

Vidyavardhini's College of Engineering & Technology
Department of Computer Engineering



Experiment No.1
Hadoop HDFS Practical
Date of Performance.
Date of Submission:

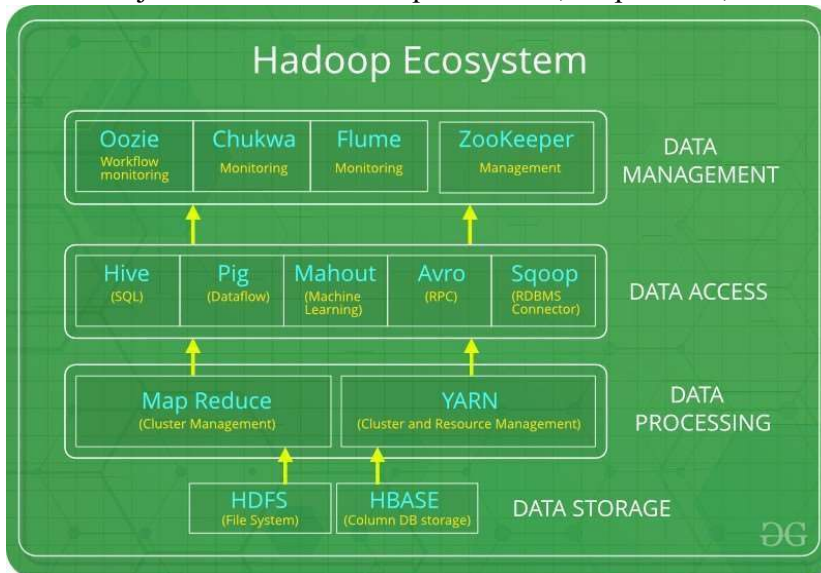


AIM : Installation, Configuration of hadoop and performing basic file management operations in hadoop.

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



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

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>)



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Java JDK 1.8.0.zip (Link:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8downloads-2133151.html>)
Check either Java 1.8.0 is already installed on your system or not, use "Javac -version" to check.

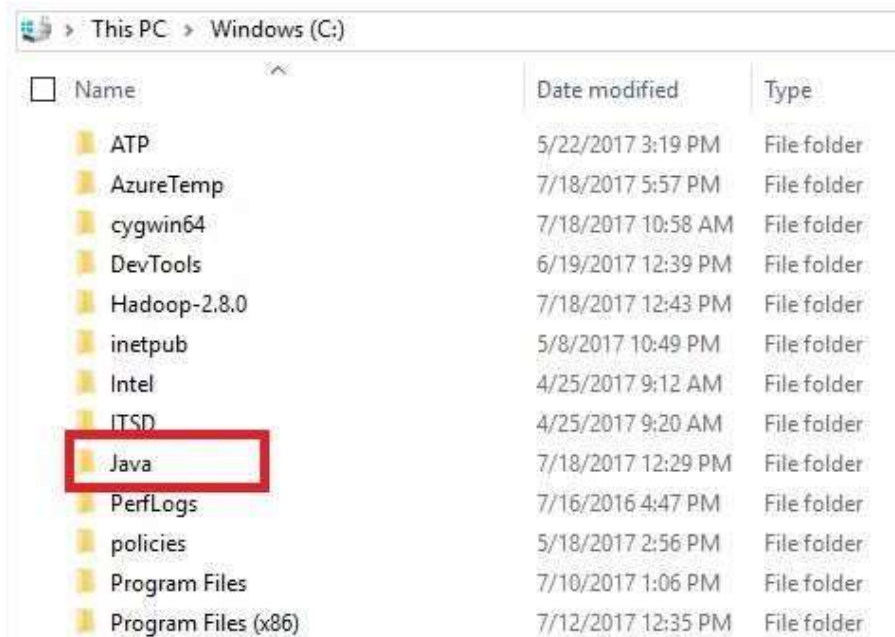
A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Windows\System32\cmd.exe". The command prompt shows the following text:
C:\>javac -version
javac 1.8.0_192
C:\>
The background of the command prompt is black, and the text is white.

