Kaustubh Shrikant Kabra

ERP Number :- 38

TE Comp 1

Logs File Analysis Hadoop

Code:

1> <u>LogFileMapper.java</u> (Use for mapping the IP addresses from input csv file)

```
package LogFileCountry;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.*;


public class LogFileMapper extends MapReduceBase implements Mapper<LongWritable,
Text, Text, IntWritable> {

	private final static IntWritable one = new IntWritable(1);


	public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable>
output, Reporter reporter) throws IOException {


		String valueString = value.toString();

		String[] SingleIpData = valueString.split("-");

		output.collect(new Text(SingleIpData[0]), one);

	}

}
```

2><u>LogFileReduce.java</u> (Use for reducing data received from mapper process to final output)

```
package LogFileCountry;

import java.io.IOException;

import java.util.*;


import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.*;


public class LogFileReducer extends MapReduceBase implements Reducer<Text, IntWritable,
Text, IntWritable> {

	public void reduce(Text t_key, Iterator<IntWritable> values,
OutputCollector<Text,IntWritable> output, Reporter reporter) throws IOException {

		Text key = t_key;

		int frequencyForIp = 0;

		while (values.hasNext()) {

			// replace type of value with the actual type of our value

			IntWritable value = (IntWritable) values.next();

			frequencyForIp += value.get();

		}

		output.collect(key, new IntWritable(frequencyForIp));

	}

}
```

3>**<u>LogFileCountryDriver.java</u>** **(**The driver code to run map-reduce on hdfs**)**

```
package LogFileCountry;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.*;

import org.apache.hadoop.mapred.*;


public class LogFileCountryDriver {
```

```
public static void main(String[] args) {
        JobClient my_client = new JobClient();
        // Create a configuration object for the job
        JobConf job_conf = new JobConf(LogFileCountryDriver.class);

        // Set a name of the Job
        job_conf.setJobName("LogFileIP");
        // Specify data type of output key and value
        job_conf.setOutputKeyClass(Text.class);
        job_conf.setOutputValueClass(IntWritable.class);
        // Specify names of Mapper and Reducer Class
        job_conf.setMapperClass(LogFileCountry.LogFileMapper.class);
        job_conf.setReducerClass(LogFileCountry.LogFileReducer.class);

        // Specify formats of the data type of Input and output
        job_conf.setInputFormat(TextInputFormat.class);
        job_conf.setOutputFormat(TextOutputFormat.class);

        // Set input and output directories using command line arguments,
        //arg[0] = name of input directory on HDFS, and arg[1] =  name of output
directory to be created to store the output file.

        FileInputFormat.setInputPaths(job_conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(job_conf, new Path(args[1]));

        my_client.setConf(job_conf);
        try {   // Run the job
                JobClient.runJob(job_conf);
        } catch (Exception e) {
                e.printStackTrace();
```

```
            }
        }
}
```

4>log_file.txt (Input file sample)

0.223.157.186 - - [15/Jul/2009:20:50:32 -0700] "GET /assets/js/the-associates.js HTTP/1.1" 304 -

10.223.157.186 - - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/home-logo.png HTTP/1.1" 304 -

10.223.157.186 - - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/dummy/primary-news-2.jpg HTTP/1.1" 304 -

10.223.157.186 - - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/dummy/primary-news-1.jpg HTTP1.1" 304 -

10.223.157.186 - - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/home-media-block-placeholder.jpg HTTP/1.1" 304 -

10.223.157.186 - - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/dummy/secondary-news-4.jpg HTTP/1.1" 304 -

