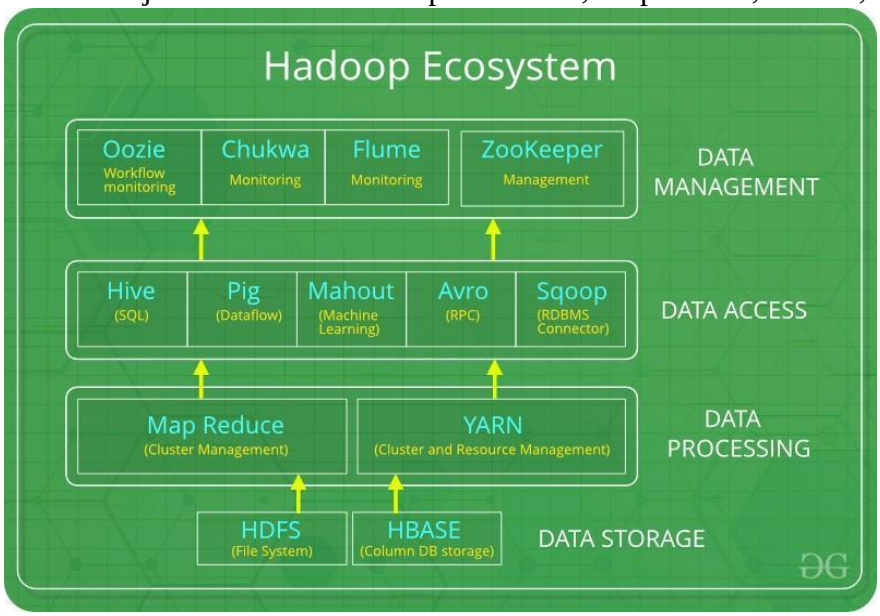
Experiment No.1
Hadoop HDFS Practical
Date of Performance:17/07/2023
Date of Submission:24/07/2023

AIM : Installation, Configuration of hadoop and performing basic file management operations in hadoop.

THEORY :

What is the Hadoop Ecosystem?

Hadoop Ecosystem is a platform or a suite which provides various services to solve the big data problems. It includes Apache projects and various commercial tools and solutions. There are four major elements of Hadoop i.e. HDFS, MapReduce, YARN, and Hadoop Common.



Following are the components that collectively form a Hadoop ecosystem:

- HDFS: Hadoop Distributed File System
- YARN: Yet Another Resource Negotiator
- MapReduce: Programming based Data Processing
- Spark: In-Memory data processing
- PIG, HIVE: Query based processing of data services
- HBase: NoSQL Database
- Mahout, Spark MLLib: Machine Learning algorithm libraries
- Solar, Lucene: Searching and Indexing
- Zookeeper: Managing cluster
- Oozie: Job Scheduling

HDFS:

HDFS is the primary or major component of Hadoop ecosystem and is responsible for storing large data sets of structured or unstructured data across various nodes and thereby maintaining the metadata in the form of log files.

HDFS consists of two core components i.e.

- Name node
- Data Node

Name Node is the prime node which contains metadata (data about data) requiring comparatively fewer resources than the data nodes that stores the actual data. These data nodes are commodity hardware in the distributed environment.

HDFS maintains all the coordination between the clusters and hardware.

YARN:

Yet Another Resource Negotiator, as the name implies, YARN is the one who helps to manage the resources across the clusters. In short, it performs scheduling and resource allocation for the Hadoop System.

Resource manager has the privilege of allocating resources for the applications in a system whereas Node managers work on the allocation of resources such as CPU, memory, bandwidth per machine and later on acknowledges the resource manager. Application manager works as an interface between the resource manager and node manager and performs negotiations as per the requirement of the two.

MapReduce:

MapReduce makes the use of two functions i.e. Map() and Reduce() whose task is:

Map() performs sorting and filtering of data and thereby organizing them in the form of group. Map generates a key-value pair based result which is later on processed by the Reduce() method.

Reduce(), as the name suggests does the summarization by aggregating the mapped data. In simple, Reduce() takes the output generated by Map() as input and combines those tuples into smaller set of tuples.

HIVE:

Hive is an ETL and Data warehousing tool used to query or analyze large datasets stored within the Hadoop ecosystem. Hive has three main functions: data summarization, query, and analysis of unstructured and semi-structured data in Hadoop. It features a SQL-like interface, HQL language that works similar to SQL and automatically translates queries into MapReduce jobs.

PIG:

Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL. It is a platform for structuring the data flow, processing and analyzing huge data sets. Pig does the work of executing commands and in the background, all the activities of MapReduce are taken care of. After the processing, pig stores the result in HDFS.

Apache Spark:

It's a platform that handles all the process consumptive tasks like batch processing, interactive or iterative real-time processing, graph conversions, and visualization, etc.

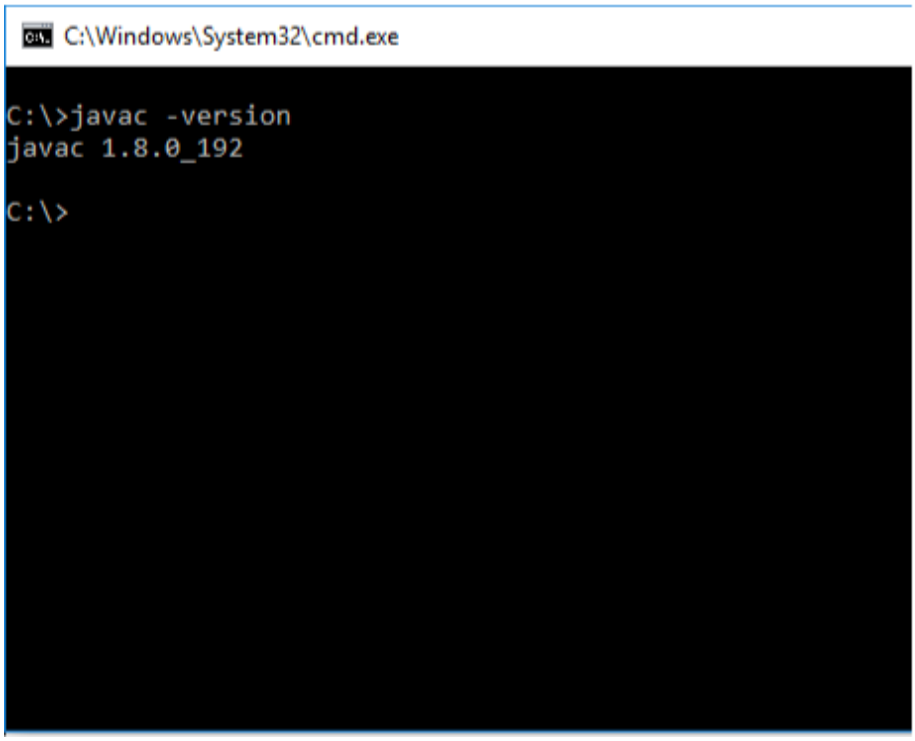
It consumes in memory resources hence, thus being faster than the prior in terms of optimization.