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



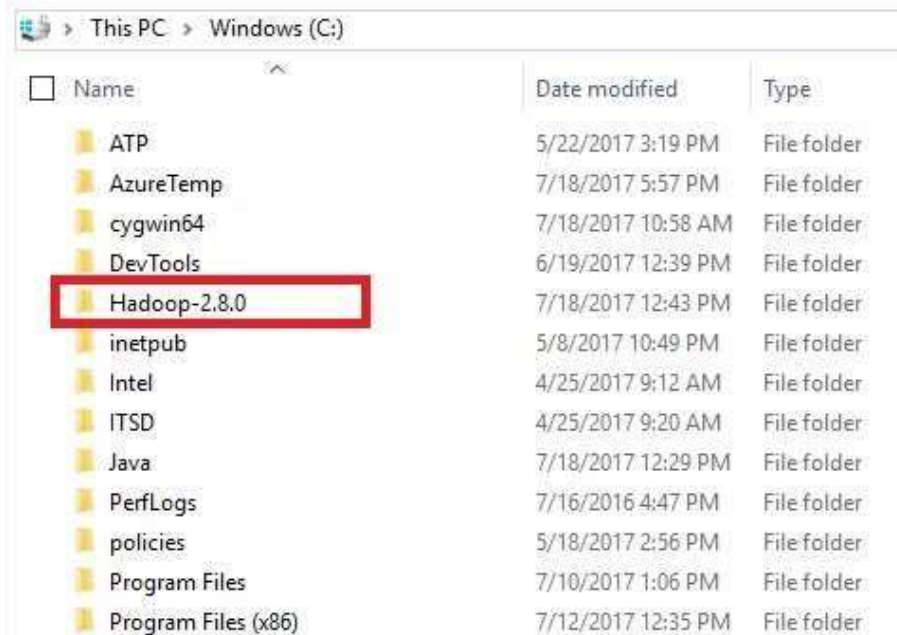
Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering



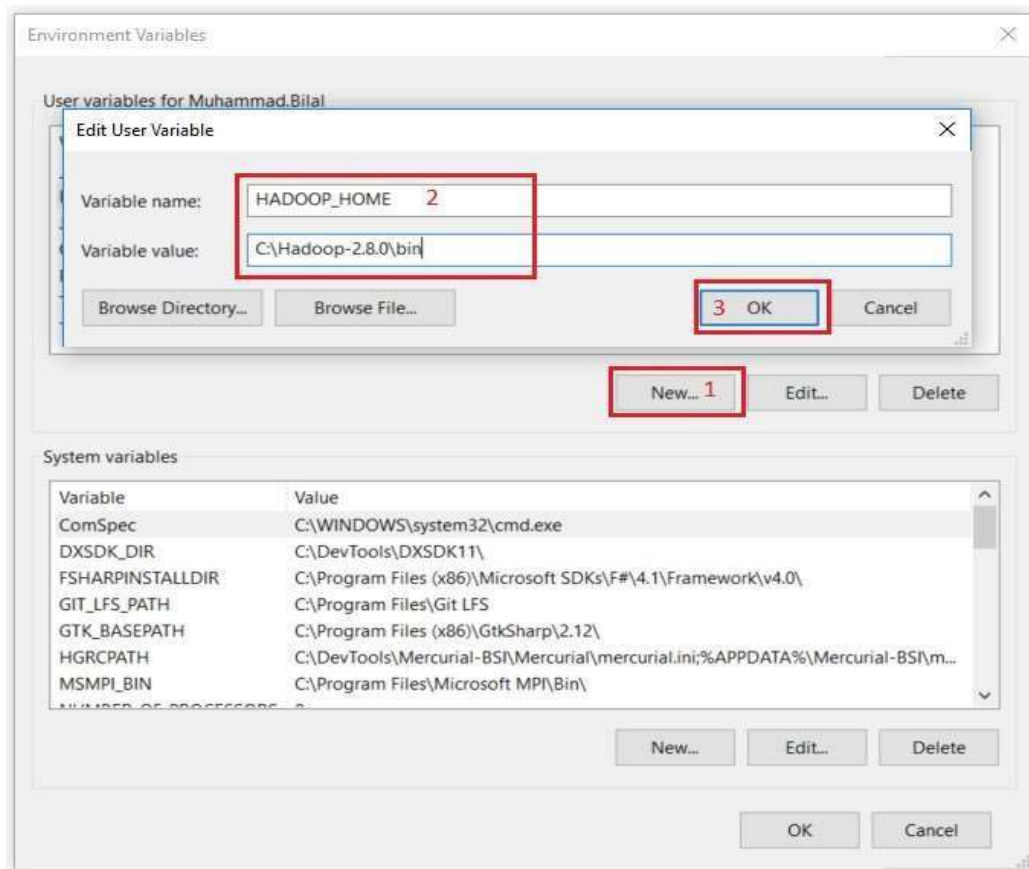
Name	Date modified	Type
ATP	5/22/2017 3:19 PM	File folder
AzureTemp	7/18/2017 5:57 PM	File folder
cygwin64	7/18/2017 10:58 AM	File folder
DevTools	6/19/2017 12:39 PM	File folder
Hadoop-2.8.0	7/18/2017 12:43 PM	File folder
inetpub	5/8/2017 10:49 PM	File folder
Intel	4/25/2017 9:12 AM	File folder
ITSD	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