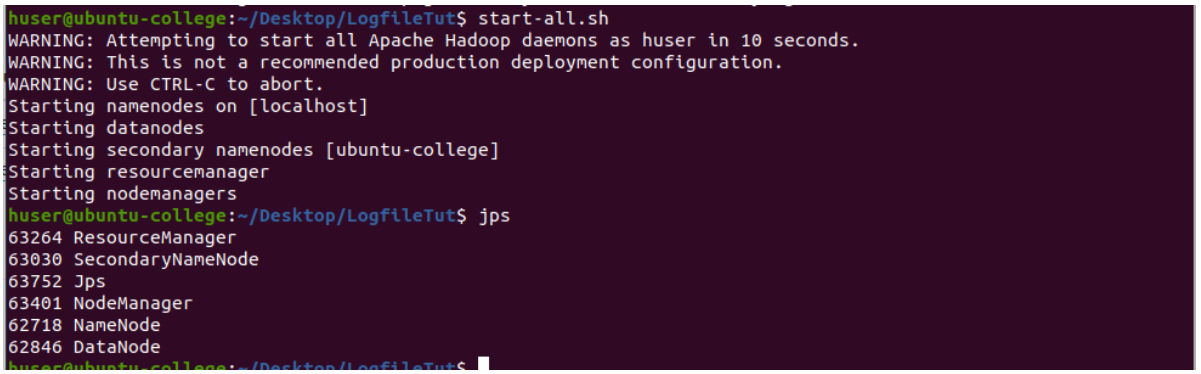
10.223.157.186 - - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/loading.gif HTTP/1.1" 304 -

10.223.157.186 - - [15/Jul/2009:20:50:33 -0700] "GET /assets/img/search-button.gif HTTP/1.1" 304 –

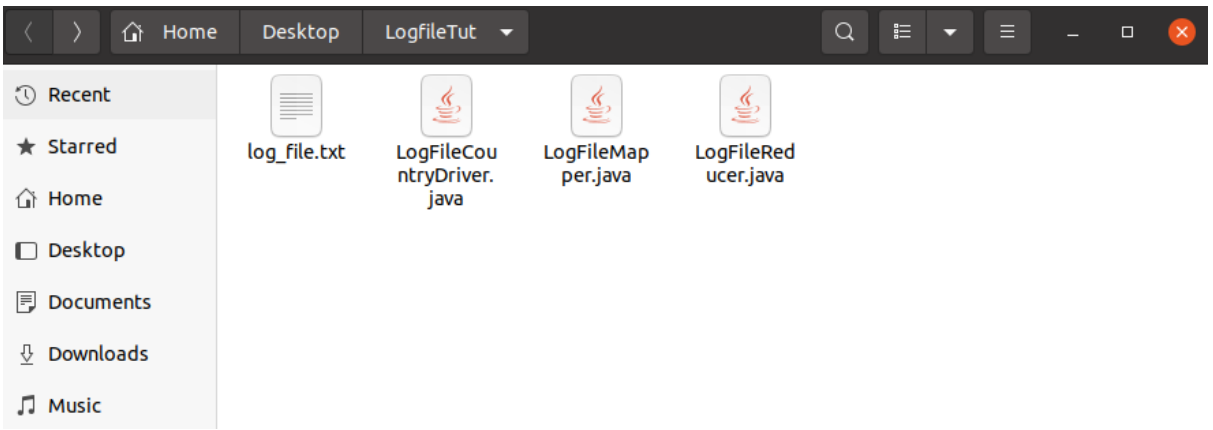
Step For Logs File Code:

1) Starting Hadoop and check if it is started.

