Experiment No.1
Hadoop HDFS Practical
Date of Performance: 17/7/23
Date of Submission: 24/7/23

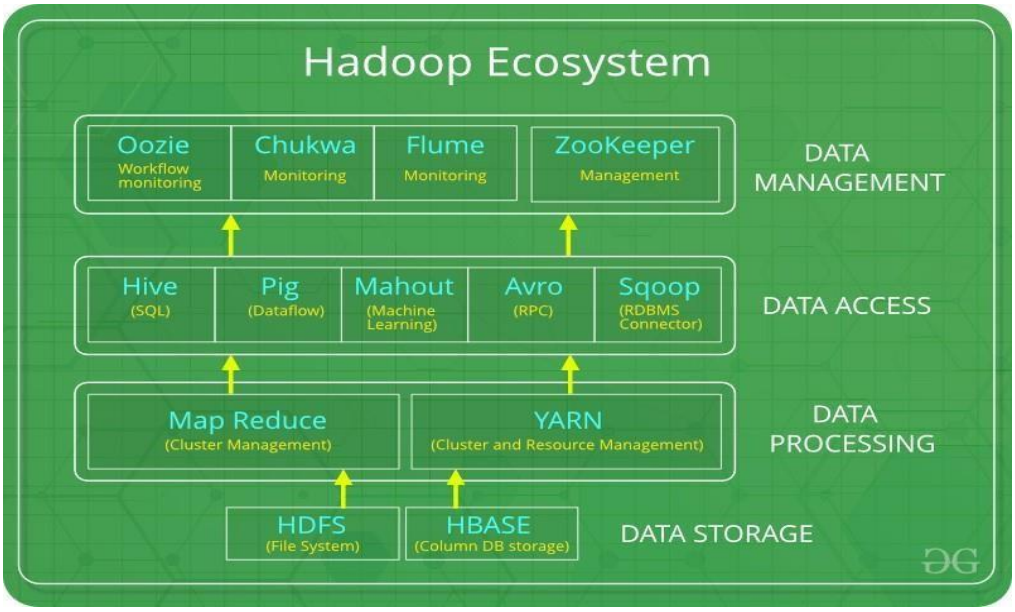
AIM:

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.

Installation of Hadoop:

Download Hadoop 2.8.0

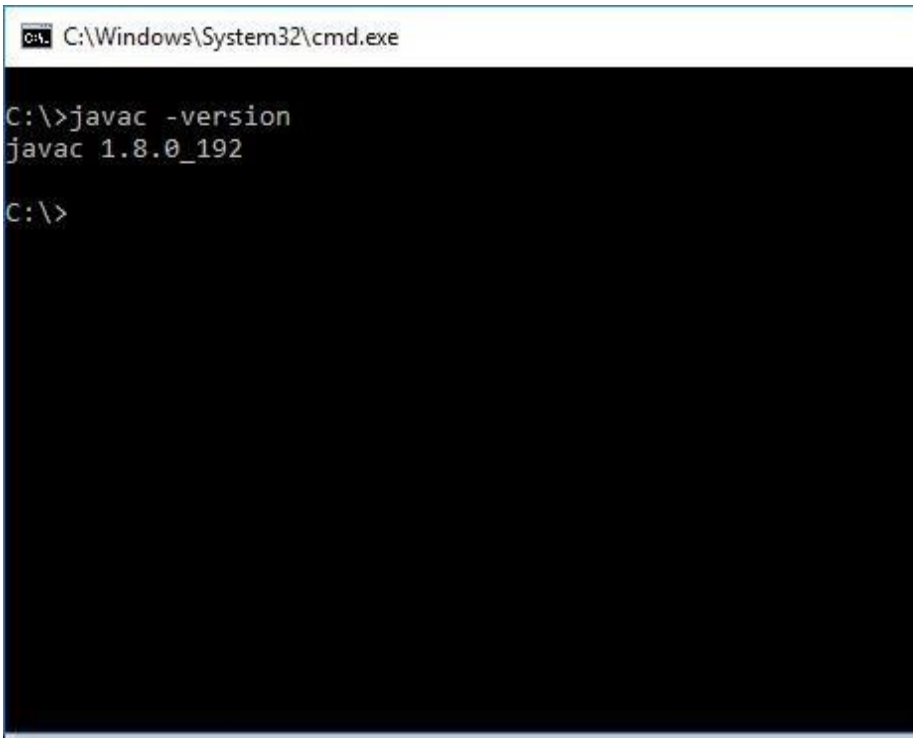
(Link: http://www-eu.apache.org/dist/hadoop/common/hadoop- 2.8.0/hadoop-2.8.0.tar.gz)

