cause'
0 row(s) in 0.0130 seconds
```

[4] List all rows in the table-

hBase>scan 'student'

```
hbase(main):007:0> scan 'student'
```

```
hbase(main):010:0> scan 'student'
ROW                                COLUMN+CELL
row1                               column=bigdata:hadoop, timestamp=1536319058700, value=hadoop course
row1                               column=bigdata:hive, timestamp=1536319159071, value=hive for analysis
row1                               column=bigdata:pig, timestamp=1536319210621, value=pig for unstructure data
1 row(s) in 0.0080 seconds

hbase(main):011:0>
```

[5] Disabling a Table using HBase Shell

To delete a table or change its settings, you need to first disable the table using the disable command. You can re-enable it using the enable command.

Given below is the syntax to disable a table:

```
hbase>disable 'emp'
```

Example

Given below is an example that shows how to disable a table.

```
hbase> disable 'emp'
```

```
0 row(s) in 1.2760 seconds
```

Verification

After disabling the table, you can still sense its existence through **list** and **exists** commands. You cannot scan it. It will give you the following error.

```
hbase> scan 'emp'
```

```
ROW      COLUMN + CELL
```

```
ERROR: emp is disabled.
```

is_disabled

This command is used to find whether a table is disabled. Its syntax is as follows.

```
hbase> is_disabled 'table name'
```

The following example verifies whether the table named emp is disabled. If it is disabled, it will return true and if not, it will return false.

```
hbase(main):031:0> is_disabled 'emp'
```

```
true
```

```
0 row(s) in 0.0440 seconds
```

disable_all

This command is used to disable all the tables matching the given regex. The syntax for **disable_all** command is given below.

```
hbase> disable_all 'r.*'
```

Suppose there are 5 tables in HBase, namely raja, rajani, rajendra, rajesh, and raju. The following code will disable all the tables starting with **raj**.

```
hbase(main):002:07> disable_all 'raj.*'
```

raja

rajani

rajendra

rajesh

raju

Disable the above 5 tables (y/n)?

y

5 tables successfully disabled

RESULT:

Thus, a procedure to installation of HBase and queries are executed successfully.

EX.NO: 6.2

THRIFT

DATE:

AIM:

To Installing thrift along with Practice examples.

THEORY:

Apache Thrift is a RPC framework founded by facebook and now it is an Apache project. Thrift lets you define data types and service interfaces in a language neutral definition file. That definition file is used as the input for the compiler to generate code for building RPC clients and servers that communicate over different programming languages.

software framework, for scalable cross-language services development, combines a software stack with a code generation engine to build services that work efficiently and seamlessly between C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Node.js, Smalltalk, OCaml and Delphi and other languages.

PROCEDURE:

Step 1: Installing Apache Thrift in Windows

Installation Thrift can be a tiresome process. But for windows the compiler is available as a prebuilt exe. Download [thrift.exe](#) and add it into your environment variables.

- **Download Apache Thrift**

<https://archive.apache.org/dist/thrift/0.8.0/thrift-0.8.0.exe>

Step 2: Build and Install the Apache Thrift compiler:

Refer

1. <https://thrift.apache.org/>
2. <https://dzone.com/articles/apache-thrift-java-quickstart>

You will then need to build the Apache Thrift compiler and install it. See the installing Thrift guide for any help with this step.

Step 3: Writing a .thrift file

After the Thrift compiler is installed, you will need to create a thrift file. This file is an interface definition made up of thrift types and Services. The services you define in this file are implemented by the server and are called by any clients. The Thrift compiler is used to generate your Thrift File into source code which is used by the different client libraries and the server you write. To generate the source from a thrift file run

```
thrift --gen <language> <Thrift filename>
```

PROGRAM:

1.Example definition file (add.thrift)

```
namespace java com.eviac.blog.samples.thrift.server // defines the namespace
```

```
typedef i32 int //typedefs to get convenient names for your types
```

```
service AdditionService { // defines the service to add two numbers  
    int add(1:int n1, 2:int n2), //defines a method  
}
```

2.Compiling Thrift definition file

To compile the .following command.