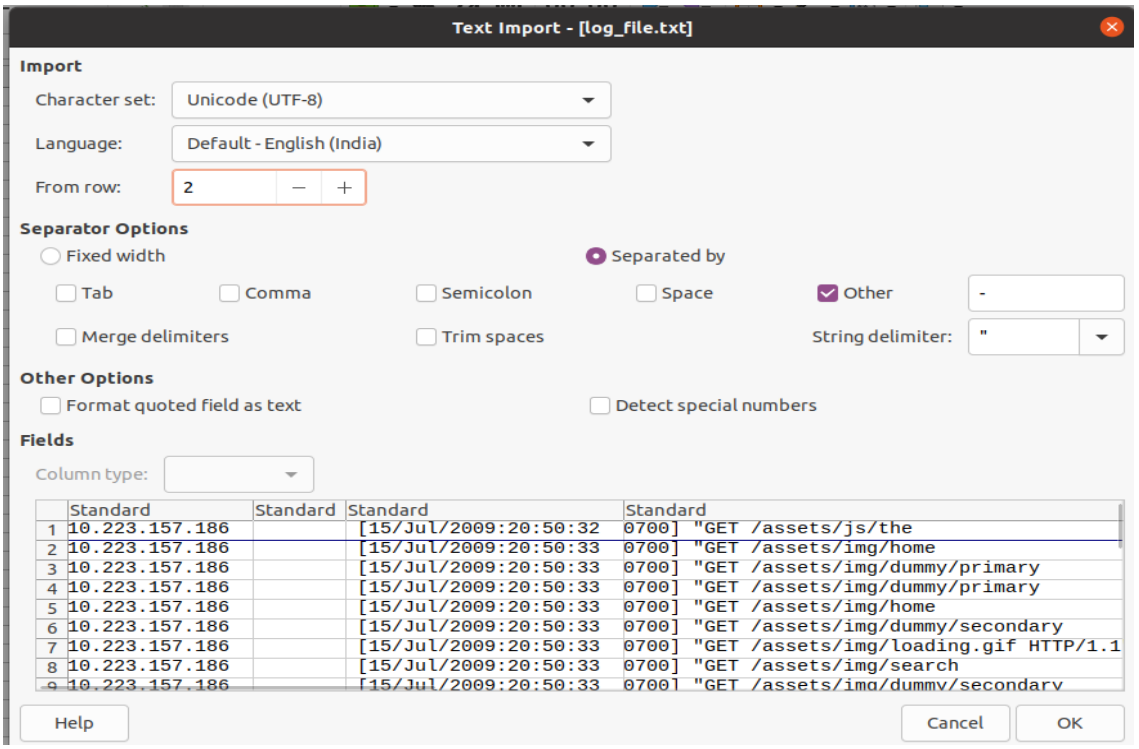
   start-all.sh

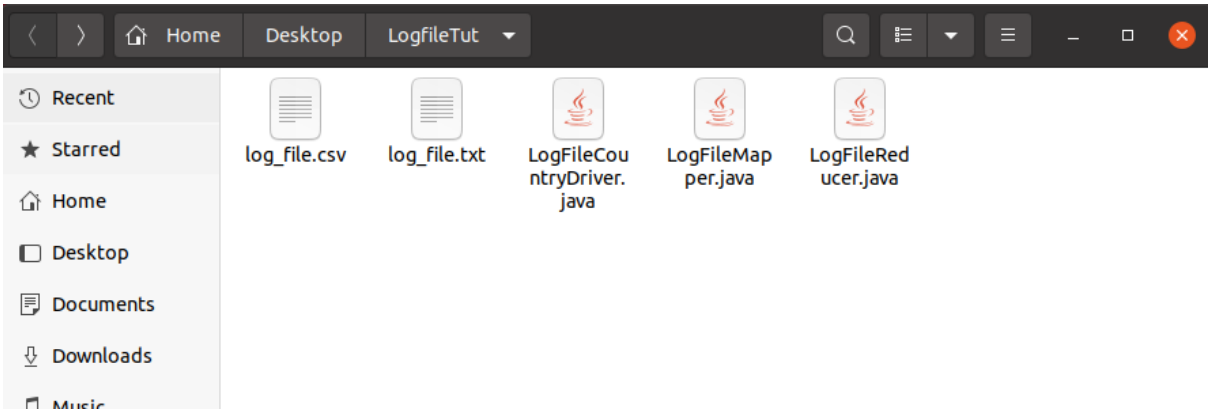


2) Create folder "LogFileTut". Copy the log_file.txt given and create the java files.
   i) LogFileMapper.java
   ii) LogFileReducer.java
   iii) LogFileCountryDriver.java



3) Convert the log_file.txt to .csv file. Open LibreOffice Calc-> Open -> log_file.txt. Save As .csv in the LogFileTut folder.





