2. Program to Perform Read and Write Operations Between Local File System and HDFS Using FileSystem API

import java.io.\*;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.\*;

public class HDFSFileOperations {

public static void main(String[] args) throws IOException {

// Load Hadoop configuration

Configuration conf = new Configuration();

// Set HDFS NameNode URI (change this based on your Hadoop cluster setup)

conf.set("fs.defaultFS", "hdfs://localhost:9000");

// Get the HDFS FileSystem object

FileSystem fs = FileSystem.get(conf);

// Define local and HDFS file paths

Path localFilePath = new Path("/home/user/sample.txt"); // Local file path

Path hdfsFilePath = new Path("/user/hadoop/sample.txt"); // HDFS destination path

// 1WRITE from Local File System to HDFS

writeFileToHDFS(fs, localFilePath, hdfsFilePath);

// 2READ from HDFS to Local File System

readFileFromHDFS(fs, hdfsFilePath);

// Close the FileSystem

fs.close();

}

// Method to Write a File from Local System to HDFS

public static void writeFileToHDFS(FileSystem fs, Path localPath, Path hdfsPath) throws IOException {

// Open input stream to read from local file system

FSDataOutputStream outputStream = fs.create(hdfsPath, true);

BufferedReader br = new BufferedReader(new FileReader(localPath.toString()));

String line;

while ((line = br.readLine()) != null) {

outputStream.writeBytes(line + "\n");

}

br.close();

outputStream.close();

System.out.println("File successfully written to HDFS: " + hdfsPath);

}

// Method to Read a File from HDFS

public static void readFileFromHDFS(FileSystem fs, Path hdfsPath) throws IOException {

if (!fs.exists(hdfsPath)) {

System.out.println("File not found in HDFS: " + hdfsPath);

return;

}

FSDataInputStream inputStream = fs.open(hdfsPath);

BufferedReader br = new BufferedReader(new InputStreamReader(inputStream));

System.out.println("Reading file from HDFS:");

String line;

while ((line = br.readLine()) != null) {

System.out.println(line);

}

br.close();

inputStream.close();

}

}