<u>Installation of Hadoop</u>

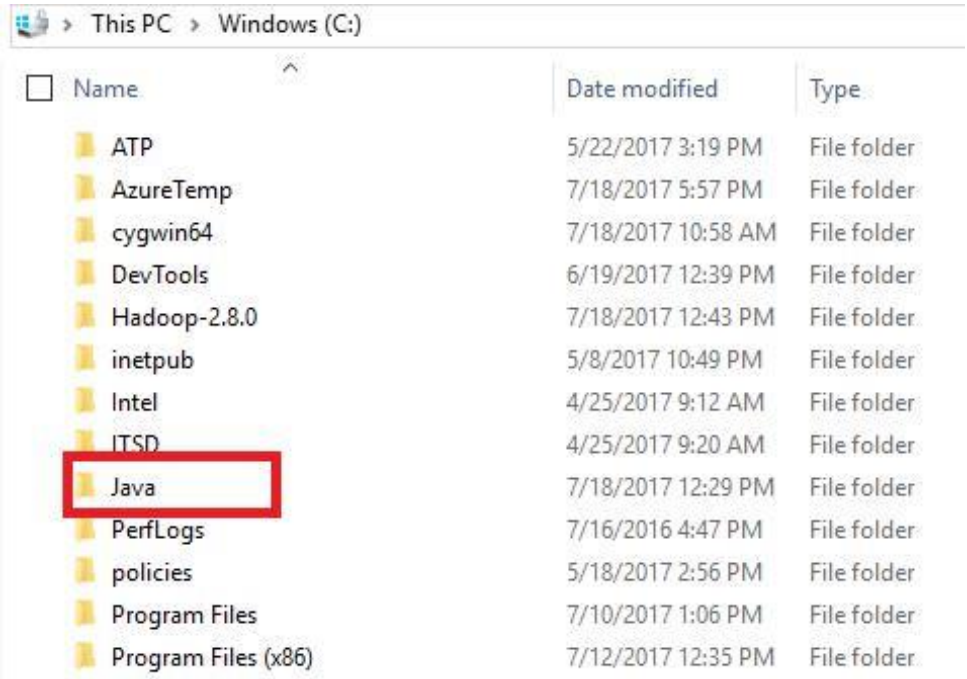
Download Hadoop 2.8.0 (Link: http://www-eu.apache.org/dist/hadoop/common/hadoop-2.8.0/hadoop-2.8.0.tar.gz OR http://archive.apache.org/dist/hadoop/core//hadoop-2.8.0/hadoop-2.8.0.tar.gz)

Java JDK 1.8.0.zip (Link: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)

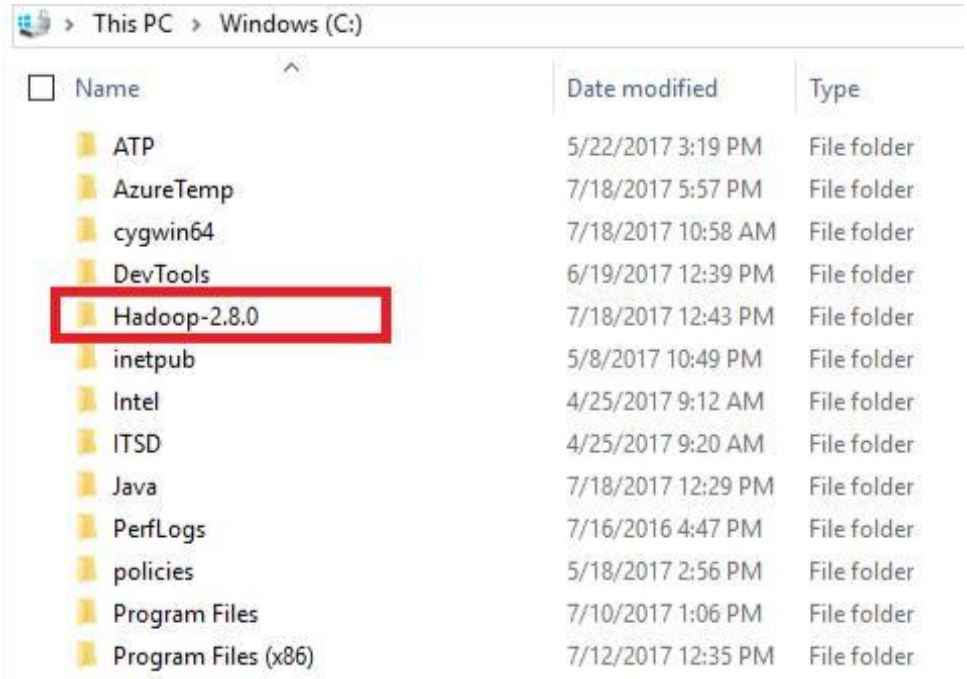
Check either Java 1.8.0 is already installed on your system or not, use "Javac -version" to check.