```
thrift --gen <language> <Thrift filename>
```

For my example the command is,

```
thrift --gen java add.thrift
```

After performing the command, inside *gen-java* directory you'll find the source codes which is useful for building RPC clients and server. it will create a java code called *AdditionService.java*

Writing a service handler

Service handler class is required to implement the *AdditionService.Iface* interface.

Example service handler (AdditionServiceHandler.java)

```
package com.eviac.blog.samples.thrift.server;  
import org.apache.thrift.TException;  
public class AdditionServiceHandler implements AdditionService.Iface {  
  
    @Override  
    public int add(int n1, int n2) throws TException {  
  
        return n1 + n2; }  
}
```

Writing a simple server

Following is an example code to initiate a simple thrift server. To enable the multithreaded server uncomment the commented parts of the example code.

Example server (MyServer.java)

```
package com.eviac.blog.samples.thrift.server; import  
org.apache.thrift.transport.TServerSocket;
```

```

import org.apache.thrift.transport.TServerTransport;
import org.apache.thrift.server.TServer;
import org.apache.thrift.server.TServer.Args;
import org.apache.thrift.server.TSimpleServer;

public class MyServer {

    public static void StartsimpleServer(AdditionService.Processor<AdditionServiceHandler> processor) { try
    {
        TServerTransport serverTransport = new TServerSocket(9090);
        TServer server = new TSimpleServer(
            new Args(serverTransport).processor(processor));

        // Use this for a multithreaded server
        // TServer server = new TThreadPoolServer(new
        //   TThreadPoolServer.Args(serverTransport).processor(processor));

        System.out.println("Starting the simple server...");
        server.serve();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    StartsimpleServer(new AdditionService.Processor<AdditionServiceHandler>(new
    AdditionServiceHandler()));
}
}

```

Writing the client

Following is an example java client code which consumes the service provided by AdditionService.

Example client code (AdditionClient.java)

```

package com.eviac.blog.samples.thrift.client;
import org.apache.thrift.TException;
import org.apache.thrift.protocol.TBinaryProtocol;
import org.apache.thrift.protocol.TProtocol; import
org.apache.thrift.transport.TSocket;

```

```
import org.apache.thrift.transport.TTransport;
import org.apache.thrift.transport.TTransportException;
public class AdditionClient {
public static void main(String[] args) {
try {
TTransport transport;

transport = new TSocket("localhost", 9090);
transport.open();

TProtocol protocol = new TBinaryProtocol(transport);
AdditionService.Client client = new AdditionService.Client(protocol);
System.out.println(client.add(100, 200));

transport.close();

} catch (TTransportException e) {
e.printStackTrace();
} catch (TException x) {
x.printStackTrace();
}
}

}

}
```

OUTPUT:

Run the server code(`MyServer.java`). It should output following and will listen to the requests.

Starting the simple server...

Then run the client code(`AdditionClient.java`). It should output following.

300

RESULT:

Thus, a procedure to installation of Thrift and programs are executed successfully.

EX.NO: 7.1

CASSANDRA

DATE:

AIM:

To Export and Import data in Cassandra

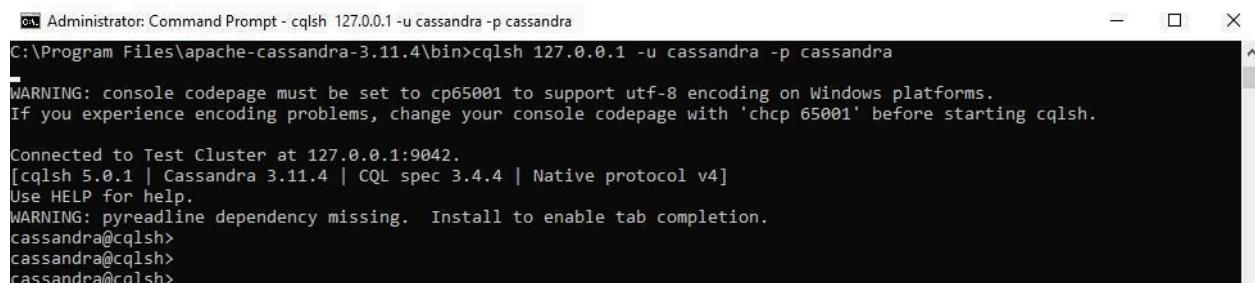
THEORY:

Cassandra is a distributed database management system which is open source with wide column store, NoSQL database to handle large amount of data across many commodity servers which provides high availability with no single point of failure. It is written in Java and developed by Apache Software Foundation.

CQL shell (cqlsh) :

This is a tool for [Cassandra Query Language](#) which supports Cassandra. cqlsh is a command-line shell for interacting with Cassandra through CQL (the Cassandra Query Language). with the help of the cql command, we can read and write data with the help of the cql query.

By default, CQL installed in bin/ directory alongside the Cassandra executable. In Cassandra, cqlsh utilizes the [Python](#) native protocol driver and connects to the single node specified on the command line.



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt - cqlsh 127.0.0.1 -u cassandra -p cassandra". The command entered was "cqlsh 127.0.0.1 -u cassandra -p cassandra". The output shows a warning about console codepage, connection to the Test Cluster at 127.0.0.1:9042, and the version of cqlsh and Cassandra. It also mentions a pyreadline dependency missing and shows three command-line prompts: "cassandra@cqlsh>", "cassandra@cqlsh>", and "cassandra@cqlsh>".

PROGRAM:

Step 1: CREATE TABLE

Create table namely as Data in which id, firstname, lastname are the fields for sample exercise.

Table name: Data

