



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

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

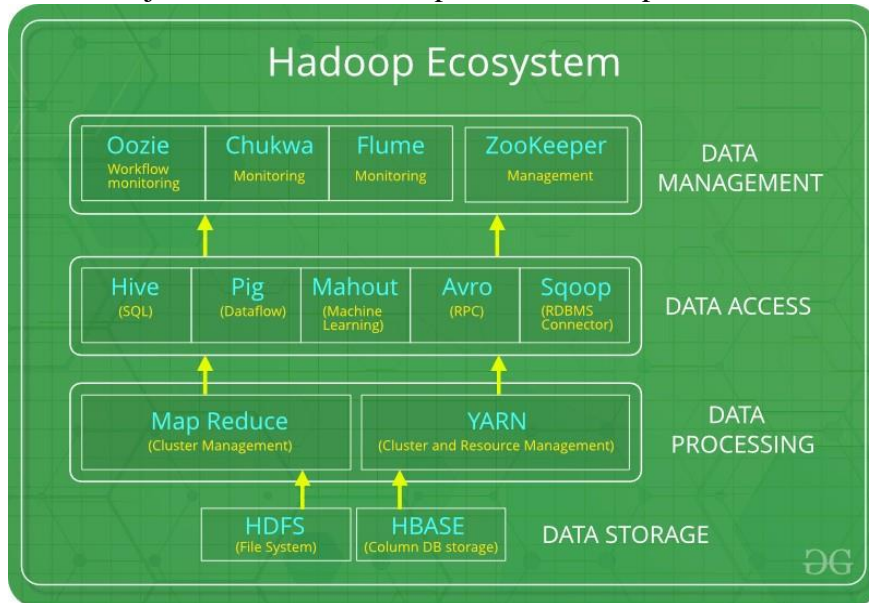


AIM : Installation, Configuration of hadoop and performing basic file management operations in hadoop.

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.

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>)

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.



```
C:\Windows\System32\cmd.exe

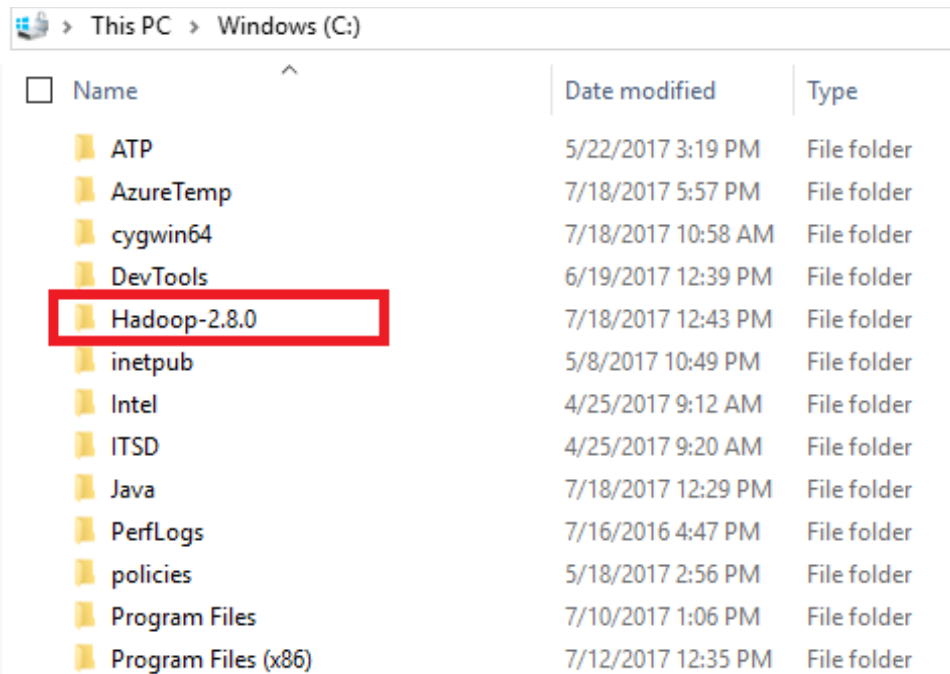
C:\>javac -version
javac 1.8.0_192

C:\>
```