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

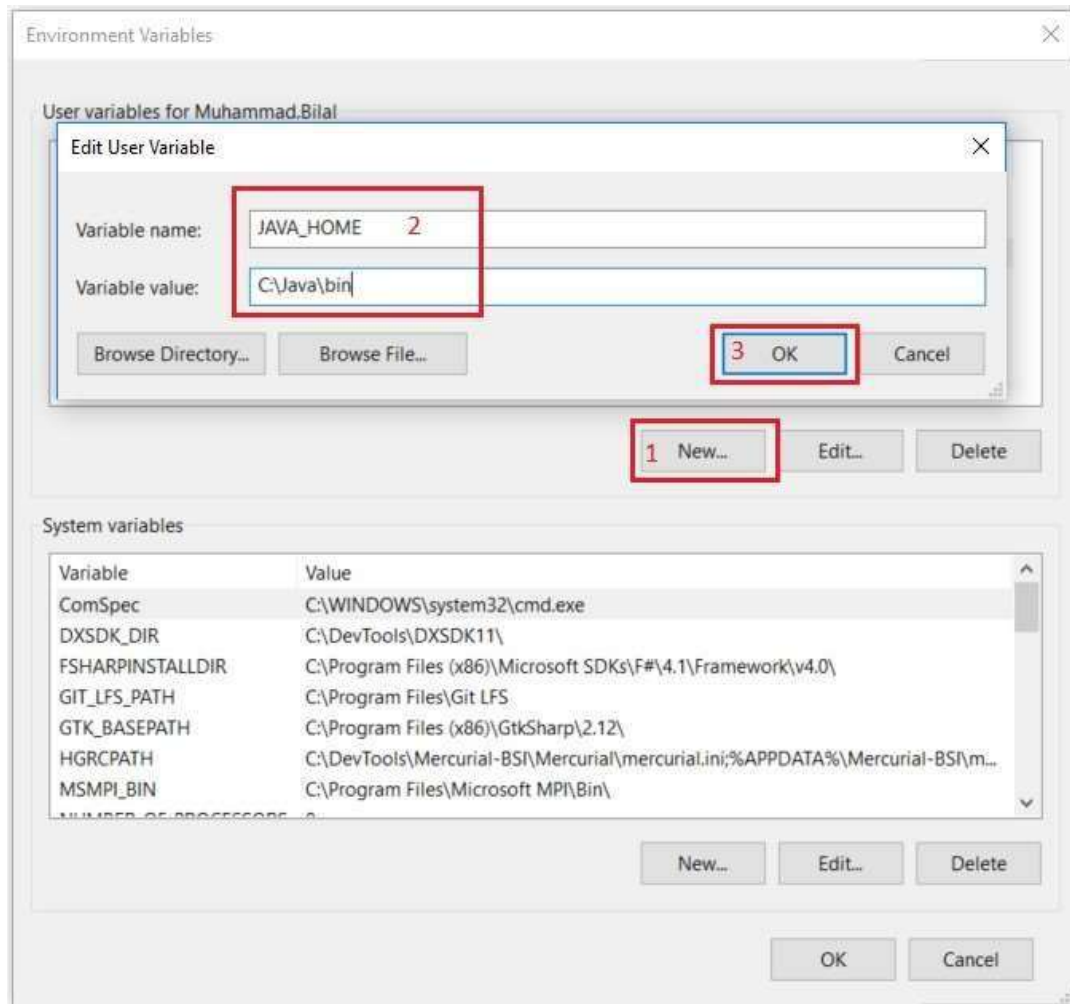


Name	Date modified	Type
ATP	5/22/2017 3:19 PM	File folder
AzureTemp	7/18/2017 5:57 PM	File folder
cygwin64	7/18/2017 10:58 AM	File folder
DevTools	6/19/2017 12:39 PM	File folder