```
CREATE TABLE Data (
    id UUID PRIMARY KEY,
    firstname text,
    lastname text
);
```

Step 2: INSERT DATA

Insert some data to export and import data.

Example:

```
INSERT INTO Data (id, firstname, lastname )
VALUES (3b6441dd-3f90-4c93-8f61-abcfa3a510e1, 'Ashish', 'Rana');
```

```
INSERT INTO Data (id, firstname, lastname)
VALUES (3b6442dd-bc0d-4157-a80f-abcfa3a510e2, 'Amit', 'Gupta');
```

```
INSERT INTO Data (id, firstname, lastname)
VALUES (3b6443dd-d358-4d99-b900-abcfa3a510e3, 'Ashish', 'Gupta');
```

```
INSERT INTO Data (id, firstname, lastname)
VALUES (3b6444dd-4860-49d6-9a4b-abcfa3a510e4, 'Dhruv', 'Gupta');
```

```
INSERT INTO Data (id, firstname, lastname)
VALUES (3b6445dd-e68e-48d9-a5f8-abcfa3a510e5, 'Harsh', 'Vardhan');
```

```
INSERT INTO Data (id, firstname, lastname)
VALUES (3b6446dd-eb95-4bb4-8685-abcfa3a510e6, 'Shivang', 'Rana');
```

Step 3: EXPORT DATA

Export Data used the following cqlsh query given below.

```
cqlsh>COPY Data(id, firstname, lastname)
TO 'AshishRana\Desktop\Data.csv' WITH HEADER = TRUE;
```

OUTPUT:

The CSV file is created:

Using 7 child processes

Starting copy of Data with columns [id, firstname, lastname].

Processed: 6 rows; Rate: 20 rows/s; Avg. rate: 30 rows/s

6 rows exported to 1 files in 0.213 seconds.

Step 4: DELETE DATA

Delete data from table ‘Data’ to import again from CSV file which is already has been created.

```
cqlsh>truncate Data;
```

Step 5: IMPORT DATA

To import Data used the following cqlsh query given below.