OR http://archive.apache.org/dist/hadoop/core//hadoop

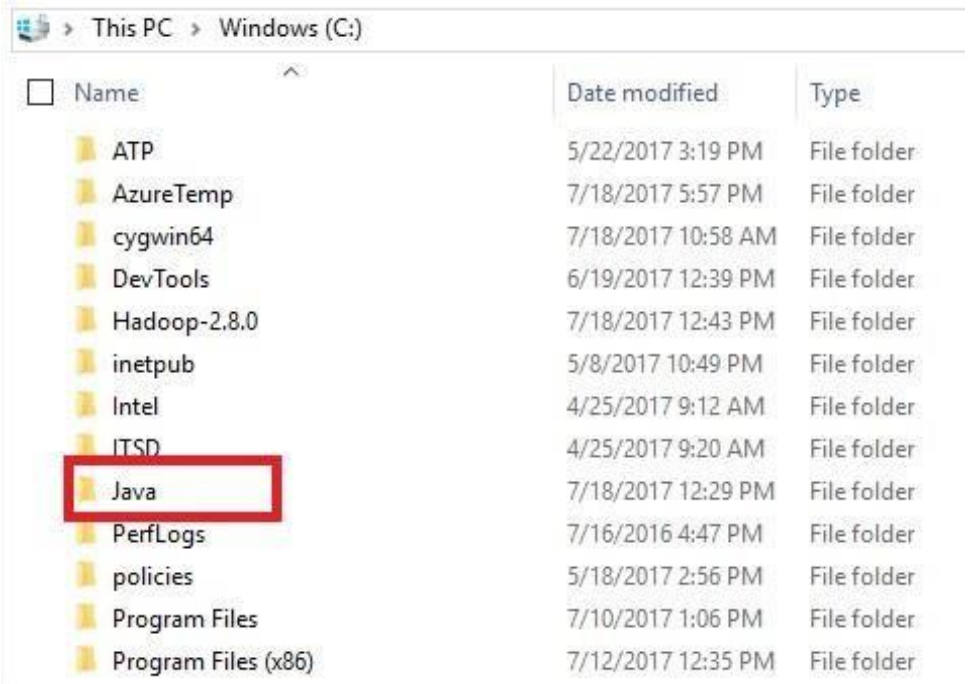
(- 2.8.0/hadoop-2.8.0.tar.gz)

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

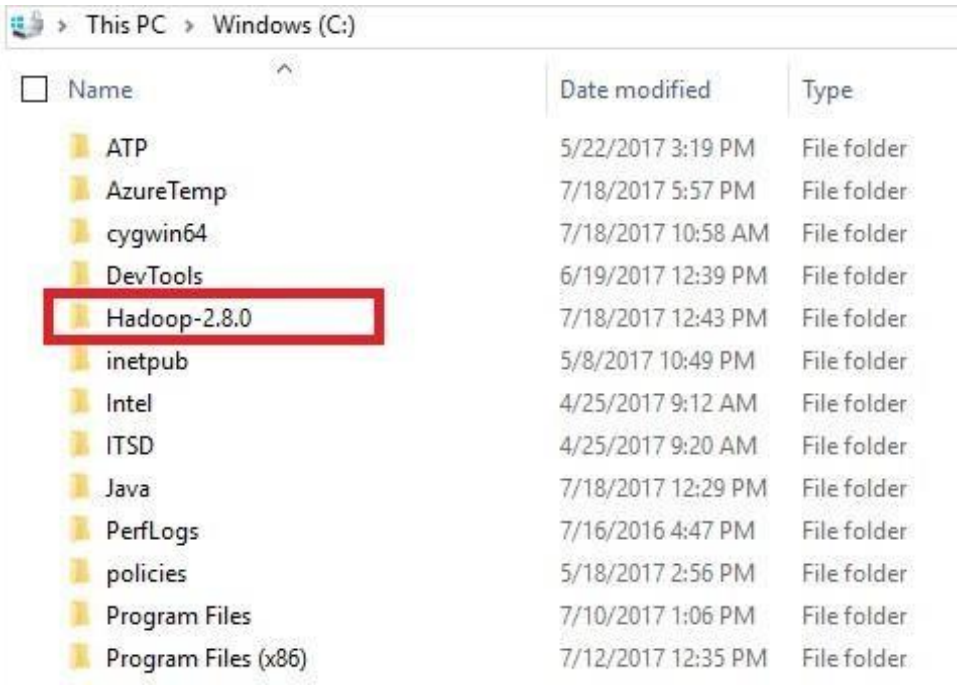
Check either Java 1.8.0 is already installed on your system or not, use "Javac -version" to check.