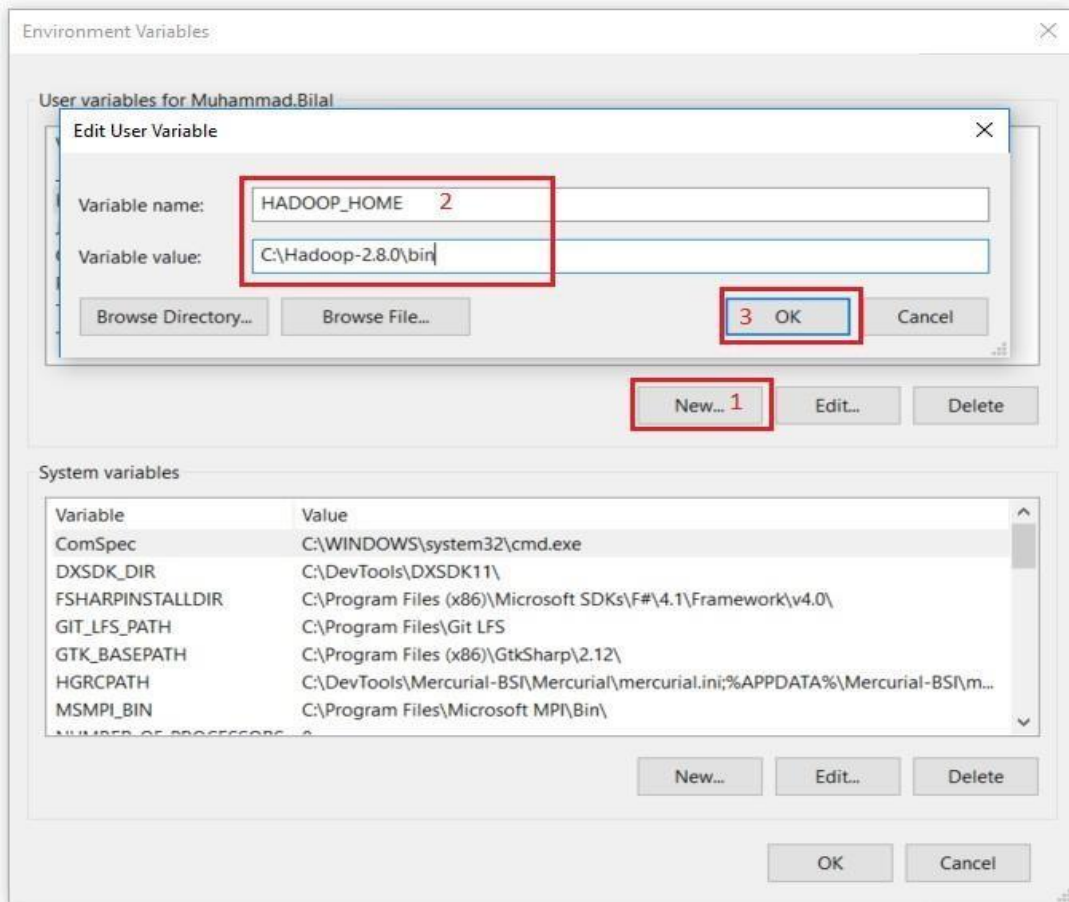
If Java is not installed on your system then first install java under "C:\JAVA"

| This PC > Windows (C:) | | |
|-------------------------------|--------------------|-------------|
| <input type="checkbox"/> 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".

