



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

---

Name: Ajay Shitkar
Roll No: 58
Experiment No.: 10
Distributed File System
Date of Performance: 05/04/2024
Date of Submission: 08/04/2024



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

---

**Aim:** To demonstrate a Distributed File System

**Objective:** Demonstrate a Distributed File System using HDFS

### Theory:

HDFS employs a NameNode and DataNode architecture to implement a distributed file system that provides high-performance access to data across highly scalable Hadoop clusters.

### 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



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

---

Usage:

`hadoop fs -copyFromLocal <localsrc> URI`

6. 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`



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

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
```

### Code and output:

```
Command Prompt
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>hadoop
Usage: hadoop [--config confdir] [--loglevel loglevel] COMMAND
where COMMAND is one of:
  fs                run a generic filesystem user client
  version           print the version
  jar <jar>         run a jar file
                    note: please use "yarn jar" to launch
                      YARN applications, not this command.
  checknative [-a|-h] check native hadoop and compression libraries availability
  conftest         validate configuration XML files
  distch path:owner:group:permission
                  distributed metadata changer
  distcp <srcurl> <desturl> copy file or directories recursively
  archive -archiveName NAME -p <parent path> <src> <dest> create a hadoop archive
  classpath        prints the class path needed to get the
                  Hadoop jar and the required libraries
  credential       interact with credential providers
  jnipath          prints the java.library.path
  kerbname         show auth_to_local principal conversion
  kdiag           diagnose kerberos problems
  key             manage keys via the KeyProvider
  trace           view and modify Hadoop tracing settings
  daemonlog       get/set the log level for each daemon
  or
  CLASSNAME       run the class named CLASSNAME

Most commands print help when invoked w/o parameters.

C:\Users\admin>hadoop version
Hadoop 3.2.4
Source code repository Unknown -r 7e5d9983b388e372fe640f21f048f2f2ae6e9eba
Compiled by ubuntu on 2022-07-12T11:58Z
Compiled with protoc 2.5.0
From source with checksum ee031c16fe785bbb35252c749418712
This command was run using /C:/hadoop/share/hadoop/common/hadoop-common-3.2.4.jar

C:\Users\admin>
```



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

```
Administration: Command Prompt
2023-09-04 06:52:25,264 INFO namenode.MNStorageRetentionManager: Going to retain 1 images with txid >= 0
2023-09-04 06:52:25,276 INFO namenode.FSNamesystem: Stopping services started for active state
2023-09-04 06:52:25,276 INFO namenode.FSNamesystem: Stopping services started for standby state
2023-09-04 06:52:25,279 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2023-09-04 06:52:25,279 INFO namenode.NameNode: SHUTDOWN_MSG:
=====
SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-IRCC854/192.168.12.91
=====

C:\Windows\System32>cd \
C:\>cd hadoop
C:\hadoop>cd sbin
C:\hadoop\sbin>start-dfs.cmd

C:\hadoop\sbin>jps
10560 NameNode
8768 DataNode
7812 Jps

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

C:\hadoop\sbin>jps
10560 NameNode
8768 DataNode
12324 Jps
12984 ResourceManager
680 NodeManager

C:\hadoop\sbin>
```

```
Environment Variables
Select Administrator: Command Prompt

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

C:\hadoop\sbin>jps
10560 NameNode
8768 DataNode
12324 Jps
12984 ResourceManager
680 NodeManager

C:\hadoop\sbin>hdfs dfs -mkdir /polomii
C:\hadoop\sbin>hdfs dfs -put C:\Users\admin\Downloads\simple.txt /polomii
C:\hadoop\sbin>hdfs dfs -tail /polomii/simple.txt
hii how are you
C:\hadoop\sbin>
```

```
simple
File Edit View
hii how are you

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

```
Administrator: Command Prompt

Reduce output records=2
Spilled Records=4
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=62
CPU time spent (ms)=280
Physical memory (bytes) snapshot=514666496
Virtual memory (bytes) snapshot=808771584
Total committed heap usage (bytes)=391643136
Peak Map Physical memory (bytes)=305762304
Peak Map Virtual memory (bytes)=439504896
Peak Reduce Physical memory (bytes)=208904192
Peak Reduce Virtual memory (bytes)=369356800

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=13
File Output Format Counters
Bytes Written=16

C:\hadoop\sbin>hdfs dfs -rm -r /output/hello.txt
Deleted /output/hello.txt

C:\hadoop\sbin>
```

### Conclusion:

The demonstration of a Distributed File System using HDFS highlights its capability to efficiently store and manage large volumes of data across multiple nodes in a distributed environment. By employing a master-slave architecture and data replication, HDFS ensures fault tolerance and high availability, crucial for modern big data applications. The seamless scalability of HDFS allows for easy expansion to accommodate growing data needs, while its optimized data processing framework facilitates parallel processing for enhanced performance. Overall, the utilization of HDFS demonstrates its effectiveness in handling the challenges of distributed storage and processing, making it a cornerstone technology in the realm of big data analytics and storage.