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

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

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

5) To verify if the directory is created

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

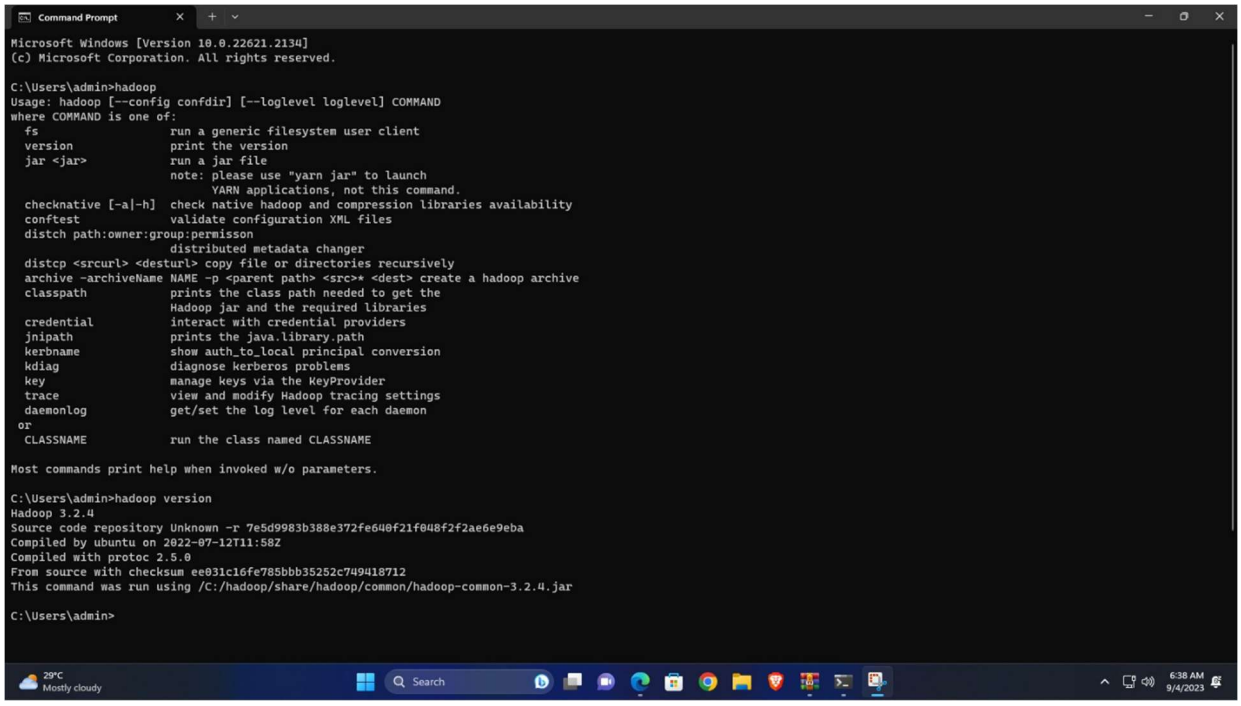
6) Copy text file from D drive to sample