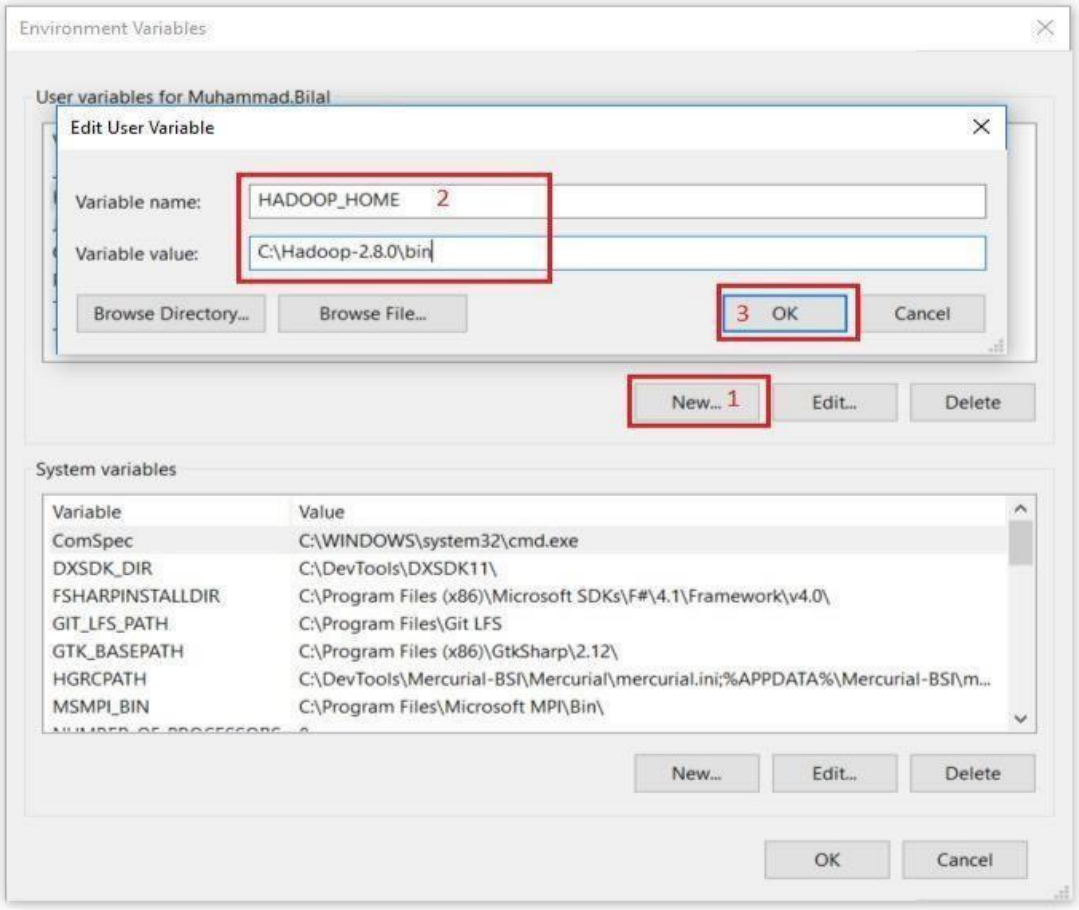
If Java is not installed on your system then first install java under "C:\JAVA"



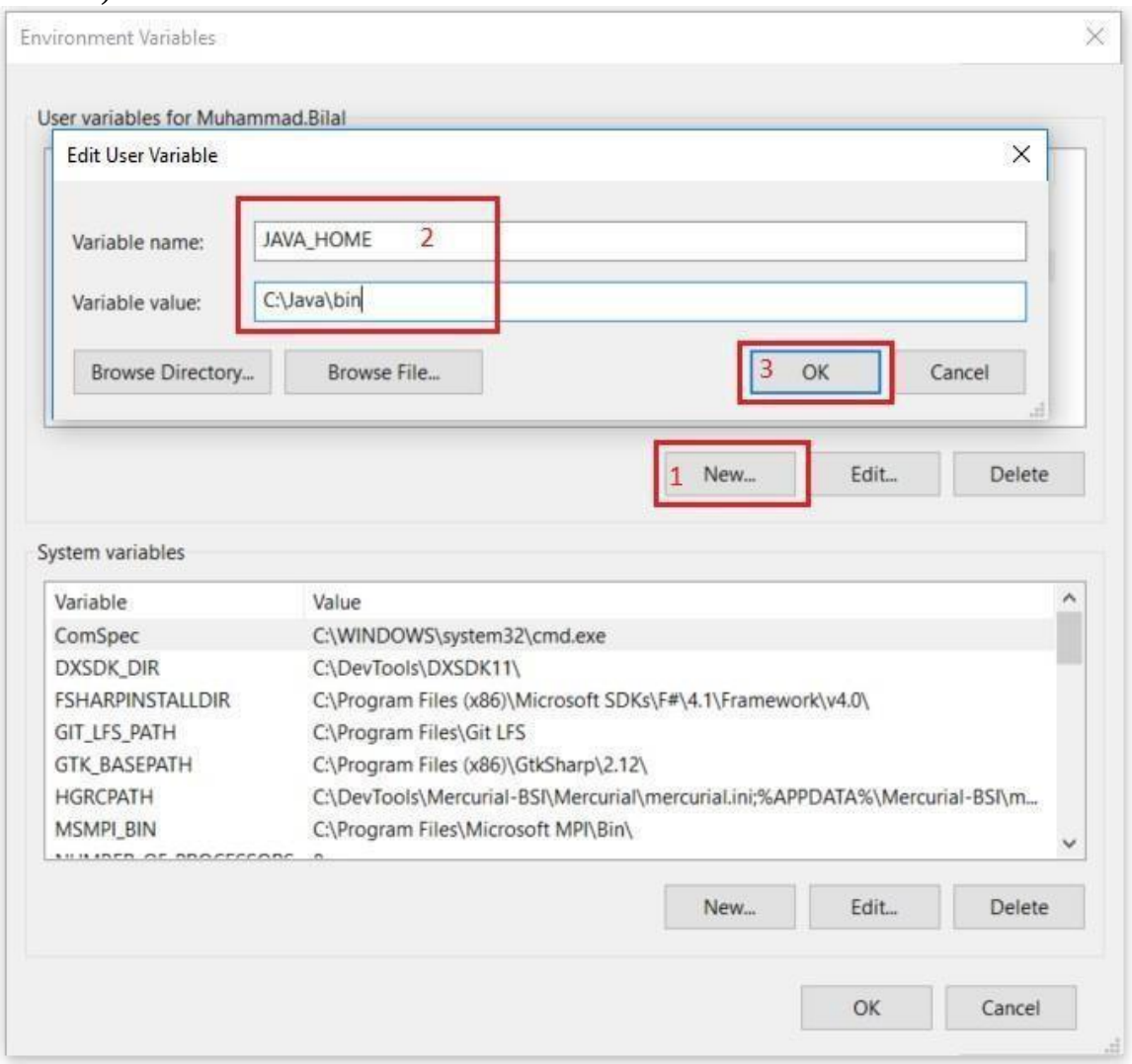
Extract file Hadoop 2.8.0.tar.gz or Hadoop-2.8.0.zip and place under "C:\Hadoop-2.8.0".



