

STEP 1: START HADOOP (Always first)

Open **CMD as Administrator**

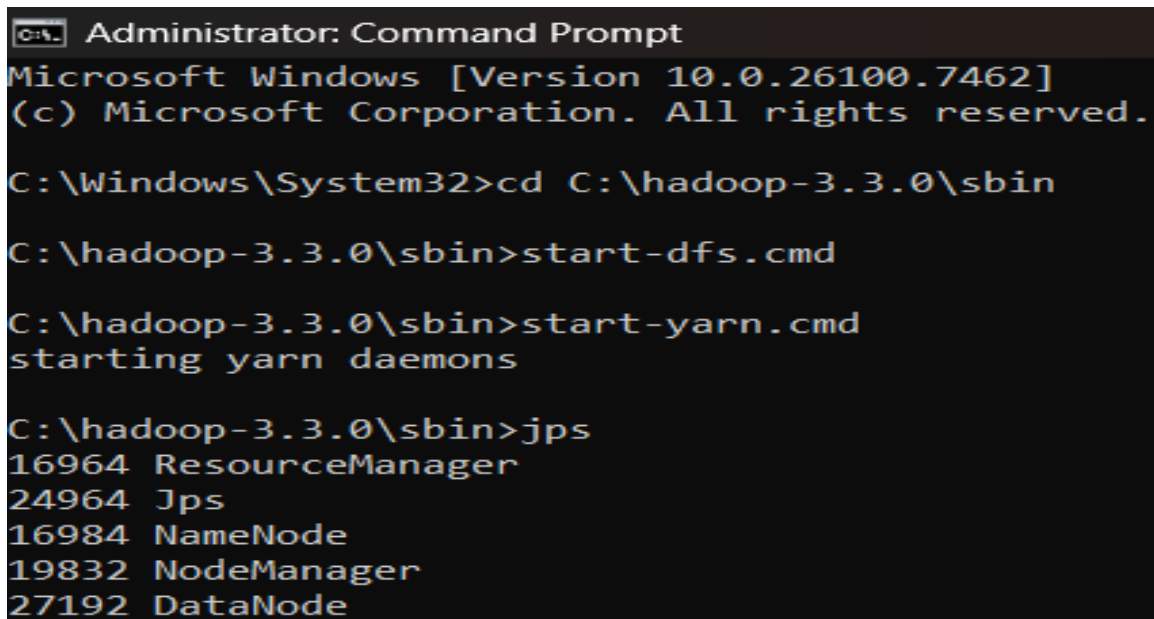
```
cd C:\hadoop-3.3.0\sbin
```

```
start-dfs.cmd
```

```
start-yarn.cmd
```

Verify:

```
jps
```



A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window shows the following commands and output:

```
Microsoft Windows [Version 10.0.26100.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\hadoop-3.3.0\sbin

C:\hadoop-3.3.0\sbin>start-dfs.cmd

C:\hadoop-3.3.0\sbin>start-yarn.cmd
starting yarn daemons

C:\hadoop-3.3.0\sbin>jps
16964 ResourceManager
24964 Jps
16984 NameNode
19832 NodeManager
27192 DataNode
```

STEP 2: CREATE PROJECT FOLDER

```
mkdir C:\HadoopWork
```

```
mkdir C:\HadoopWork\src
```

```
cd C:\HadoopWork\src
```

STEP 3: WRITE INPUT FILE

Create input.txt inside C:\HadoopWork\src

```
0,0,10
```

```
0,1,20
```

```
1,0,30
```

```
1,1,40
```

```
0,0,5
```

STEP 4: WRITE JAVA PROGRAM

Create MatrixMultiplication.java in the **same folder**.

```
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class MatrixMultiplication {


    public static class MatrixMapper extends Mapper<LongWritable, Text, Text, Text> {

        @Override

        protected void map(LongWritable key, Text value, Context context) throws IOException,
        InterruptedException {

            String[] elements = value.toString().split(",");

            if (elements.length == 3) {

                context.write(new Text(elements[0] + "," + elements[1]), new Text(elements[2]));

            }

        }

    }


    public static class MatrixReducer extends Reducer<Text, Text, Text, IntWritable> {

        @Override

        protected void reduce(Text key, Iterable<Text> values, Context context) throws IOException,
        InterruptedException {

            int sum = 0;
```

```

        for (Text value : values) {
            sum += Integer.parseInt(value.toString());
        }
        context.write(key, new IntWritable(sum));
    }
}