Hadoop-2.8.0	7/18/2017 12:43 PM	File folder
inetpub	5/8/2017 10:49 PM	File folder
Intel	4/25/2017 9:12 AM	File folder
ITSD	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

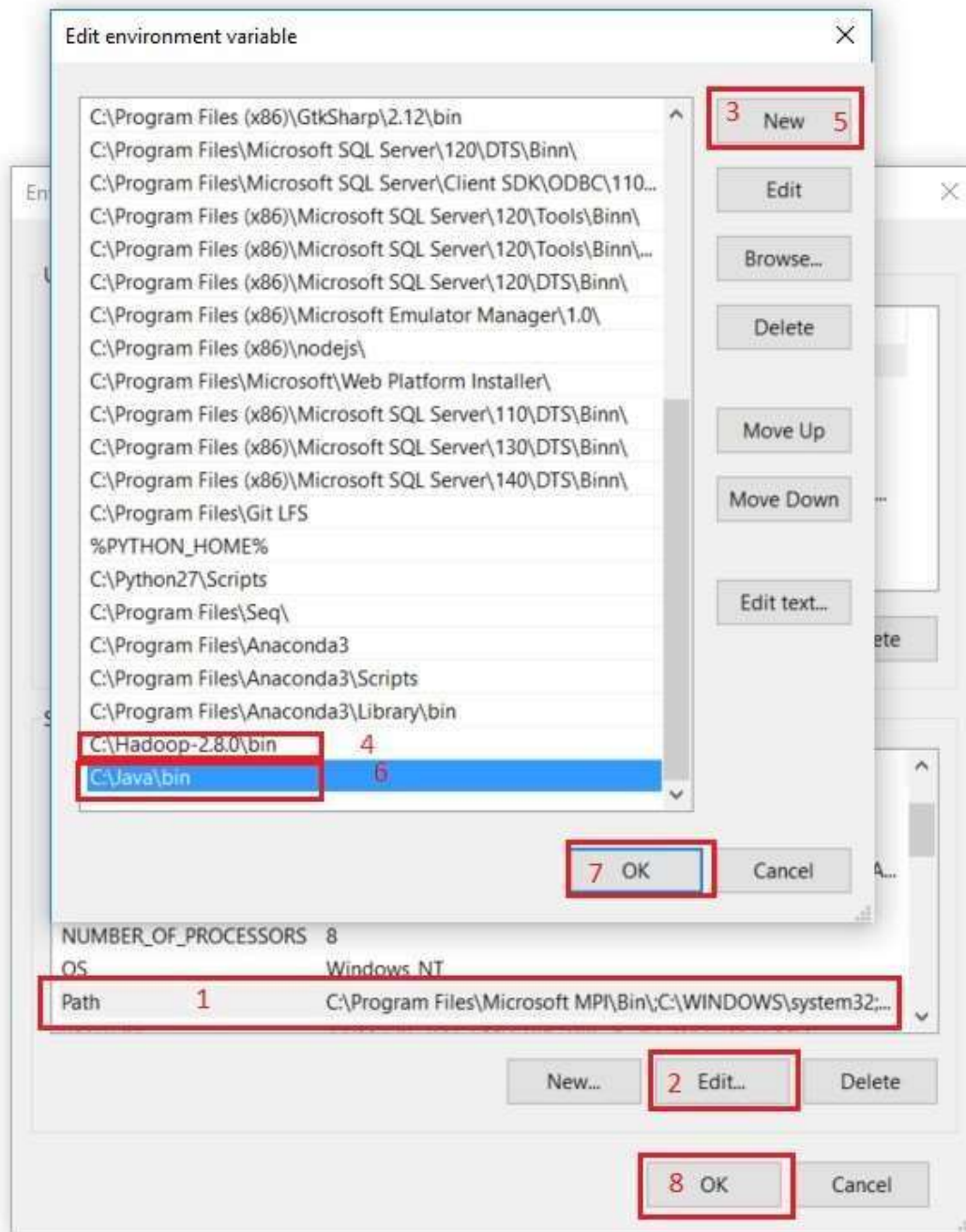
Set the path HADOOP_HOME Environment variable on windows 10(see Step 1,2,3 and 4 below).