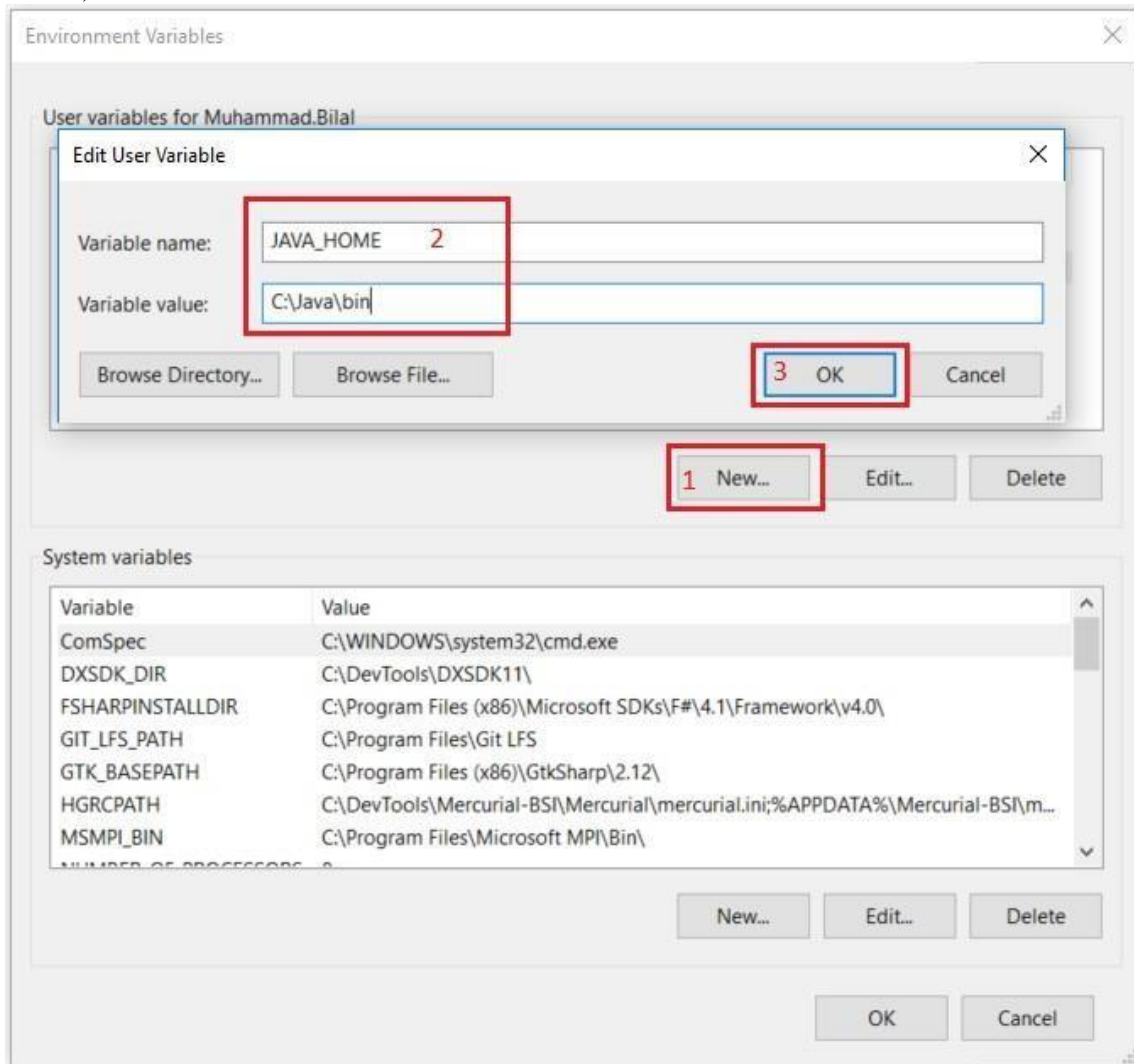


Set the path HADOOP_HOME Environment variable on windows 10(see Step 1,2,3 and 4 below).