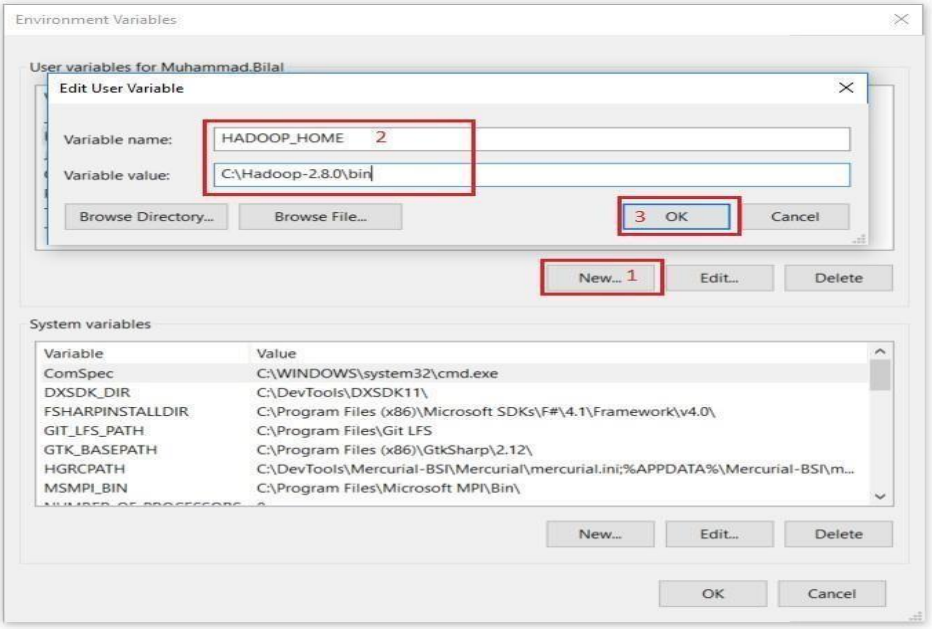
If Java is not installed on your system then first install java under "C:\JAVA"



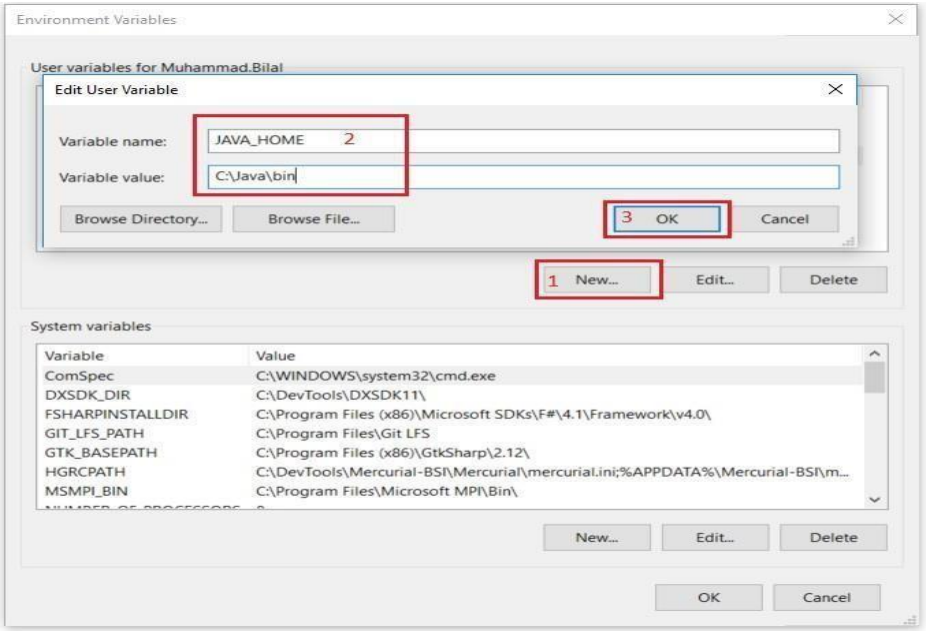
Extract file Hadoop 2.8.0.tar.gz or Hadoop-2.8.0.zip and place under "C:\Hadoop-2.8.0".