```

```

public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "Matrix Summation");

    // CRITICAL FIX FOR WINDOWS:
    job.setJar("matrix.jar");

    job.setMapperClass(MatrixMapper.class);
    job.setReducerClass(MatrixReducer.class);
    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(Text.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
}

```

STEP 5: CREATE JAR FILE

jar cf matrix.jar MatrixMultiplication*.class

Check:

dir

You must see:

matrix.jar

STEP6:COMPILE

```
javac -classpath "C:\hadoop-3.3.0\share\hadoop\common\*;C:\hadoop-3.3.0\share\hadoop\common\lib\*;C:\hadoop-3.3.0\share\hadoop\mapreduce\*;C:\hadoop-3.3.0\share\hadoop\mapreduce\lib\*" MatrixMultiplication.java
```

STEP 7: CREATE HDFS INPUT DIRECTORY

```
hdfs dfs -mkdir /matrix_input
```

```
hdfs dfs -put input.txt /matrix_input
```

```
hdfs dfs -ls /matrix_input
```

STEP 8: RUN MAPREDUCE JOB

```
hadoop jar matrix.jar MatrixMultiplication /matrix_input /matrix_output
```

✓ If you see **Running job:** → SUCCESS

```
C:\HadoopWork\src>hadoop jar matrix.jar MatrixMultiplication /matrix_input /matrix_output
2026-01-21 06:16:10,498 INFO client.DefaultHadoopFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2026-01-21 06:16:11,257 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2026-01-21 06:16:11,283 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/bavana/.staging/job_1768956245873_0001
2026-01-21 06:16:12,229 INFO input.FileInputFormat: Total input files to process : 1
2026-01-21 06:16:12,314 INFO mapreduce.JobSubmitter: number of splits:1
2026-01-21 06:16:12,532 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1768956245873_0001
2026-01-21 06:16:12,532 INFO mapreduce.JobSubmitter: Executing with tokens: []
2026-01-21 06:16:12,776 INFO conf.Configuration: resource-types.xml not found
2026-01-21 06:16:12,776 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2026-01-21 06:16:13,344 INFO impl.YarnClientImpl: Submitted application application_1768956245873_0001
2026-01-21 06:16:13,440 INFO mapreduce.Job: The url to track the job: https://BAVANA:8088/proxy/application_1768956245873_0001/
2026-01-21 06:16:13,443 INFO mapreduce.Job: Running job: job_1768956245873_0001
2026-01-21 06:16:28,802 INFO mapreduce.Job: Job job_1768956245873_0001 running in uber mode : false
2026-01-21 06:16:28,804 INFO mapreduce.Job:  map 0% reduce 0%
2026-01-21 06:16:36,959 INFO mapreduce.Job:  map 100% reduce 0%
2026-01-21 06:16:45,098 INFO mapreduce.Job:  map 100% reduce 100%
2026-01-21 06:16:46,153 INFO mapreduce.Job: Job job_1768956245873_0001 completed successfully
2026-01-21 06:16:46,278 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=50
    FILE: Number of bytes written=530299
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=148
    HDFS: Number of bytes written=28
    HDFS: Number of read operations=8
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
    HDFS: Number of bytes read erasure-coded=0
  Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=5399
    Total time spent by all reduces in occupied slots (ms)=6196
    Total time spent by all map tasks (ms)=5399
    Total time spent by all reduce tasks (ms)=6196
    Total vcore-milliseconds taken by all map tasks=5399
    Total vcore-milliseconds taken by all reduce tasks=6196
    Total megabyte-milliseconds taken by all map tasks=5528576
    Total megabyte-milliseconds taken by all reduce tasks=6344704
  Map-Reduce Framework
    Map input records=5
    Map output records=5
    Map output bytes=34
```

STEP 9: VIEW OUTPUT

```
hdfs dfs -ls /matrix_output
```

```
hdfs dfs -cat /matrix_output/part-r-00000
```

FINAL OUTPUT (stored in HDFS)

0,0 15

0,1 20

1,0 30

1,1 40