```
cqlsh>COPY Data (id, firstname, lastname) FROM 'AshishRana\Desktop\Data.csv' WITH HEADER = TRUE;
```

OUTPUT:

Using 7 child processes

Starting copy of Data with columns [id, firstname, lastname].

Processed: 6 rows; Rate: 10 rows/s; Avg. rate: 14 rows/s

6 rows imported from 1 files in 0.423 seconds (0 skipped).

Step 6: RETRIEVE IMPORTED DATA

To View the results whether it is successfully imported or not.

cqlsh>SELECT * FROM Data;

OUTPUT:

id	firstname	lastname
3b6446dd-eb95-4bb4-8685-abcdefa3a510e6	Shivang	Rana
3b6444dd-4860-49d6-9a4b-abcdefa3a510e4	Dhruv	Gupta
3b6445dd-e68e-48d9-a5f8-abcdefa3a510e5	Harsh	Vardhan
3b6441dd-3f90-4c93-8f61-abcdefa3a510e1	Ashish	Rana
3b6442dd-bc0d-4157-a80f-abcdefa3a510e2	Amit	Gupta
3b6443dd-d358-4d99-b900-abcdefa3a510e3	Ashish	Gupta

RESULT:

Thus, a procedure to installation of Cassandra was successfully completed and execution of files are exported and imported successfully.

EX.NO: 7.2

MongoDB

DATE:

AIM:

To Export and Import data in MongoDB

THEORY:

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database.

MongoDB is a cross-platform, document-oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

Database

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

Collection

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

Document

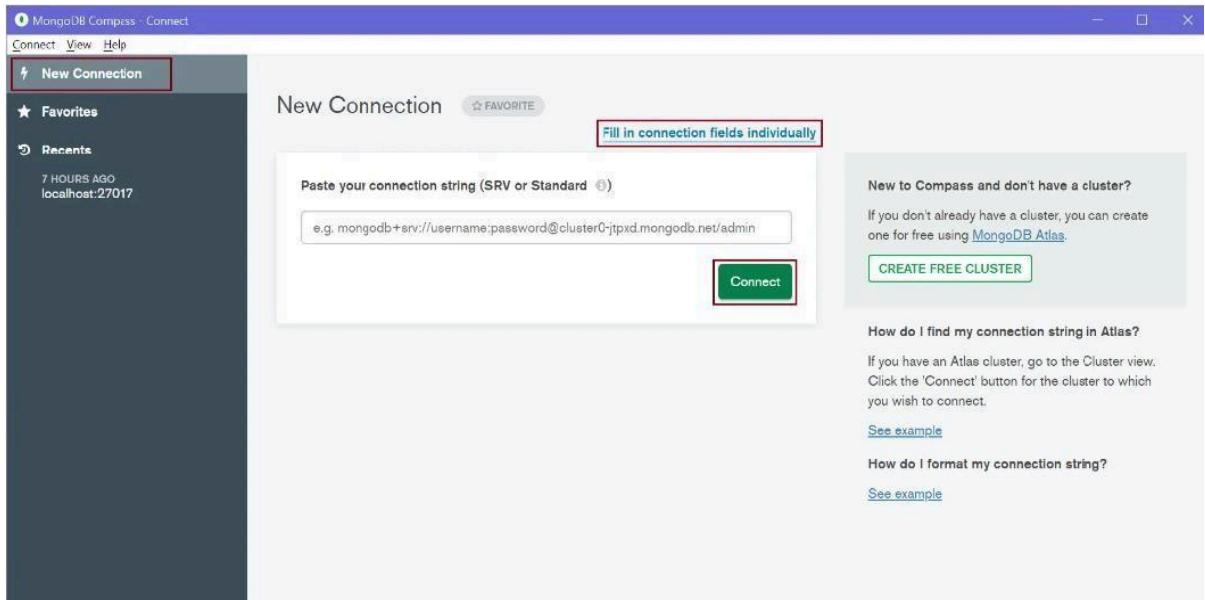
A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

PROCEDURE:

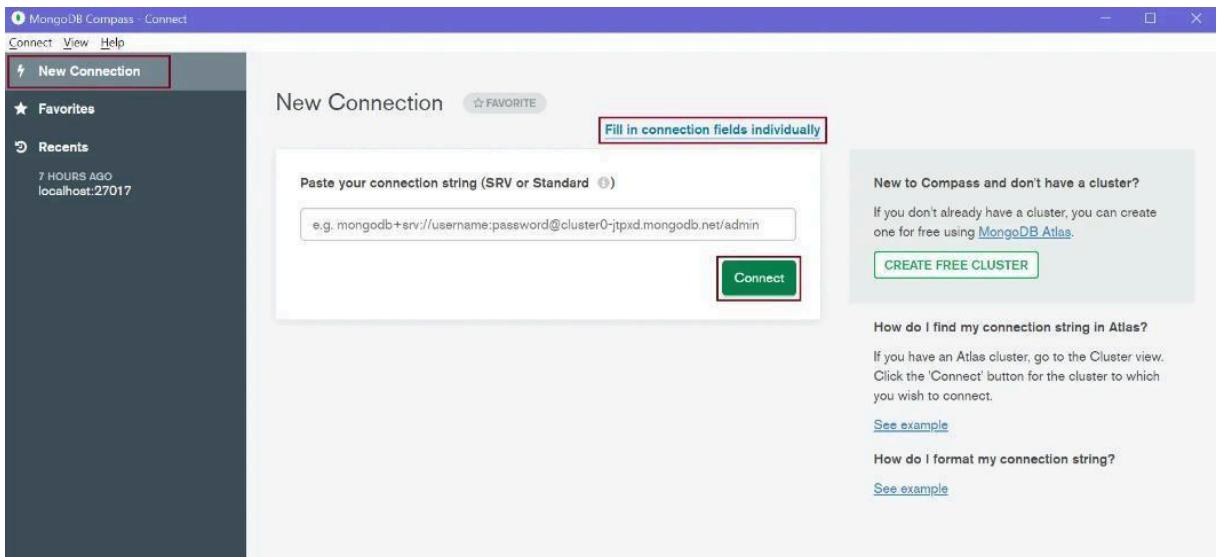
Step 1:

Install MongoDB Compass On Windows

MongoDB Compass is a GUI based tools (unlike MongoDB Shell) to interact with local or remote MongoDB server and databases. Use Compass to visually explore your data, run ad hoc queries, perform CRUD operations, and view and optimize your query performance. It can be installed on Linux, Mac, or Windows.



just click OK to connect with your local server, as shown below.



As you can see above, it will display all the databases on the connected MongoDB server. On the left pane, it displays information about the connected server.

Now, you can create, modify, delete databases, collections, documents using MongoDB Compass.

Click on the CREATE DATABASE button to create a new database. This will open Create Database popup, as shown below.

Create Database

Database Name

humanResourceDB

Collection Name

employees

Capped Collection

Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. [ⓘ](#)

Use Custom Collation

Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. [ⓘ](#)

Time-Series

Time-series collections efficiently store sequences of measurements over a period of time.

Cancel

Create Database

Enter your database name and collection name and click Create Database. This will create a new database humanResourceDB with the new employees collection shown below.

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays the connection details: HOST localhost:27017, CLUSTER Standalone, and EDITION MongoDB 5.0.3 Community. Under the 'humanResourceDB' section, the 'employees' collection is selected. The main pane, titled 'Collections', lists the 'employees' collection with 0 documents, 0.0 B total size, 1 index, and 4.0 KB total index size. A green 'CREATE COLLECTION' button is visible at the top of the list.

Click on employees collection to insert, update, find documents in it. This will open the following window to manage documents.

The screenshot shows the 'Documents' view for the 'employees' collection. The top navigation bar includes 'Local', '3 DBS', '1 COLLECTIONS', and 'humanResourceDB.employees'. The main area shows the collection name 'humanResourceDB.employees' with 0 documents, 0 B total size, and 1 index, both with 4.0KB total size. Below this, there are tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. A red arrow points to the 'Validation' tab. The 'Documents' tab has a 'FILTER' field containing '{ Field: "value" }', and buttons for 'OPTIONS', 'FIND', 'RESET', and '...'. It also displays 'Displaying documents 0 - 0 of N/A' and 'REFRESH'. A large green 'ADD DATA' button is at the bottom left. In the center, there is a message: 'This collection has no data' with a small icon of two overlapping documents below it. At the bottom right is a green 'Import Data' button.

PROGRAMS:

1. Create a new database:

Command: use <database-name>

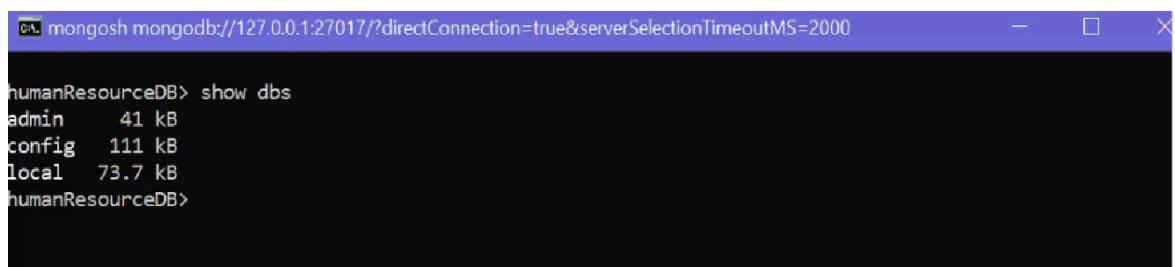
Example: use humanResourceDB

2. Check databases

list: Command: show dbs

Example: show dbs

OUTPUT:



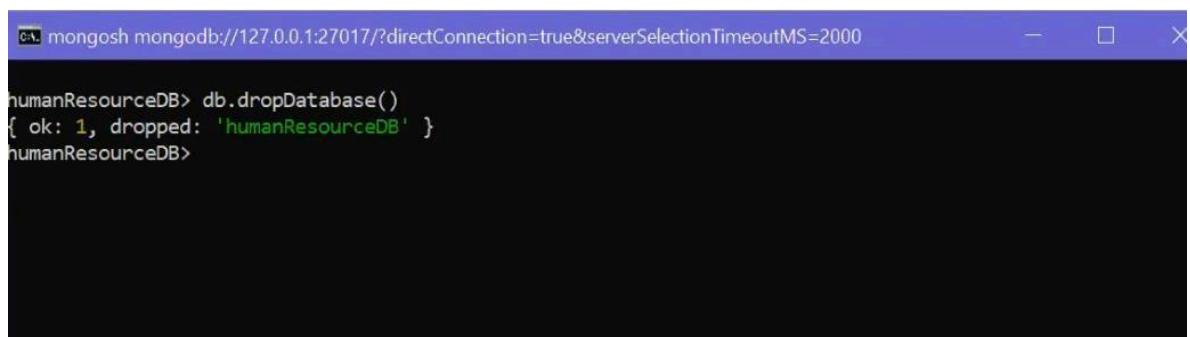
```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
humanResourceDB> show dbs
admin      41 kB
config     111 kB
local      73.7 kB
humanResourceDB>
```

3. Delete a database:

Command: db.dropDatabase()

Example: db.dropDatabase()

OUTPUT:



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
humanResourceDB> db.dropDatabase()
{ ok: 1, dropped: 'humanResourceDB' }
humanResourceDB>
```