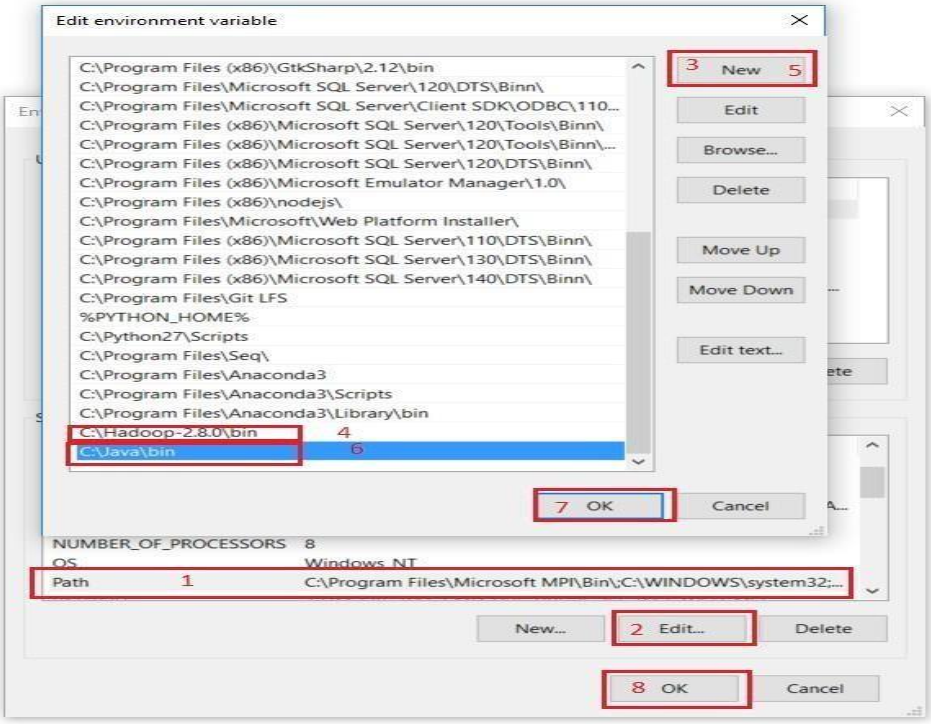
Set the path HADOOP_HOME Environment variable on windows 10(see Step 1,2,3 and 4 below).



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.