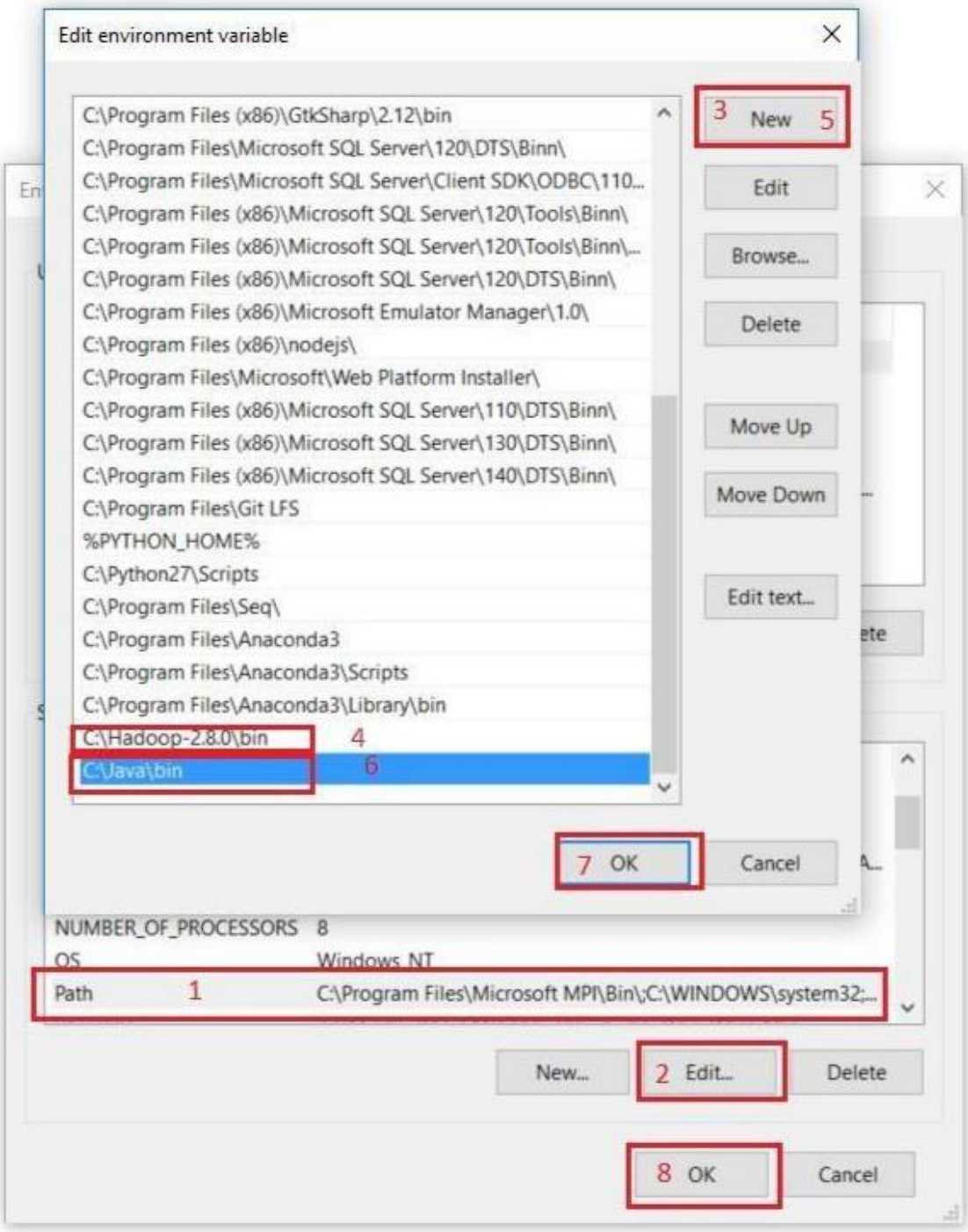
Set the path HADOOP_HOME Environment variable on windows 10(see Step 1,2,3 and 4 below).