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.



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

```
<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:/Hadoop2.8.0/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"

<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

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>
```



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

```
<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.

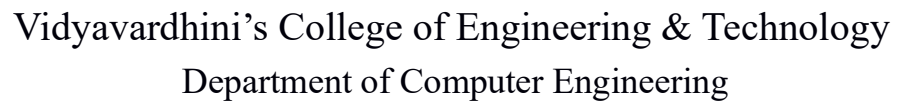
```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>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

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 :

Download file Hadoop Configuration.zip (Link: <https://github.com/MuhammadBilalYar/HADOOP-INSTALLATION-ONWINDOW10/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 :



```
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>
```

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



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

```
Apache Hadoop Distribution - hadoop_namenode
17/07/2017 15:50:09 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:12 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:15 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:18 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:21 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:24 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:27 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:30 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:33 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:36 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:39 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:42 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:46 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:49 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:52 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:55 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:50:58 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:01 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:04 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:07 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:10 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:13 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:16 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:19 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:22 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:25 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:29 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:32 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/07/2017 15:51:35 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
```

Open: <http://localhost:8088>

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	V-Cores Used	V-Cores Total	V-Cores 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

Showing 0 to 0 of 0 entries

Open: <http://localhost:50070>



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

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, r91f2b7a13d1e97b7cc29ac0009
Compiled:	Fri Mar 17 09:12:00 +0500 2017 by jdu from branch-2.8.0
Cluster ID:	CID-098b09fc-fc7b674
Block Pool ID:	BP-108050487106632

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

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:

Usage: `hadoop fs -cat`

`<path[filename]>` 4. Copy a file from
source to destination



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

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



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

OUTPUT:

The screenshot displays the Hadoop All Applications web interface. The top navigation bar includes links for Namenode information, DataNode Information, and All Applications. The main content area is titled "All Applications" and shows various cluster metrics and application status. The interface is logged in as "dr:who".

Cluster Metrics

Metric	Value
Apps Submitted	0
Apps Pending	0
Apps Running	0
Apps Completed	0
Containers Running	0
Used Resources	<memory:0 B, vCores:0>
Total Resources	<memory:8 GB, vCores:8>
Reserved Resources	<memory:0 B, vCores:0>
Physical Mem Used %	90
Physical VCores Used %	0

Cluster Nodes Metrics

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

Scheduler Metrics

Metric	Value
Scheduler Type	Capacity Scheduler
Scheduling Resource Type	[memory-mb (unit-M), vcores]
Minimum Allocation	<memory:1024, vCores:1>
Maximum Allocation	<memory:8192, vCores:4>
Maximum Cluster Application Priority	0
Scheduler Busy %	0

Applications Table

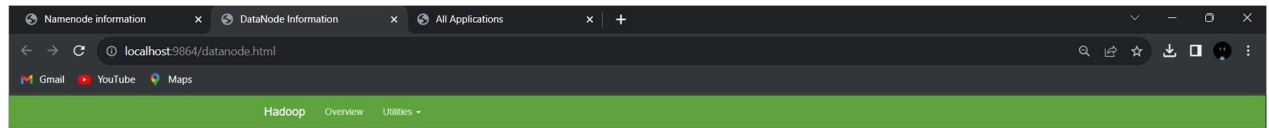
ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Allocated GPUs	Reserved CPU VCores	Reserved Memory MB	Reserved GPUs	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No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