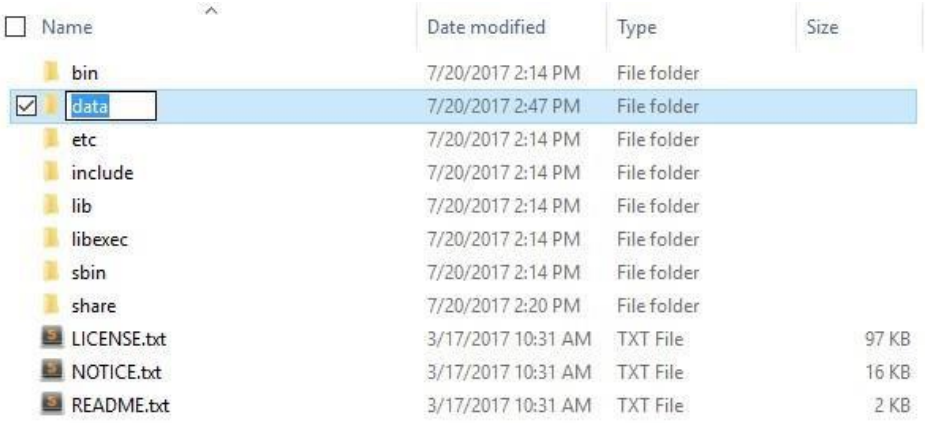
```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

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:

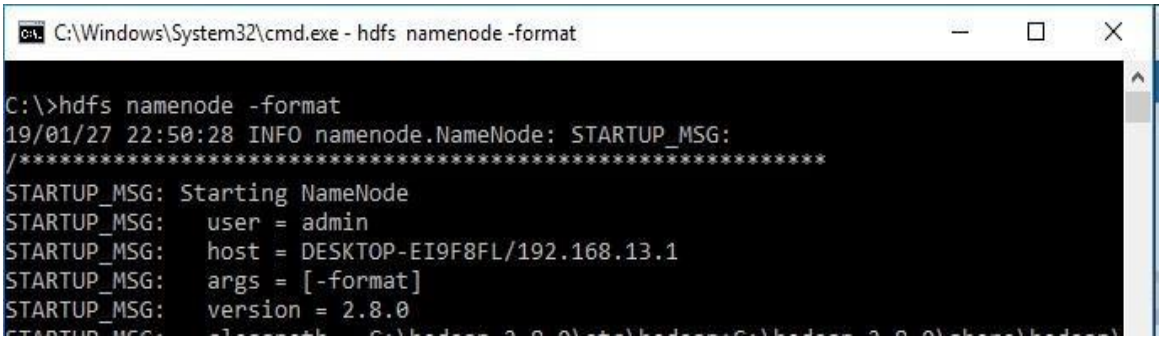
Dowload file Hadoop Configuration.zip

(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



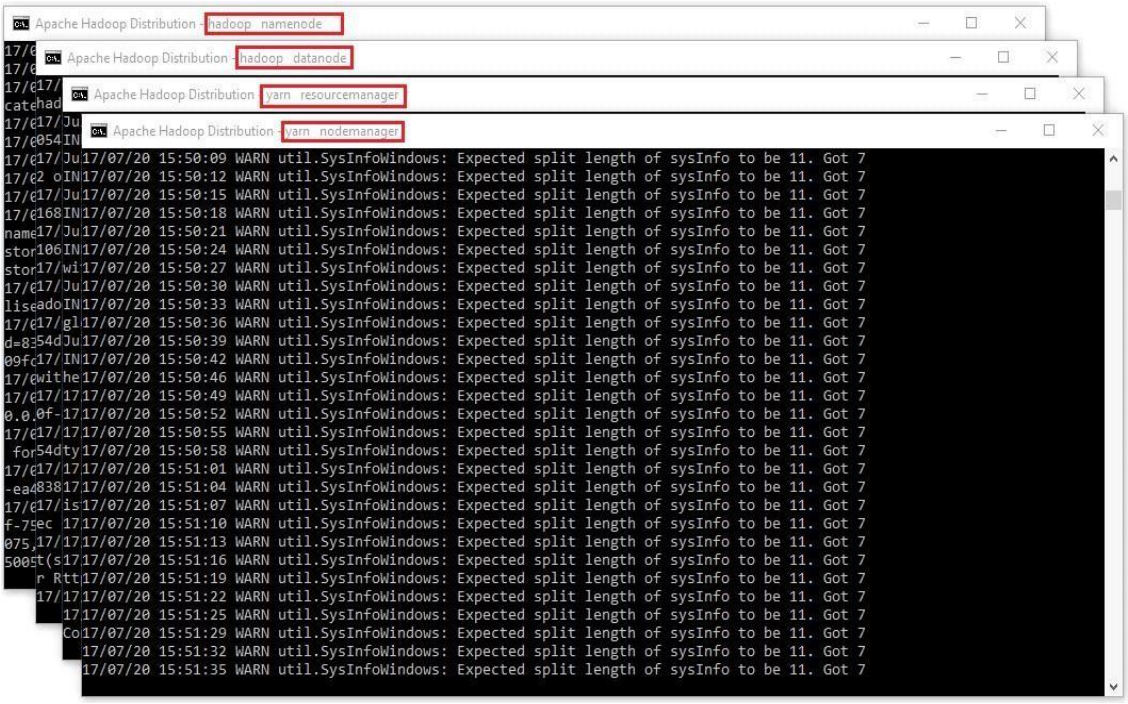
TESTING:

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

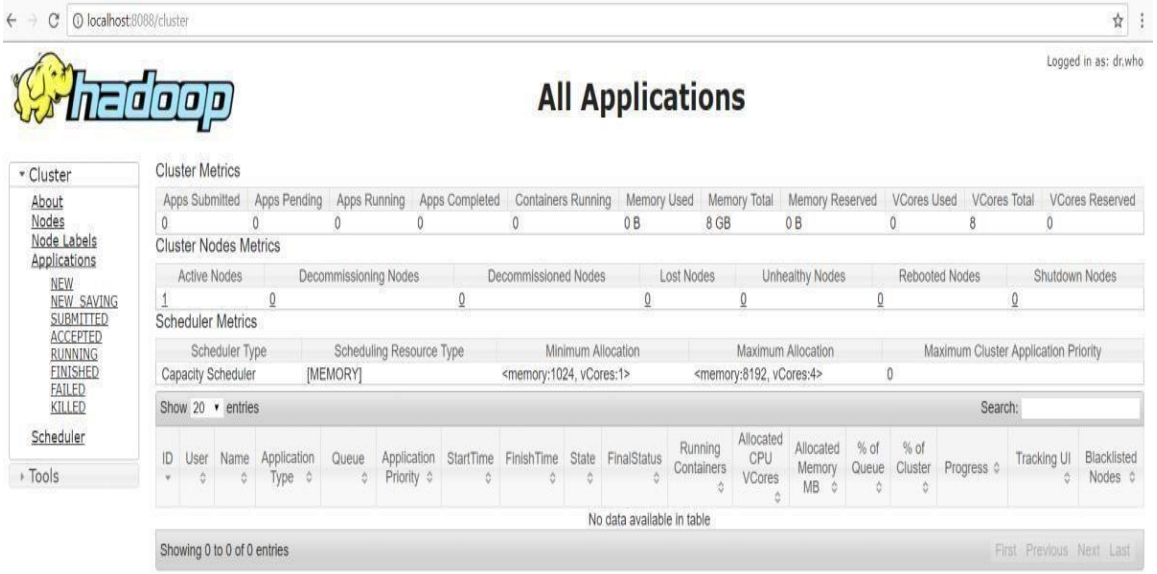
```
C:\hadoopsetup\hadoop-3.3.6\sbin> .\start-dfs.cmd