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



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



CONFIGURATION :

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

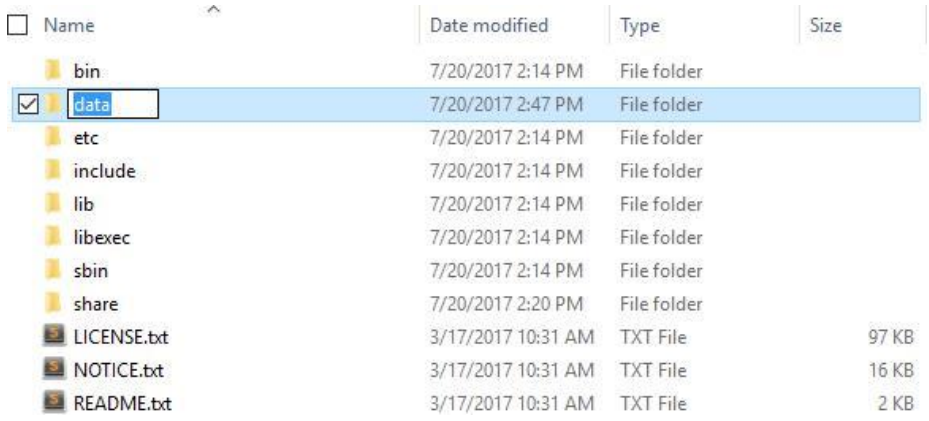
```
<configuration>
  <property>
    <name>fs.defaultFS</name>
```

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

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>
```

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"



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

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>
```

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\jdk\bin" (On C:\java this is path to file jdk.18.0)

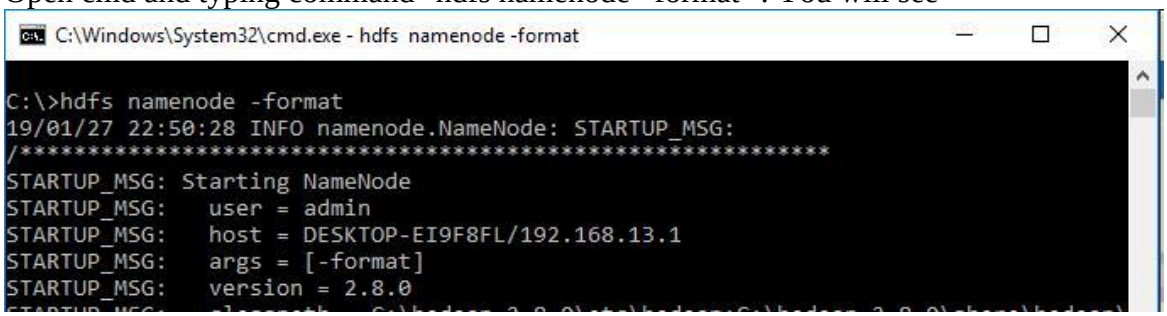
```
@rem The java implementation to use.  Required.
@rem set JAVA_HOME=%JAVA_HOME%
set JAVA_HOME=C:\java
```

HADOOP CONFIGURATION :

Dowload file Hadoop Configuration.zip (Link: https://github.com/MuhammadBilalYar/HADOOP-INSTALLATION-ON-WINDOW-10/blob/master/Hadoop%20Configuration.zip)

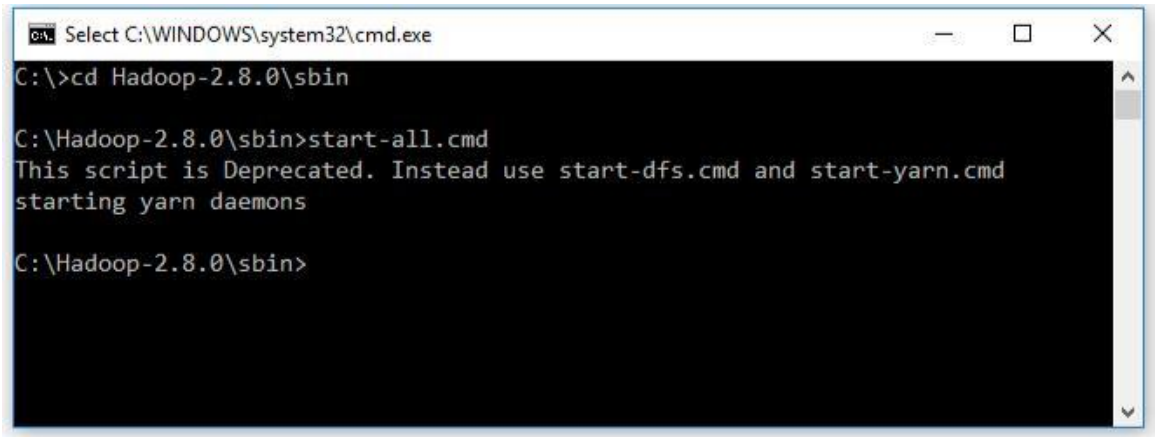
Delete file bin on C:\Hadoop-2.8.0\bin, replaced by file bin on file just download (from Hadoop Configuration.zip).

Open cmd and typing command "hdfs namenode –format" . You will see

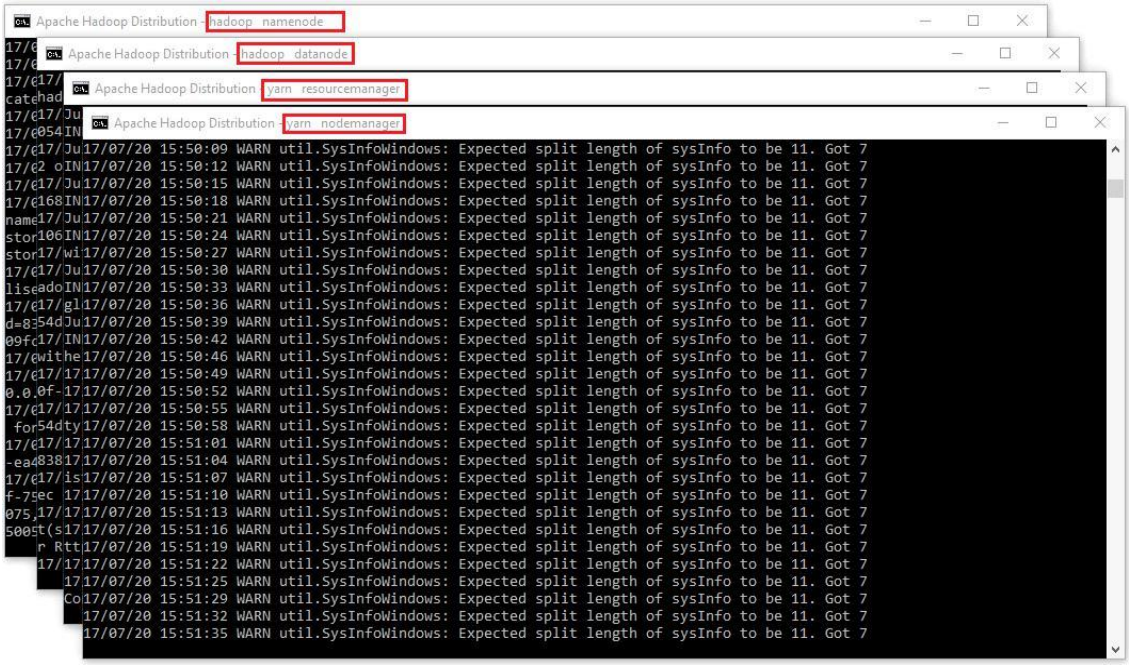


TESTING :