Showing 0 to 0 of 0 entries



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering



DataNode on Shubham:9866

Cluster ID:	CID-8522f14d-a9d-422d-90eb-69d22659b532
Started:	Thu Oct 19 22:37:40 +0530 2023
Version:	3.2.4, r7e5d9983b388e372fe640f21f048f22ae6e9eba

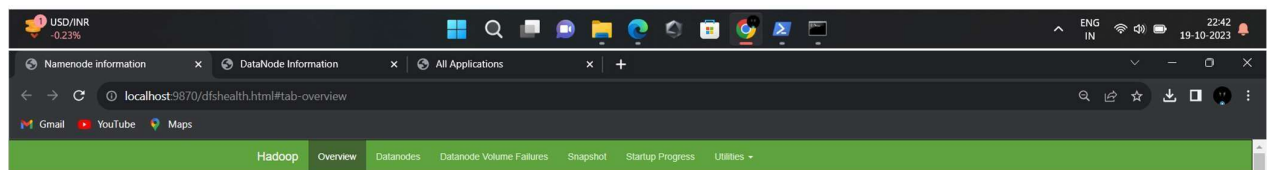
Block Pools

namenode address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
localhost:9820	BP-446742625-192.168.73.157-1697140882384	RUNNING	1s	4 minutes	0 B (64 MB)

Volume Information

Directory	Storage Type	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
C:\hadoop-3.2.4\data\dfs\datanode	DISK	340 B	48.9 GB	0 B	0 B	0

Hadoop, 2022.



Overview 'localhost:9820' (active)

Started:	Thu Oct 19 22:37:39 +0530 2023
Version:	3.2.4, r7e5d9983b388e372fe640f21f048f22ae6e9eba
Compiled:	Tue Jul 12 17:28:00 +0530 2022 by ubuntu from branch-3.2.4
Cluster ID:	CID-8522f14d-a9d-422d-90eb-69d22659b532
Block Pool ID:	BP-446742625-192.168.73.157-1697140882384

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 36.49 MB of 209.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 51.25 MB of 52.98 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	280.09 GB
Configured Remote Capacity:	0 B
DFS Used:	340 B (0%)
Non DFS Used:	231.19 GB
DFS Remaining:	48.9 GB (17.46%)
Block Pool Used:	340 B (0%)



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> cd \
PS C:\> cd .\hadoop-3.2.4\
PS C:\hadoop-3.2.4> cd sbin
PS C:\hadoop-3.2.4\sbin> .\start-dfs.cmd
PS C:\hadoop-3.2.4\sbin> .\start-yarn.cmd
starting yarn daemons
PS C:\hadoop-3.2.4\sbin> jps
21648 NodeManager
3472 NameNode
24024 ResourceManager
13628 Jps
15884 DataNode
PS C:\hadoop-3.2.4\sbin>
```

```
Administrator: Windows PowerShell
PS C:\hadoop-3.2.4\sbin> hdfs dfs -mkdir /prac_1
PS C:\hadoop-3.2.4\sbin> hdfs dfs -ls /
Found 1 items
drwxr-xr-x - sdnak supergroup          0 2023-10-19 22:50 /prac_1
PS C:\hadoop-3.2.4\sbin> hdfs dfs -copyFromLocal C:\Users\sdnak\OneDrive\Desktop /prac_1
PS C:\hadoop-3.2.4\sbin> hdfs dfs -ls /prac_1
Found 1 items
drwxr-xr-x - sdnak supergroup          0 2023-10-19 22:52 /prac_1/Desktop
PS C:\hadoop-3.2.4\sbin>
```



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CONCLUSION :

The Experiment centred around the installation and configuration of Hadoop, a distributed data processing framework. It effectively demonstrated the setup of various Hadoop components, such as HDFS and MapReduce, illustrating its scalability and resilience to faults. We delved into fundamental file management tasks within Hadoop, including actions like uploading data, creating directories, and manipulating files. The advantages of parallel processing through MapReduce, the significance of data replication, and the importance of data locality became evident. The experiment underscored the learning curve associated with Hadoop while highlighting its potential for efficient data management and processing in large-scale applications, positioning it as a valuable skill in the age of big data.



Vidyavardhini's College of Engineering & Technology
Department of Computer Engineering