C:\hadoopsetup\hadoop-3.3.6\sbin> .\start-yarn.cmd
starting yarn daemons

C:\hadoopsetup\hadoop-3.3.6\sbin>jps
14736 DataNode
3588 NodeManager
3512 ResourceManager
8296 Jps
```

Make sure these apps are running:

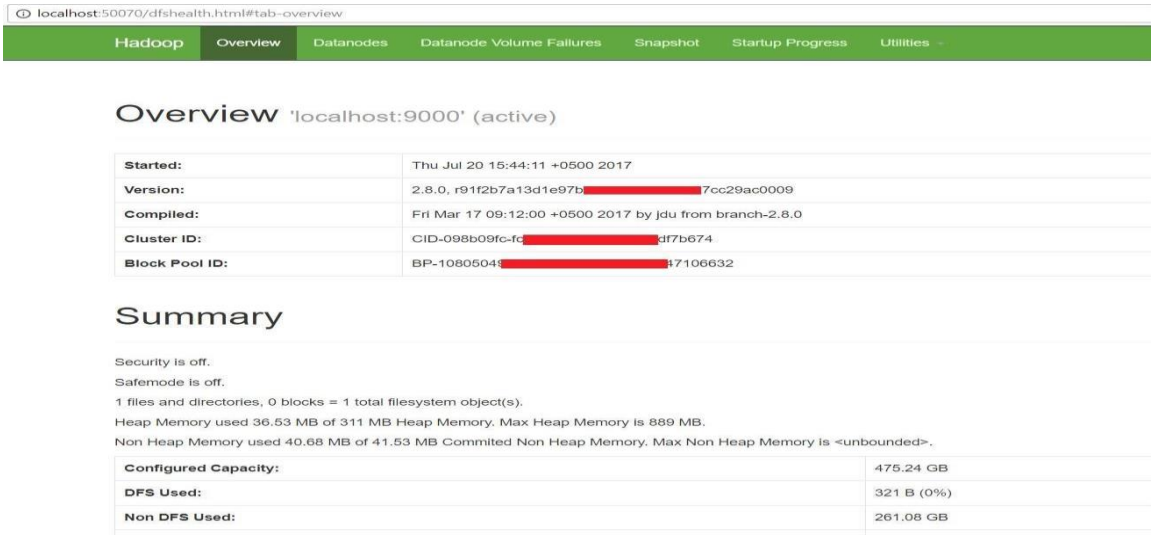
- Hadoop Namenode
- Hadoop Datanode ➢ YARN Resource Manager
- YARN Node Manager



Open: http://localhost:8088



Open: http://localhost:50070



File management tasks in Hadoop:

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: hadoop fs -cat <path[filename]>

4. Copy afile 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 /

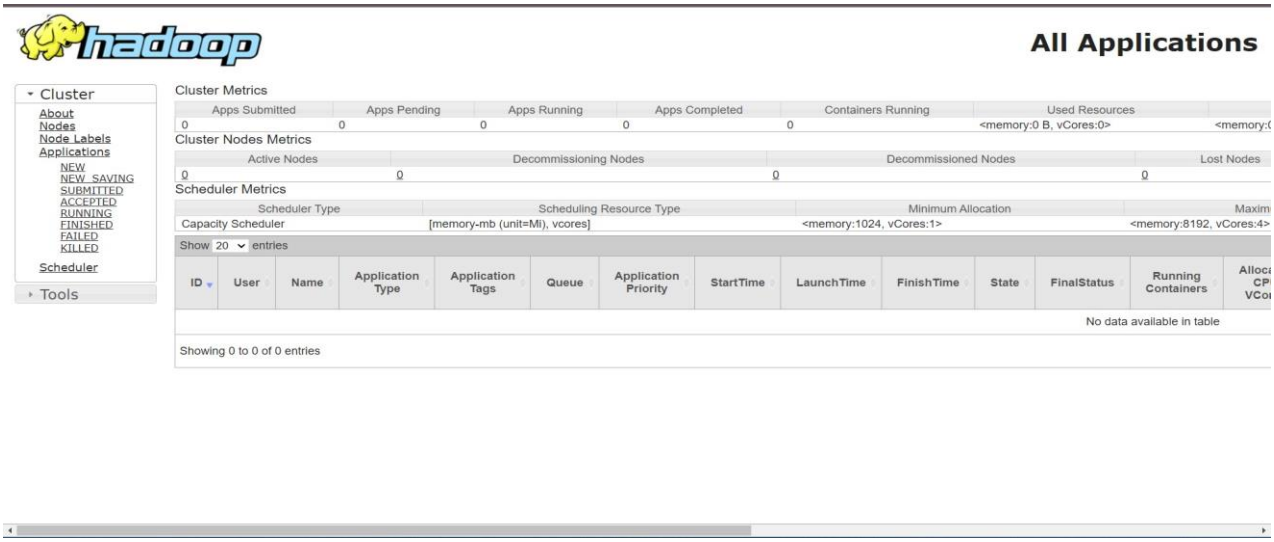
6) Copy text file from D drive to sample

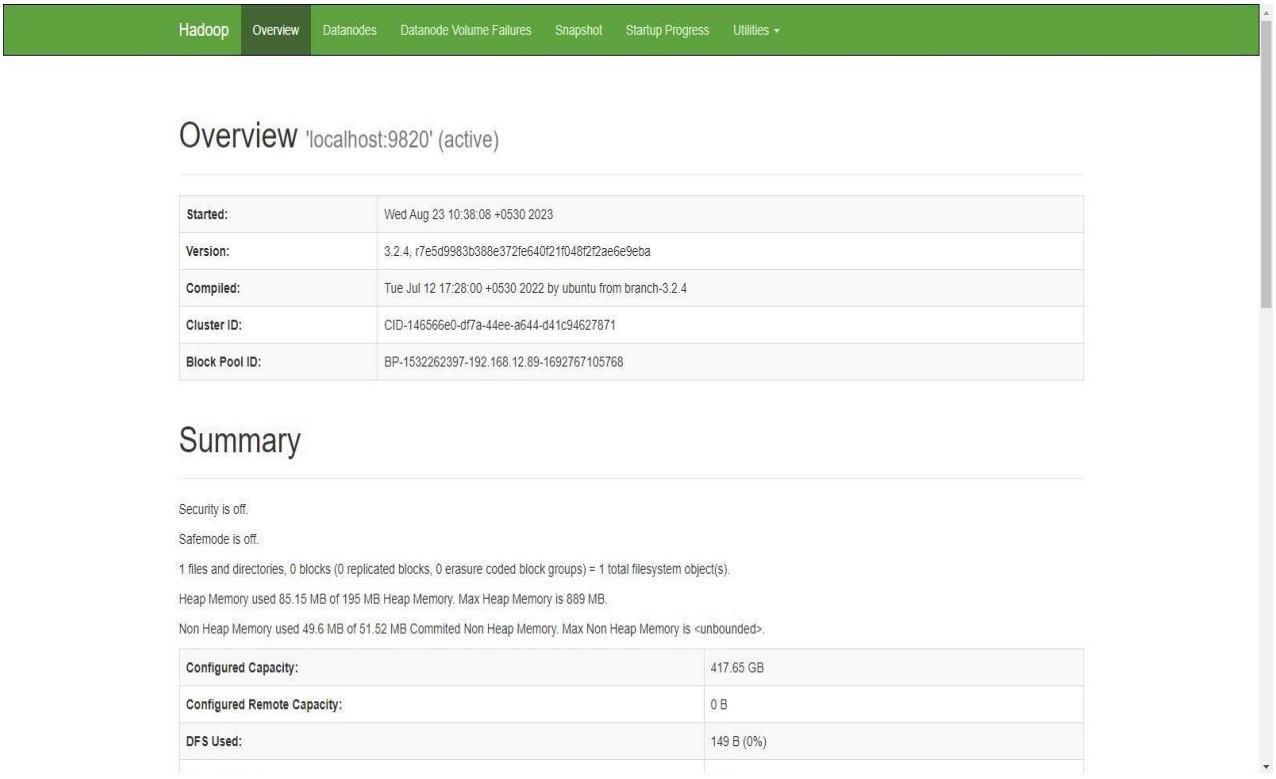
C:\hadoop-2.8.0\sbin>hdfs dfs -copyFromLocal d:\rally.txt /sample

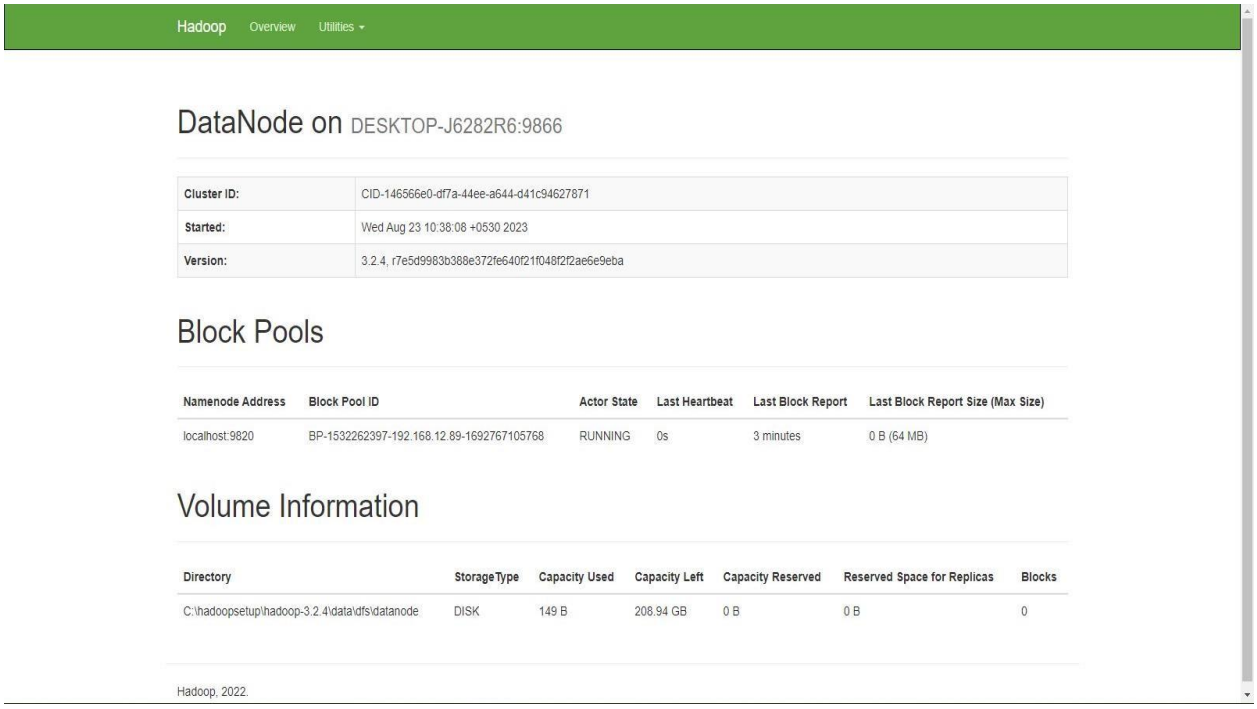
7) To verify if the file is copied

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

OUTPUT:


hadoop
All Applications
Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools
Cluster Metrics
Apps Submitted
Apps Pending
Apps Running
Apps Completed
Containers Running