4. Create a collection:

Command: db.createCollection()

Example: db.createCollection("employee")

OUTPUT:

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
humanResourceDB> db.createCollection("employees")
{ ok: 1 }
humanResourceDB>
```

Creates multiple collections.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
humanResourceDB> db.createCollection("departments")
{ ok: 1 }
humanResourceDB> db.createCollection("addresses")
{ ok: 1 }
humanResourceDB>
```

To show Db:

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
humanResourceDB> show collections
addresses
departments
employees
humanResourceDB>
```

To delete a collection, use the db.<collection-name>.drop() method

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
humanResourceDB> db.addresses.drop() ←
true
humanResourceDB> show collections ←
departments
employees
humanResourceDB>
```

5. Insert documents into a collection:

5.1 insertOne() - Inserts a single document into a collection. Command: db.<collection>.insertOne()

Example: db.employees.insertOne({
 firstName: "John",
 lastName: "King",
 email: "john.king@abc.com"
})

OUTPUT:

```
{  
    acknowledged: true,  
    insertedId: ObjectId("616d44bea861820797edd9b0")  
}
```

5.2 insert() - Inserts one or more documents into a collection. Command: db.<collection>.insert()

Example: db.employees.insert(

```
[  
    {  
        firstName: "John",  
        lastName: "King",  
        email: "john.king@abc.com"  
    },  
    {  
        firstName: "Sachin",  
        lastName: "T",  
        email: "sachin.t@abc.com"  
    }]
```

OUTPUT:

```
{  
    acknowledged: true,  
    insertedIds: {  
        '0': ObjectId("616d63eda861820797edd9b3"),  
        '1': 1,  
        '2': ObjectId("616d63eda861820797edd9b5")  
    }  
}
```