4) Give Read permission to all the files in directories.

sudo chmod +r *.*

5) Set HADOOP_CLASSPATH environment variable.

export HADOOP_CLASSPATH=$(hadoop classpath) or

export CLASSPATH= "$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-core-3.2.2.jar: $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client-common-3.2.2.jar: $HADOOP_HOME/share/hadoop/common/hadoop-common-3.2.2.jar: $HADOOP_HOME/lib/*: ~/home/huser/Desktop/LogFileTut/*"

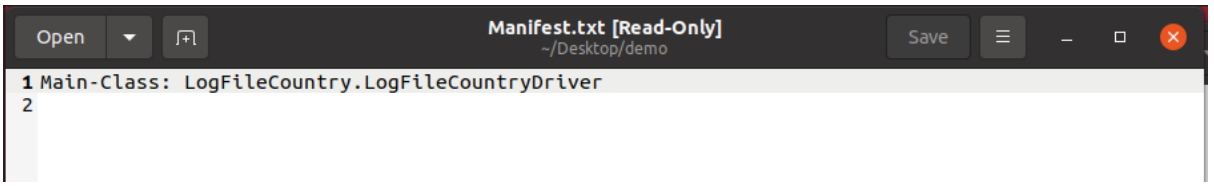
6) Compile the java code:

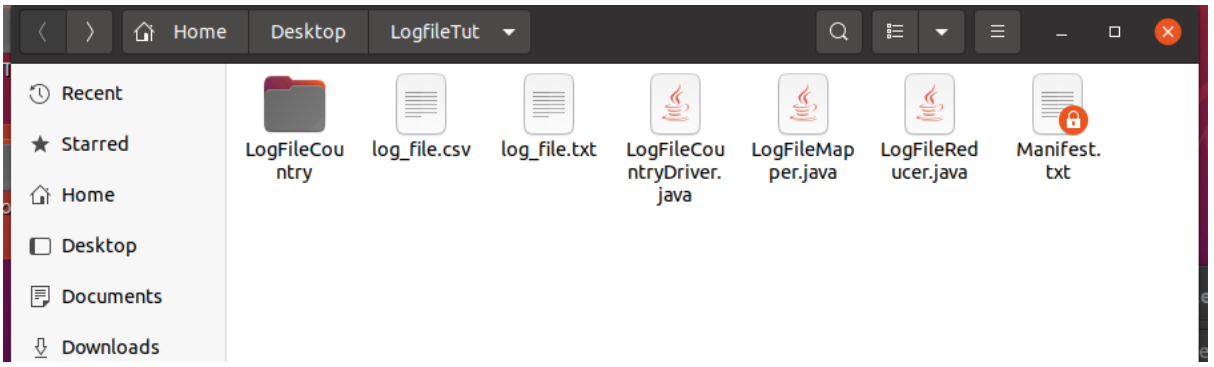
javac -classpath $(HADOOP_CLASSPATH) -d . <java file (3 files)>

7) Create Manifest.txt file.

```
huser@ubuntu-college:~/Desktop/LogfileTut$ javac -d '/home/huser/Desktop/LogfileTut/exp_classfile' LogFileMapper.java LogFileReducer.java LogFileCountryDriver.java
huser@ubuntu-college:~/Desktop/LogfileTut$ sudo gedit Manifest.txt
[sudo] password for huser:

(gedit:59507): Tepl-WARNING **: 22:24:16.402: GVfs metadata is not supported. Fallback to TeplMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not supported on this platform. In the latter case, you should configure Tepl with --disable-gvfs-metadata.
```