Used Resources
0
0
0
0
0
<memory:0 B, vCores:0>
Cluster Nodes Metrics
Active Nodes
Decommissioning Nodes
Decommissioned Nodes
Lost Nodes
0
0
0
0
Scheduler Metrics
Scheduler Type
Scheduling Resource Type
Minimum Allocation
Capacity Scheduler
[memory-mb (unit=Mi), vcores]
<memory:1024, vCores:1>
<memory:8192, vCores:4>
Show 20 entries
ID
User
Name
Application Type
Application Tags
Queue
Application Priority
StartTime
LaunchTime
FinishTime
State
FinalStatus
Running Containers
No data available in table
Showing 0 to 0 of 0 entries


Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Overview 'localhost:9820' (active)
Started: Wed Aug 23 10:38:08 +0530 2023
Version: 3.2.4, r7e5d9983b388e372fe640f21f048f2f2ae6e9eba
Compiled: Tue Jul 12 17:28:00 +0530 2022 by ubuntu from branch-3.2.4
Cluster ID: CID-146566e0-df7a-44ee-a644-d41c94627871
Block Pool ID: BP-1532262397-192.168.12.89-1692767105768
Summary
Security is off.
Safemode is off.
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).
Heap Memory used 85.15 MB of 195 MB Heap Memory. Max Heap Memory is 889 MB.