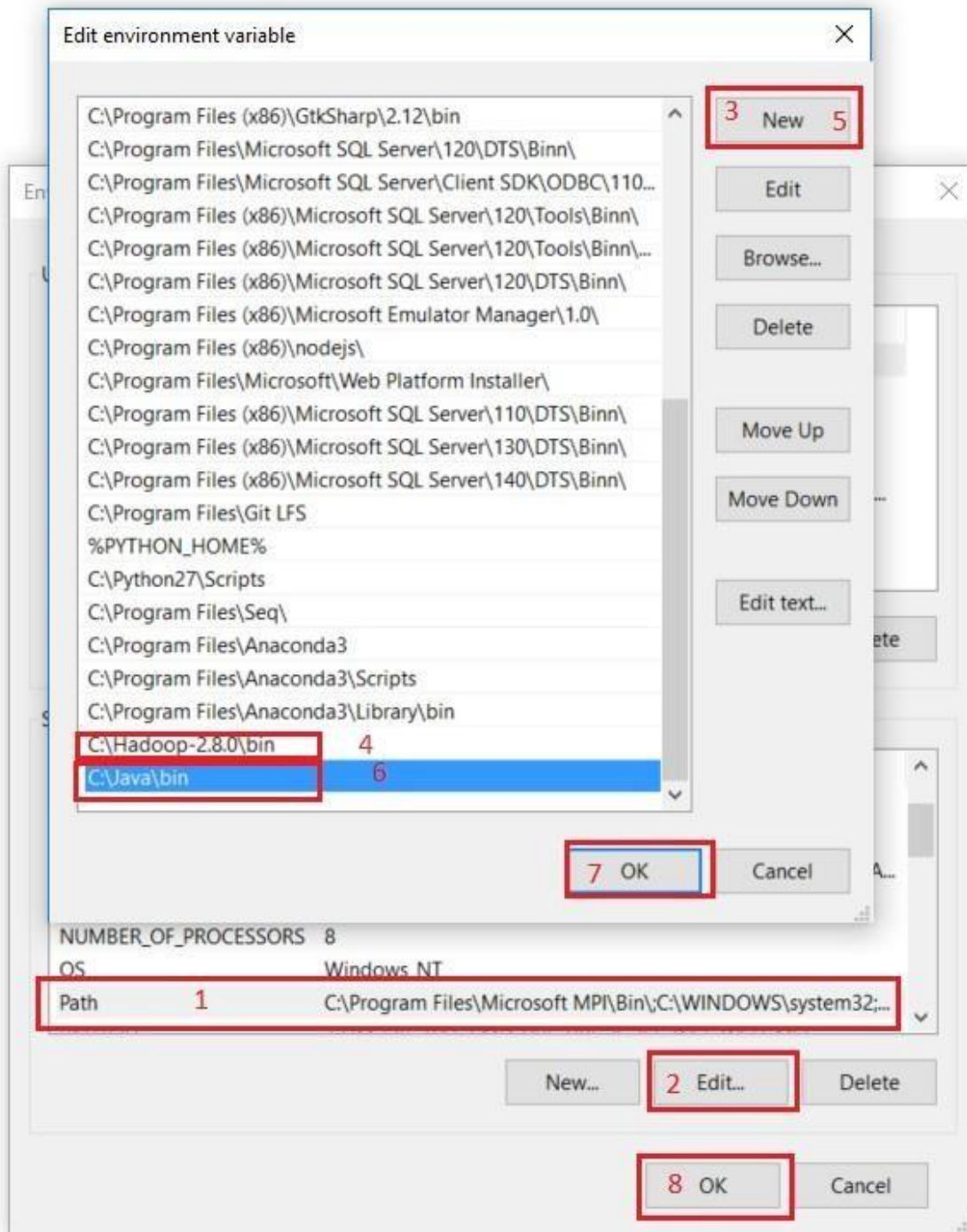




Set the path JAVA_HOME Environment variable on windows 10(see Step 1,2,3 and 4 below).



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"

| <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>
<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)

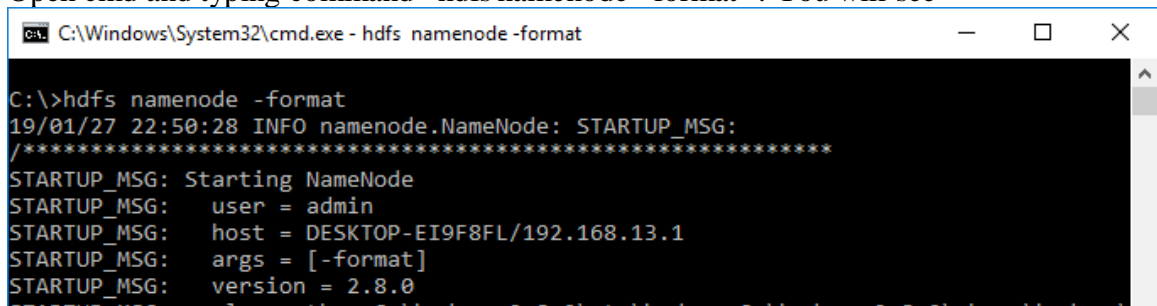
```
@rem The java implementation to use. Required.
@rem set JAVA_HOME=%JAVA_HOME%
set JAVA_HOME=C:\java
```

HADOOP CONFIGURATION :

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



```
C:\Windows\System32\cmd.exe - hdfs namenode -format

C:\>hdfs namenode -format
19/01/27 22:50:28 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   user = admin
STARTUP_MSG:   host = DESKTOP-EI9F8FL/192.168.13.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.8.0
STARTUP_MSG:   ...
```

TESTING :

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



```
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>
```

Make sure these apps are running :

Hadoop Namenode

Hadoop datanode

YARN Resourc Manager

YARN Node Manager

```
Apache Hadoop Distribution - hadoop namenode
17/6/17/07/20 Apache Hadoop Distribution - hadoop datanode
17/6/17/07/20 Apache Hadoop Distribution - yarn resourcemanager
17/6/17/07/20 Apache Hadoop Distribution - yarn nodemanager

17/6/17/07/20 15:50:00 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:12 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:15 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:18 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:21 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:24 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:27 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:30 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:33 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:36 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:39 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:42 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:46 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:49 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:52 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:55 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:50:58 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:01 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:04 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:07 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:10 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:13 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:16 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:19 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:22 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:25 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:29 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:32 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/6/17/07/20 15:51:35 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
```

Open: <http://localhost:8088>



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

← → localhost:8088/cluster

hadoop

Logged in as: dr.who

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 | 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 |

Show: 20 entries

Search:

| ID | User | Name | Application Type | Queue | Application Priority | StartTime | FinishTime | State | FinalStatus | Running Containers | Allocated CPU V-Cores | 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