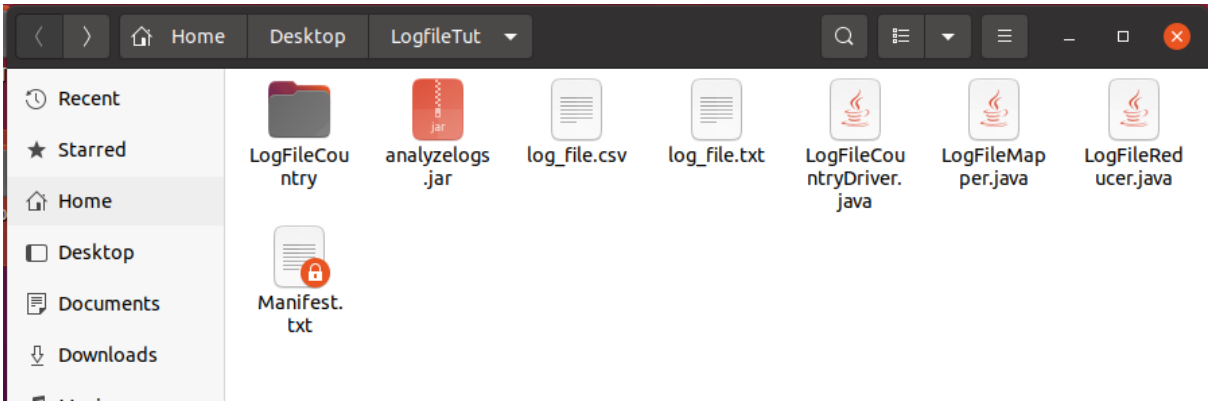




8) Creation .jar file of classes:

jar -cvfm <jar file name > Manifest.txt <classes folder/*.class>

```
huser@ubuntu-college:~/Desktop/LogfileTut$ jar -cvfm analyzelogs.jar Manifest.txt LogFileCountry/*.class
added manifest
adding: LogFileCountry/LogFileCountryDriver.class(in = 1677) (out= 825)(deflated 50%)
adding: LogFileCountry/LogFileMapper.class(in = 1713) (out= 645)(deflated 62%)
adding: LogFileCountry/LogFileReducer.class(in = 1580) (out= 635)(deflated 59%)
```