Non Heap Memory used 49.6 MB of 51.52 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.
Configured Capacity: 417.65 GB
Configured Remote Capacity: 0 B
DFS Used: 149 B (0%)



C:\hadoopsetup\hadoop-3.3.6\sbin> hdfs dfs -mkdir /pp

C:\hadoopsetup\hadoop-3.3.6\sbin> hdfs dfs -ls /

Found 1 items

drwxr-xr-x - admin admingroup 0 2023-10-8 2:10 /pp

C:\hadoopsetup\hadoop-3.3.6\sbin> hdfs dfs -copyFromLocal

C:\Users\Pooja\OneDrive\Desktop\pp

C:\hadoopsetup\hadoop-3.3.6\sbin> hdfs dfs -ls /pp

Found 1 items

-rw-r--r-- 1 admin admingroup 14 2023-10-8 2:20 /pp/pp.txt

CONCLUSION:

The Hadoop Ecosystem, a platform or suite, provides a variety of solutions for handling big data challenges. To enable the Hadoop ecosystem to operate effectively, it's necessary to store vast datasets, whether structured or unstructured, across numerous nodes. The pivotal element responsible for safeguarding this data in the form of log files is HDFS (Hadoop Distributed File System). In our experiment, we managed to install and set up Hadoop successfully, and we also executed some elementary file management operations.