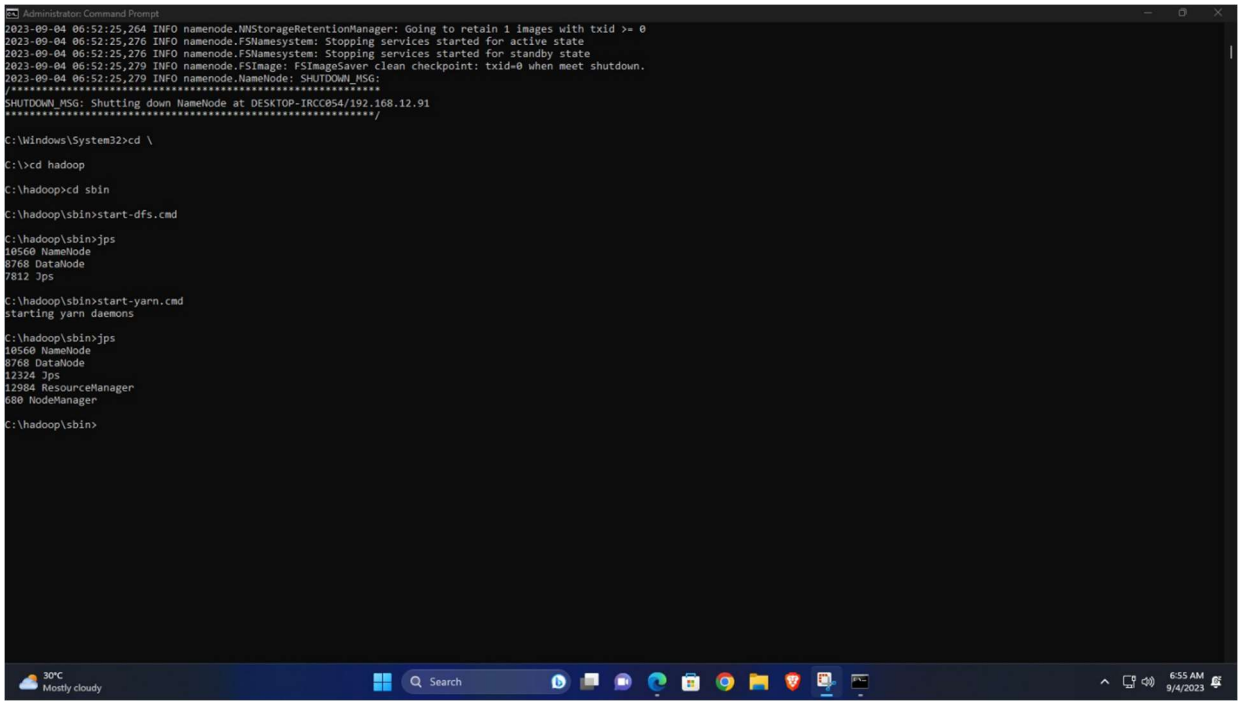
C:\hadoop-2.8.0\sbin>hdfs dfs -copyFromLocal d:\rally.txt /sample