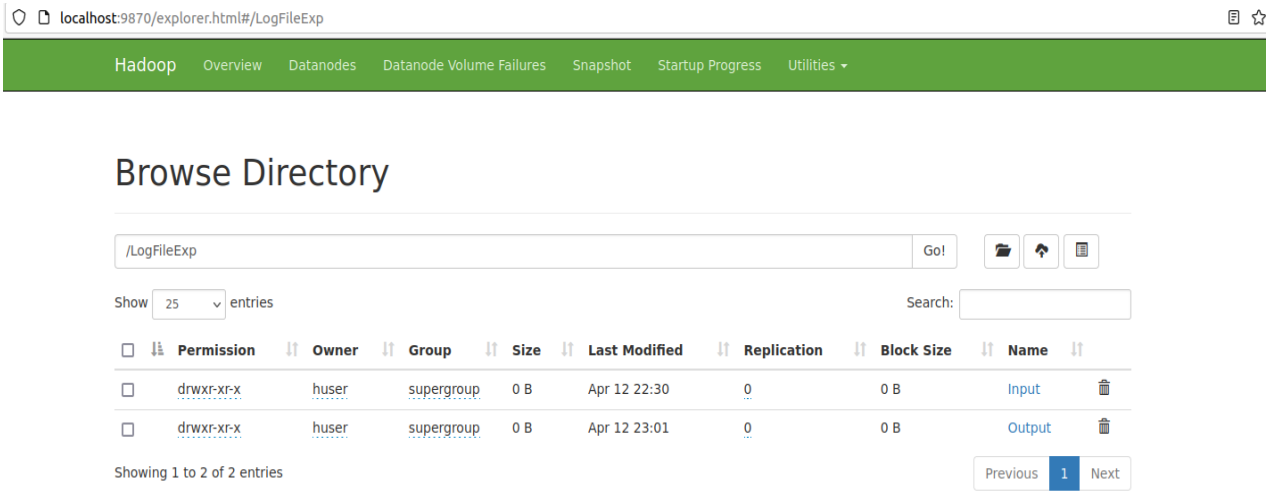


9) Create a directory on HDFS .And check on localhost:9870

hdfs dfs -mkdir / LogFileExp

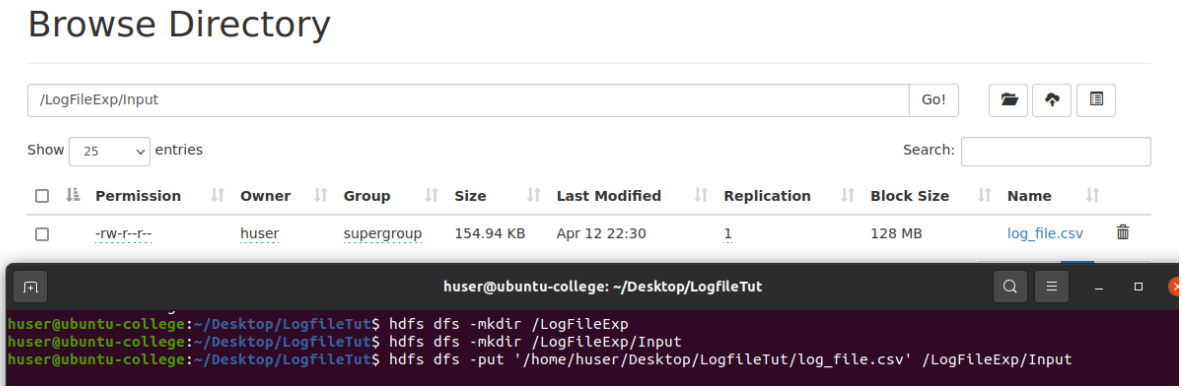
hdfs dfs -mkdir / LogFileExp/Input

hdfs dfs -mkdir / LogFileExp/Output



10) Upload the log_file.csv in hadoop dir /LogFileExp/Input

hdfs dfs -put <Input file > <hdfs input dir>



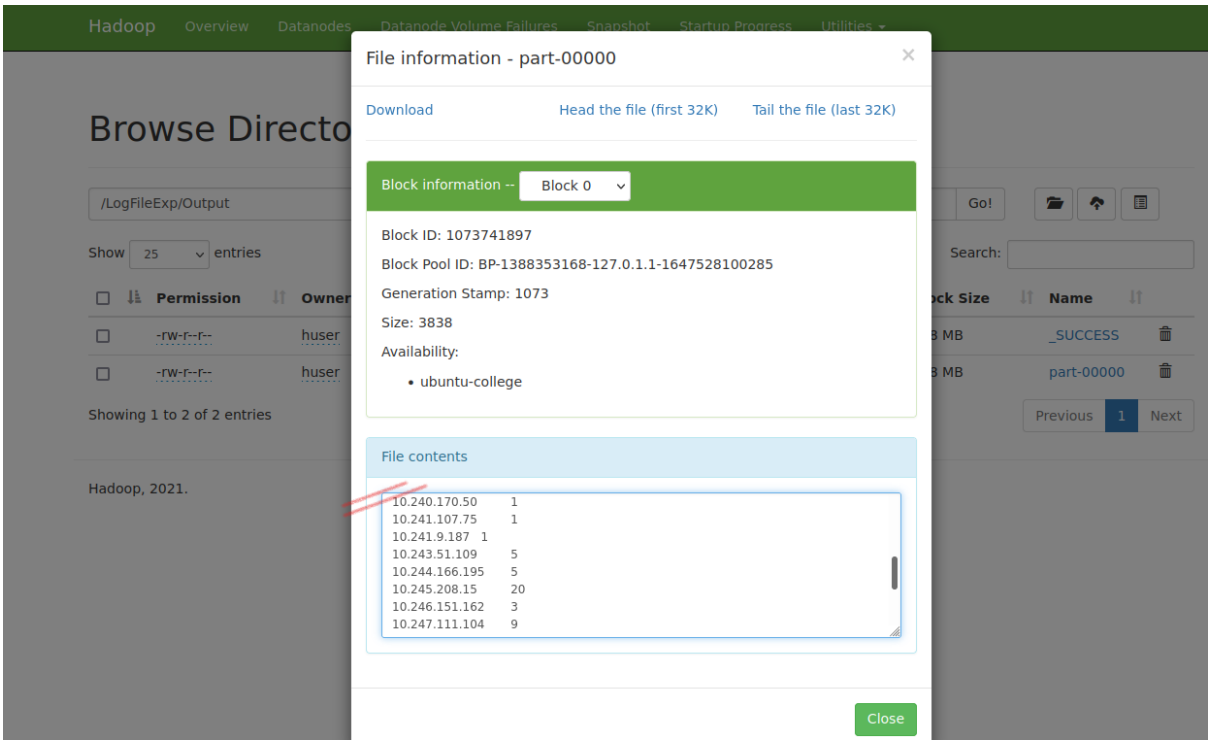
11) Running the jar file on Hadoop.