5.3 insertMany() - Insert multiple documents into a collection. Commands: db.<collection>.insertMany()

Example 1:

```
db.employees.insertMany(  
[  
    {  
        firstName: "John",  
        lastName: "King",  
        email: "john.king@abc.com"  
    },  
    {  
        firstName: "Sachin",  
        lastName: "T",  
        email: "sachin.t@abc.com"  
    },  
    {  
        firstName: "James",  
        lastName: "Bond",  
        email: "jamesb@abc.com"  
    },  
])
```

Example 2: insertMany() with Custom _id

```
db.employees.insertMany([
```

```
_id:1,  
firstName: "John",  
lastName: "King",  
email: "john.king@abc.com",  
salary: 5000  
},  
{  
_id:2,  
firstName: "Sachin",  
lastName: "T",  
email: "sachin.t@abc.com",  
salary: 8000  
},  
{  
_id:3,  
firstName: "James",  
lastName: "Bond",  
email: "jamesb@abc.com",  
salary: 7500  
},  
{  
_id:4,  
firstName: "Steve",  
lastName: "J",  
email: "steve.j@abc.com",  
salary: 9000  
},  
{  
_id:5,  
firstName: "Kapil",  
lastName: "D",  
email: "kapil.d@abc.com",  
salary: 4500  
},
```

```
_id:6,  
firstName: "Amitabh",  
lastName: "B",  
email: "amitabh.b@abc.com",  
salary: 11000  
}  
])
```

6. Find the data:

1. findOne() - returns a the first document that matched with the specified criteria.
2. find() - returns a cursor to the selected documents that matched with the specified criteria.

Command: find()

Example : db.employees.find().pretty()

OUTPUT:

```
{ _id: ObjectId("616d44bea861820797edd9b0"),  
firstName: "John",  
lastName: "King",
```

```
email: "john.king@abc.com"  
}
```

Example: db.employees.findOne({firstName: "Kapil"})

OUTPUT:

```
{  
_id: 5,  
firstName: 'Kapil',  
lastName: 'D',  
email: 'kapil.d@abc.com',  
salary: 4500  
}
```

6.1 Find Multiple Documents:

Example: db.employees.find({salary: 7000})

OUTPUT:

```
[{
    _id:4,
    firstName: "Steve",
    lastName: "J",
    email: "steve.j@abc.com",
    salary: 7000
},
{
    _id:6,
    firstName: "Amitabh",
    lastName: "B",
    email: "amitabh.b@abc.com",
    salary: 7000
}
]
```

7. Import Data: Import data into a collection from either a **JSON** or **CSV** file

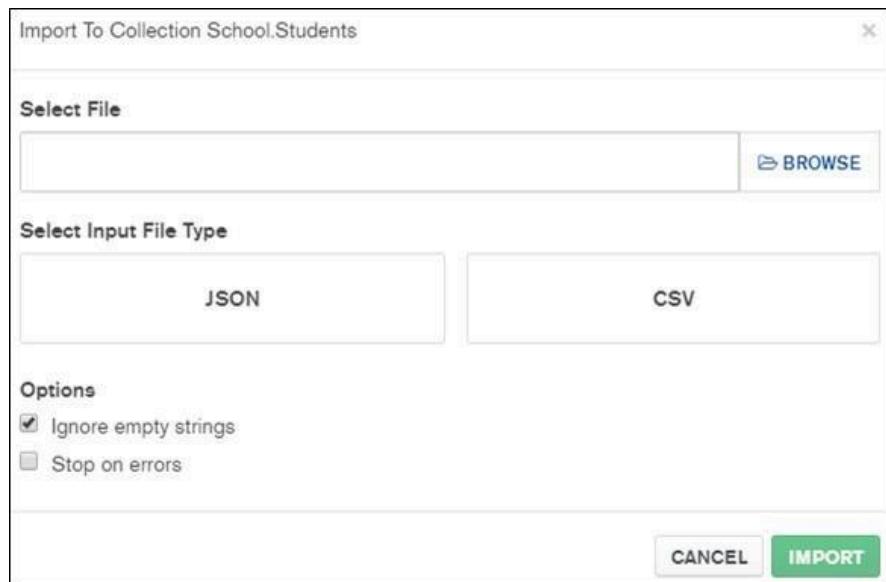
Step 1: Navigate to your target collection: Either select the collection from the Collections tab or click the collection in the left-hand pane.