Open: <http://localhost:50070>

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-fd7b674 |
| 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

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`



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

7) To verify if the file is copied
C:\hadoop-2.8.0\sbin>hdfs dfs -ls /sample

OUTPUT:

The screenshot shows the Hadoop Overview page in a web browser. The browser tabs include 'Namenode information', 'DataNode Information', 'All Applications', and 'BDA Lab Manual 23-24'. The address bar shows 'localhost:9870/dfshealth.html#tab-overview'. The page has a green header with 'Hadoop' and a navigation menu with 'Overview', 'Datanodes', 'Datanode Volume Failures', 'Snapshot', 'Startup Progress', and 'Utilities'. The main content area is titled 'Overview 'localhost:9820' (active)'. It contains a table with the following information:

| | |
|----------------|--|
| 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 |

Below the table is a 'Summary' section. It states: '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 689 MB.', and 'Non Heap Memory used 49.6 MB of 51.52 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.' Below this is another table:

| | |
|-----------------------------|------------|
| Configured Capacity: | 417.65 GB |
| Configured Remote Capacity: | 0 B |
| DFS Used: | 149 B (0%) |

The Windows taskbar at the bottom shows the date and time as 10:48 AM on 8/23/2023.

The screenshot shows the Hadoop DataNode page in a web browser. The browser tabs include 'Namenode information', 'DataNode Information', 'All Applications', and 'BDA Lab Manual 23-24'. The address bar shows 'localhost:3864/datanode.html'. The page has a green header with 'Hadoop' and a navigation menu with 'Overview' and 'Utilities'. The main content area is titled 'DataNode on DESKTOP-J6282R6:9866'. It contains a table with the following information:

| | |
|-------------|--|
| Cluster ID: | CID-146566e0-df7a-44ee-a644-d41c94627871 |
| Started: | Wed Aug 23 10:38:08 +0530 2023 |
| Version: | 3.2.4, r7e5d9983b388e372fe640f21f048f2f2ae6e9eba |

Below the table is a 'Block Pools' section. It contains a table with the following information:

| Namenode Address | Block Pool ID | Actor State | Last Heartbeat | Last Block Report | Last Block Report Size (Max Size) |
|------------------|---|-------------|----------------|-------------------|-----------------------------------|
| localhost:9820 | BP-1532262397-192.168.12.89-1692767105768 | RUNNING | 0s | 3 minutes | 0 B (64 MB) |

Below the table is a 'Volume Information' section. It contains a table with the following information:

| Directory | Storage Type | Capacity Used | Capacity Left | Capacity Reserved | Reserved Space for Replicas | Blocks |
|---|--------------|---------------|---------------|-------------------|-----------------------------|--------|
| C:\hadoopsetup\hadoop-3.2.4\data\dfs\datanode | DISK | 149 B | 208.94 GB | 0 B | 0 B | 0 |

The Windows taskbar at the bottom shows the date and time as 10:48 AM on 8/23/2023.



Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

Cluster Metrics

| Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Used Resources | Total Resources | Reserved Resources | Physical Mem Us |
|----------------|--------------|--------------|----------------|--------------------|------------------------|-------------------------|------------------------|-----------------|
| 0 | 0 | 0 | 0 | 0 | <memory:0 B, vCores:0> | <memory:8 GB, vCores:8> | <memory:0 B, vCores:0> | 75 |

Cluster Nodes Metrics

| Active Nodes | Decommissioning Nodes | Decommissioned Nodes | Lost Nodes | Unhealthy Nodes | Rebooted Nodes |
|--------------|-----------------------|----------------------|------------|-----------------|----------------|
| 1 | 0 | 0 | 0 | 0 | 0 |

Scheduler Metrics

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

Application List

| 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 |
|----------------------------|------|------|------------------|-------|----------------------|-----------|------------|------------|-------|-------------|--------------------|----------------------|---------------------|----------------|---------------------|--------------------|---------------|------------|--------------|
| No data available in table | | | | | | | | | | | | | | | | | | | |

Showing 0 to 0 of 0 entries

- 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's main objective is to install and configure Hadoop, which is a distributed data processing framework. It effectively showcases the setup of key Hadoop components such as HDFS and MapReduce. Additionally, it covers fundamental file management tasks within Hadoop, which involve tasks like directory creation and file manipulation. To harness the full potential of Hadoop, it's essential to gain proficiency in its diverse elements, including the Hadoop Distributed File System (HDFS) and MapReduce, and also explore the extensive array of tools and frameworks that enhance Hadoop, such as Apache Hive, Pig, and Spark.