hadoop jar <jar file> <class name> <hdfs input dir> <hdfs output dir>

```
huser@ubuntu-college:~/Desktop/LogfileTut$ hadoop jar analyzelogs.jar /LogFileExp/Input /LogFileExp/Output
2022-04-12 22:51:25,988 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2022-04-12 22:51:27,403 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2022-04-12 22:51:35,208 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement th
e Tool interface and execute your application with ToolRunner to remedy this.
2022-04-12 22:51:36,276 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hu
ser/.staging/job_1649777619248_0001
2022-04-12 22:51:39,085 INFO mapred.FileInputFormat: Total input files to process : 1
2022-04-12 22:51:40,281 INFO mapreduce.JobSubmitter: number of splits:2
2022-04-12 22:51:40,821 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1649777619248_0001
2022-04-12 22:51:40,823 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-04-12 22:51:42,337 INFO conf.Configuration: resource-types.xml not found
2022-04-12 22:51:42,337 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2022-04-12 22:52:55,956 INFO impl.YarnClientImpl: Submitted application application_1649777619248_0001
2022-04-12 22:52:58,540 INFO mapreduce.Job: The url to track the job: http://ubuntu-college:8088/proxy/application_1649777
619248_0001/
2022-04-12 22:52:58,584 INFO mapreduce.Job: Running job: job_1649777619248_0001
2022-04-12 22:57:02,946 INFO mapreduce.Job: Job job_1649777619248_0001 running in uber mode : false
2022-04-12 22:57:03,272 INFO mapreduce.Job:  map 0% reduce 0%
2022-04-12 23:00:09,974 INFO mapreduce.Job:  map 83% reduce 0%
2022-04-12 23:00:27,106 INFO mapreduce.Job:  map 100% reduce 0%
2022-04-12 23:01:05,301 INFO mapreduce.Job:  map 100% reduce 100%
2022-04-12 23:01:08,520 INFO mapreduce.Job: Job job_1649777619248_0001 completed successfully
2022-04-12 23:01:24,647 INFO mapreduce.Job: Counters: 54
```

12) Check the Output file.

```
huser@ubuntu-college:~/Desktop/LogfileTut$ hdfs dfs -cat /LogFileExp/Output/part-00000
10.1.1.236      7
10.1.181.142    14
10.1.232.31     5
10.10.55.142    14
10.102.101.66   1
10.103.184.104  1
10.103.190.81   53
10.103.63.29    1
10.104.73.51    1
10.105.160.183  1
10.108.91.151   1
10.109.21.76    1
10.11.131.40    1
10.111.71.20    8
```



13) Stop all processes :

stop-all.sh

```
huser@ubuntu-college:~/Desktop/LogfileTut$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as huser in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [ubuntu-college]
Stopping nodemanagers
Stopping resourcemanager
```