Step 2: Click the Add Data dropdown and select Import JSON or CSV file.

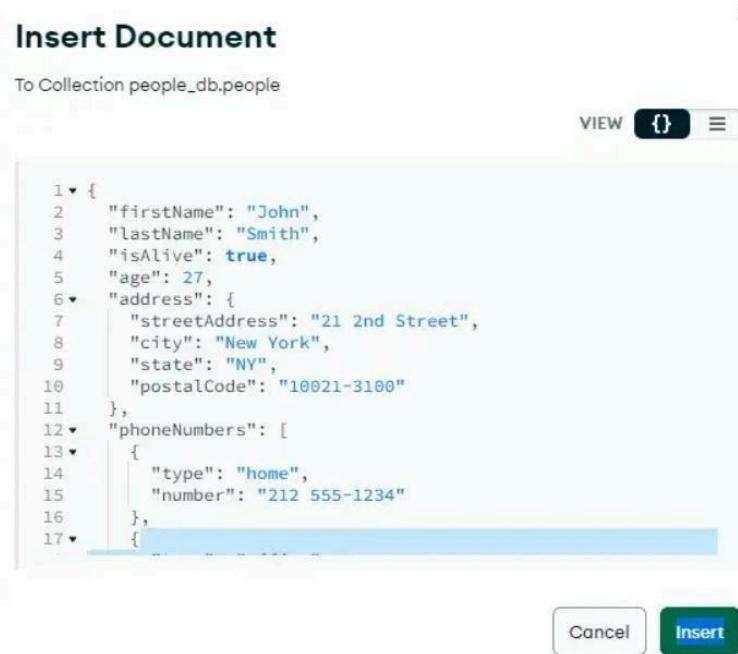
The screenshot shows the MongoDB Compass interface connected to a cluster named 'iot-cluster.zvpcdgg.mongodb.net'. The current database is 'people_db' and the collection is 'people'. The 'Documents' tab is selected. On the left, there's a sidebar with 'My Queries' and 'Databases' sections, and a search bar. The main area has a 'Filter' dropdown and a text input 'Type a query: { field: 'value' }'. Below the filter is a 'ADD DATA' button with options to 'Import JSON or CSV file' or 'Insert document'.

Step 3: Select the appropriate file type.

Select either a JSON or CSV file to import and click **Select**.

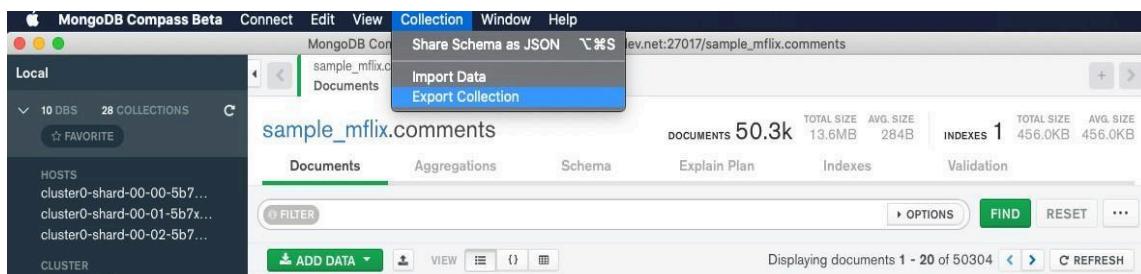


Step 4: Click insert or Import.



8. Export Data from a Collection: Export data from a collection as either a **JSON** or **CSV** file.

Step 1: Click the Export Data dropdown and select Export the full collection.



Step 2: Select your file type: You can select either **JSON** or **CSV**.

Export Collection sample_mflix.comments X

Select Export File Type

JSON CSV

Output

BROWSE

< BACK CANCEL EXPORT

Step 3: Click Export and Choose where to export the file and click Select.

RESULT:

Thus, a procedure to installation of MongoDB was completed and execution of files are exported and imported successfully.