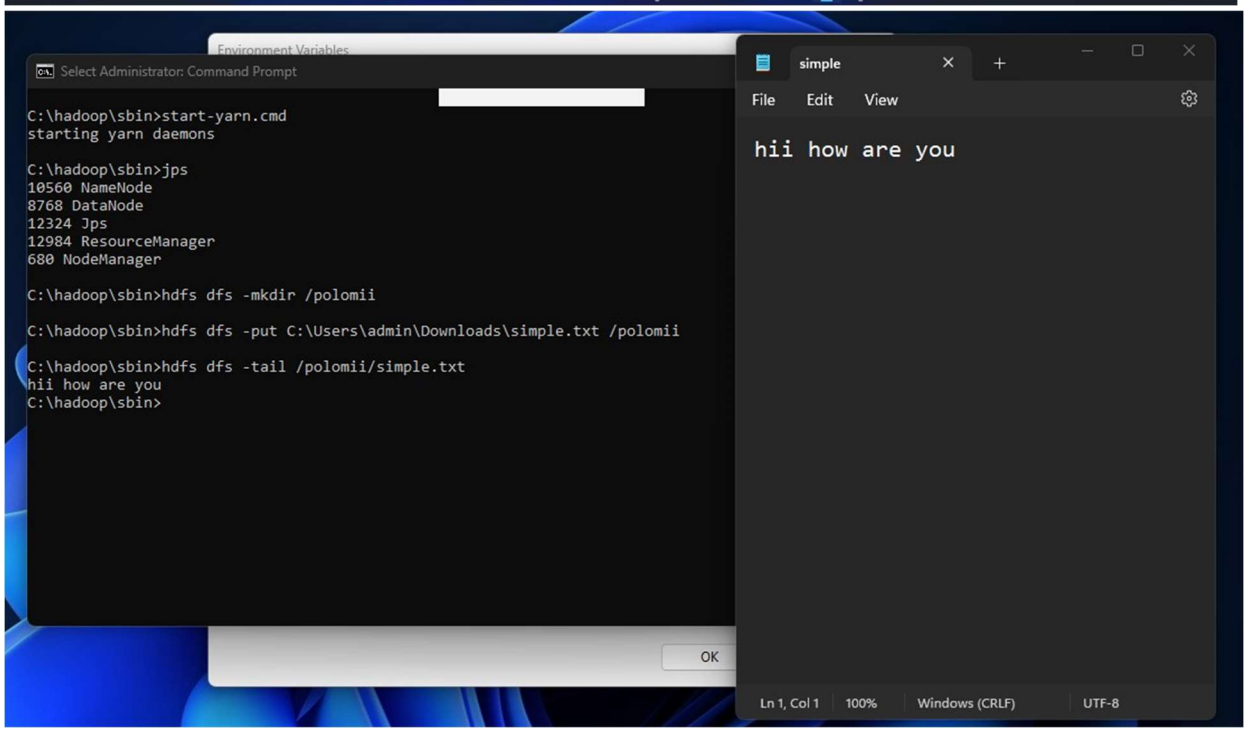
7) To verify if the file is copied

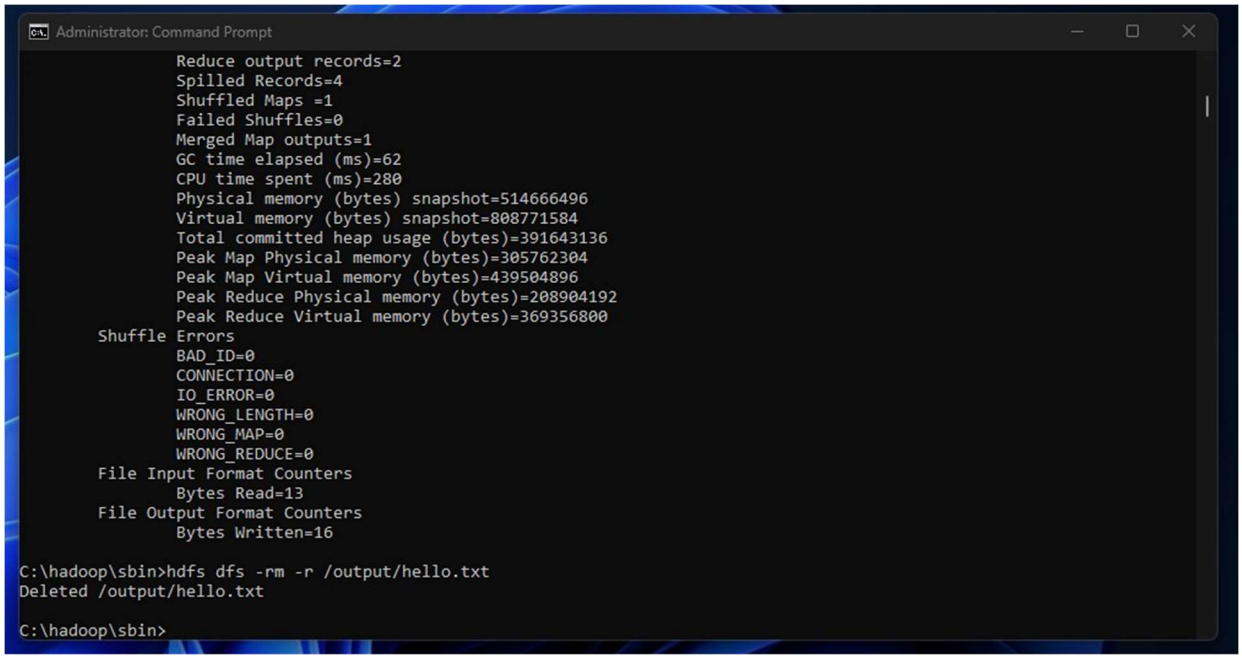
C:\hadoop-2.8.0\sbin>hdfs dfs -ls /sample

Code and output:




Administrator: Command Prompt
2023-09-04 06:52:25,264 INFO namenode.NNStorageRetentionManager: 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-IRCC054/192.168.12.91
**/
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>
30°C
Mostly cloudy
Search
6:55 AM
9/4/2023


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>
OK
simple
File
Edit
View
hii how are you
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8



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.