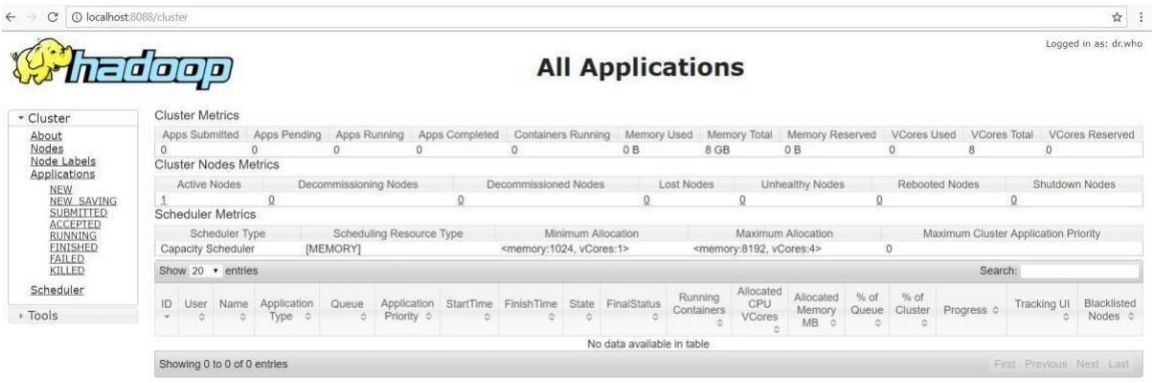
Open cmd and change directory to "C:\Hadoop-2.8.0\sbin" and type "start-all.cmd" to start apache.



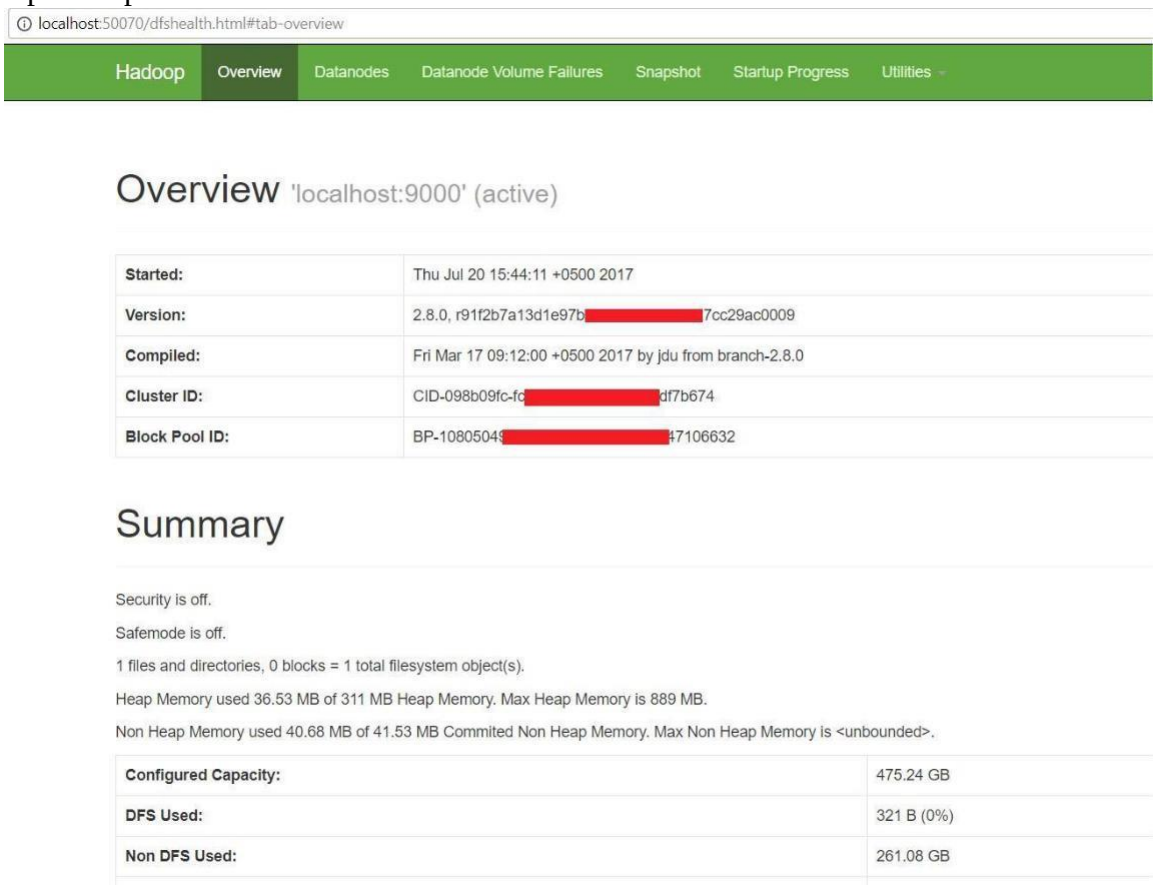
Make sure these apps are running :
Hadoop Namenode
Hadoop datanode
YARN Resourc Manager
YARN Node Manager



Open: http://localhost:8088



Open: http://localhost:50070



<u>File management tasks in hadoop</u>

In order to perform operations on Hadoop like copy, delete, move etc., following steps can be used:

Basic operations:

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

hadoop fs -mkdir <paths>

2. List the contents of a directory. Usage :
hadoop fs -ls <args>

3. See contents of a file
Same as unix cat command:
Usage:
hadoop fs -cat <path[filename]>
4. Copy a file from source to destination
This command allows multiple sources as well in which case the destination must be a directory.
Usage:
hadoop fs -cp <source> <dest>
5. Copy a file from/To Local file system to HDFS copyFromLocal
Usage:
hadoop fs -copyFromLocal <localsrc> URI
Similar to put command, except that the source is restricted to a local file reference. copyToLocal
Usage:
hadoop fs -copyToLocal [-ignorecrc] [-crc] URI <localdst>
Similar to get command, except that the destination is restricted to a local file reference.
7. Move file from source to destination.
Note:- Moving files across filesystem is not permitted.
Usage :
hadoop fs -mv <src> <dest>
8. Remove a file or directory in HDFS.
Remove files specified as argument. Deletes directory only when it is empty Usage :
hadoop fs -rm <arg>

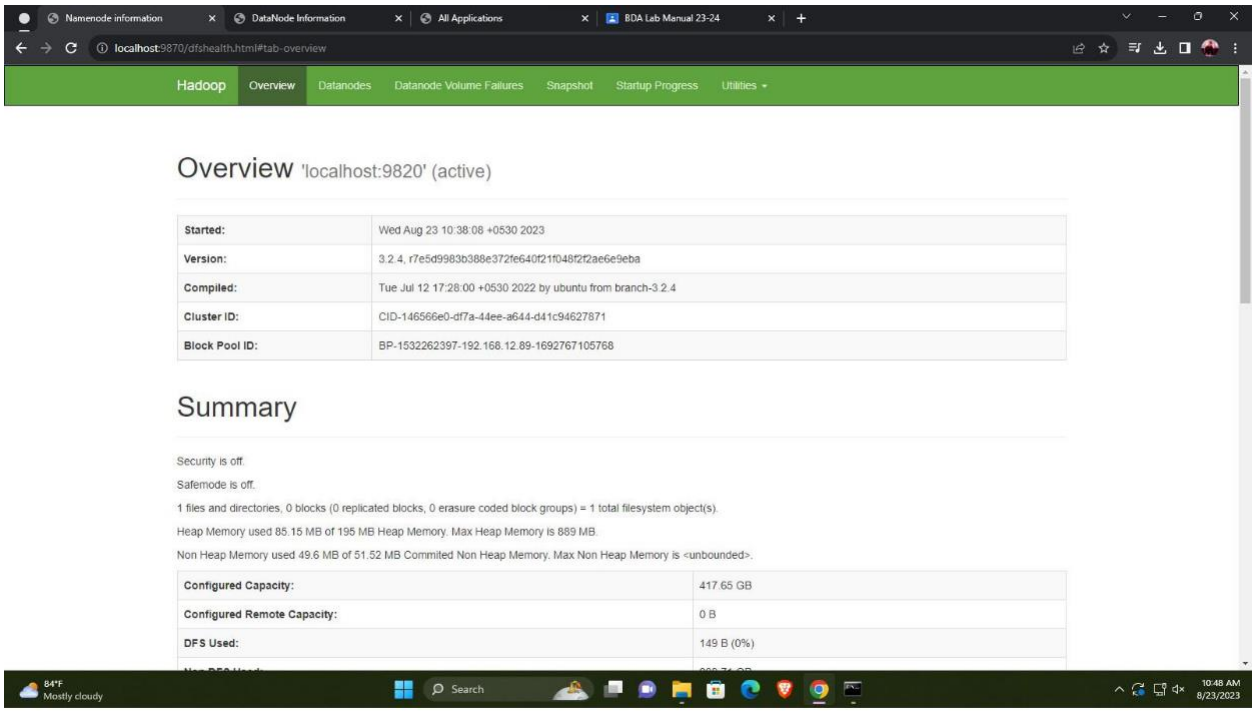
Steps for copying file

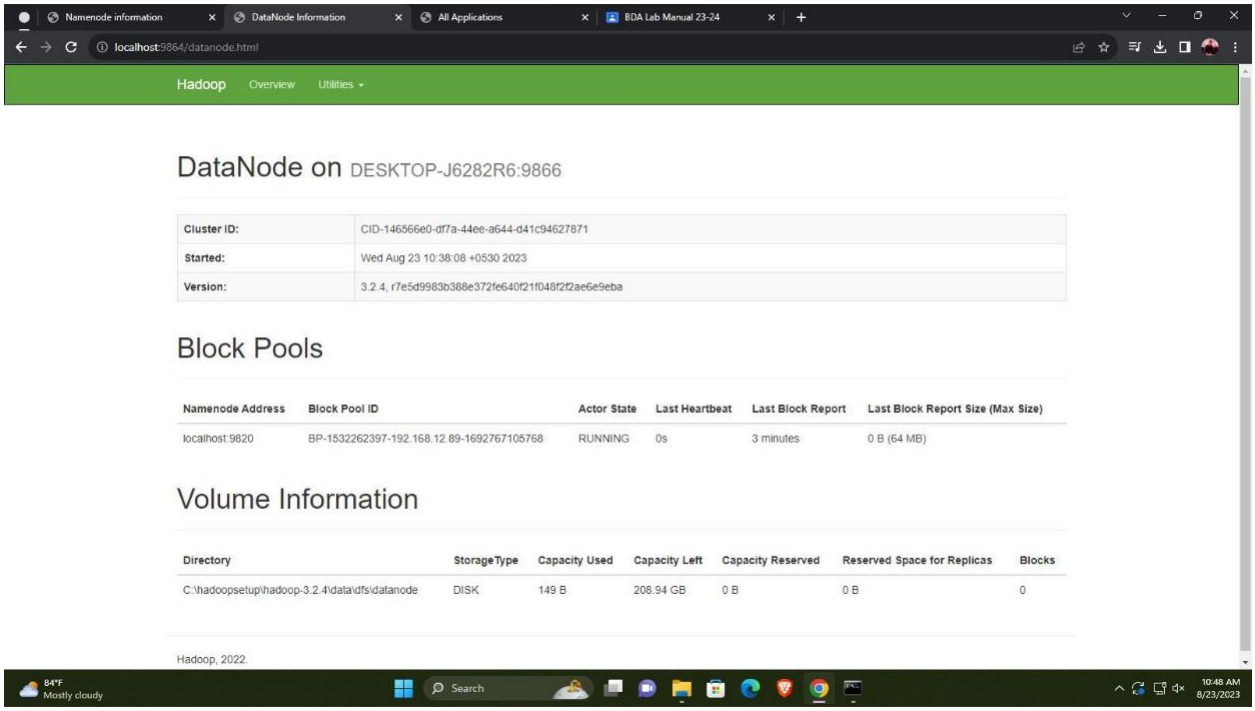
1) Go to Hadoop folder and then to sbin
C:\>cd C:\hadoop-2.8.0\sbin
2) Start namenode and datanode with this command, Two more cmd windows will open C:\hadoop-2.8.0\sbin>start-dfs.cmd
3) Now start yarn through following command, Two more windows will open, one for yarn resource manager and one for yarn node manager C:\hadoop-2.8.0\sbin>start-yarn.cmd
4) Create a directory named 'sample' in the hadoop directory using the following command C:\hadoop-2.8.0\sbin> hdfs dfs -mkdir /sample
5) To verify if the directory is created
C:\hadoop-2.8.0\sbin>hdfs dfs -ls /
6) Copy text file from D drive to sample C:\hadoop-2.8.0\sbin>hdfs dfs -copyFromLocal d:\rally.txt /sample

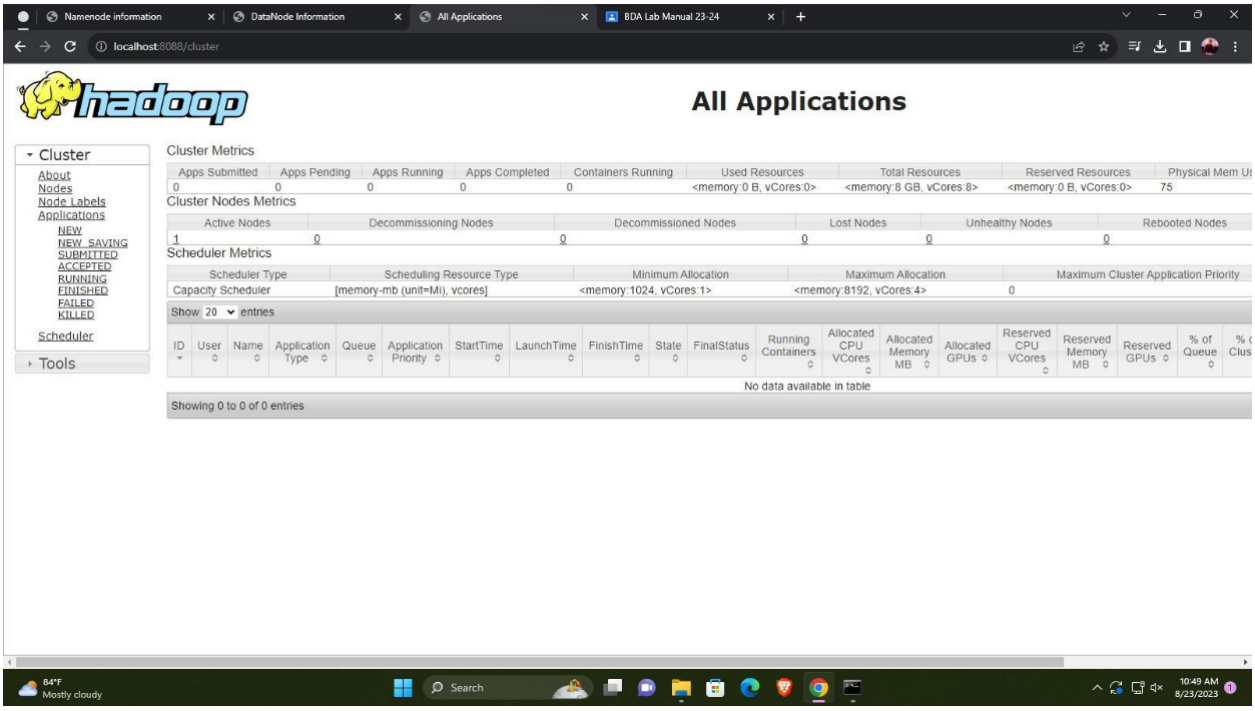
7) To verify if the file is copied

C:\hadoop-2.8.0\sbin>hdfs dfs -ls /sample

OUTPUT:







- C:\hadoopFiles\hadoop-3.2.4\sbin>hdfs dfs -mkdir /sample
- C:\hadoopFiles\hadoop-3.2.4\sbin>hdfs dfs -ls /
  Found 1 items
  drwxr-xr-x - admin supergroup 0 2023-08-23 10:58 /sample
- C:\hadoopFiles\hadoop-3.2.4\sbin>hdfs dfs -copyFromLocal C:\Users\admin\Desktop\hello.txt /sample
- C:\hadoopFiles\hadoop-3.2.4\sbin>hdfs dfs -ls
  /sample Found 1 items
  -rw-r--r-- 1 admin supergroup 12 2023-08-23 10:59 /sample/hello.txt

CONCLUSION:

The experiment showed how to install and set up Hadoop, a framework for processing large amounts of data across multiple computers. It successfully demonstrated the setup of Hadoop components like HDFS and MapReduce, which are scalable and fault-tolerant. The experiment also explored basic file management operations within Hadoop, such as uploading data, creating directories, and manipulating files. It showed the power of parallel processing with MapReduce, data replication, and data locality. While Hadoop has a learning curve, the experiment emphasized its potential for efficient data management and processing in large-scale applications, making it a valuable skillset in the era of big data.