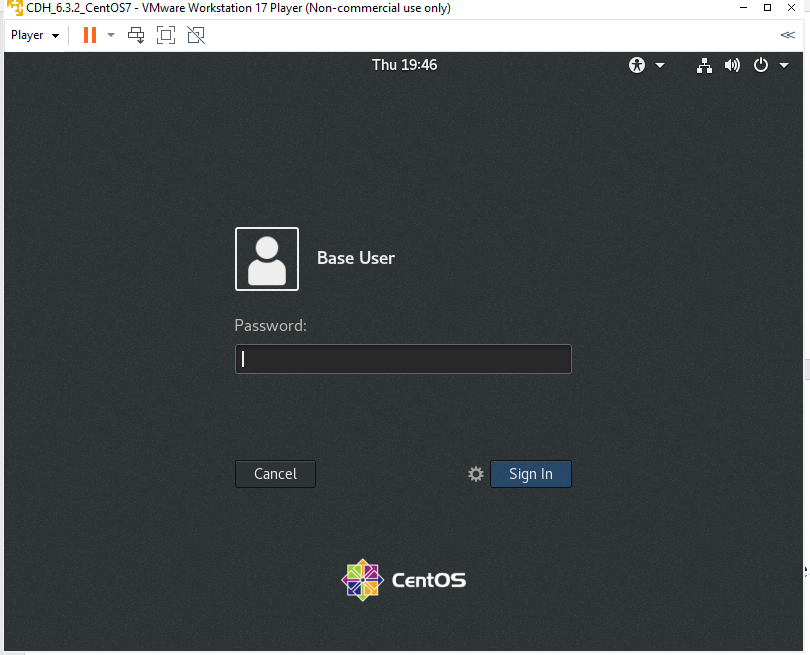
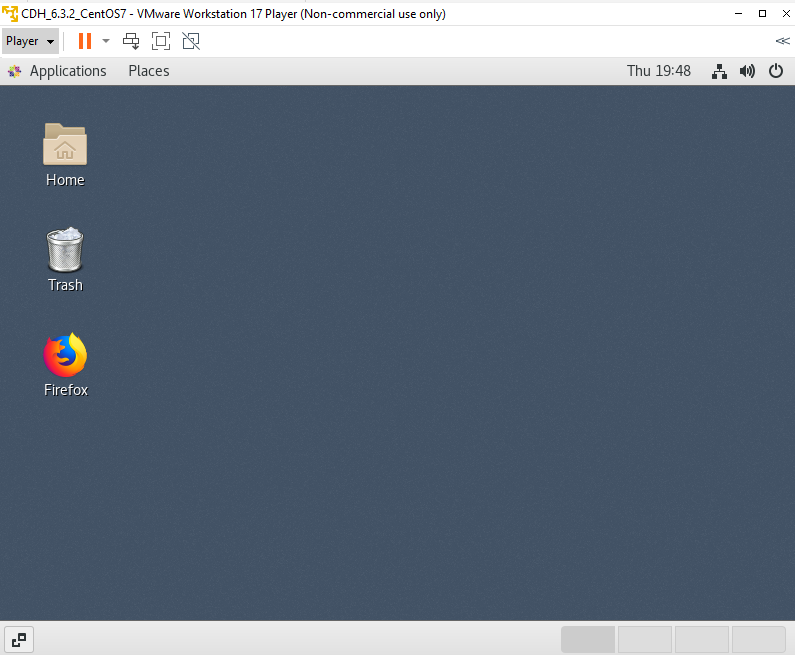
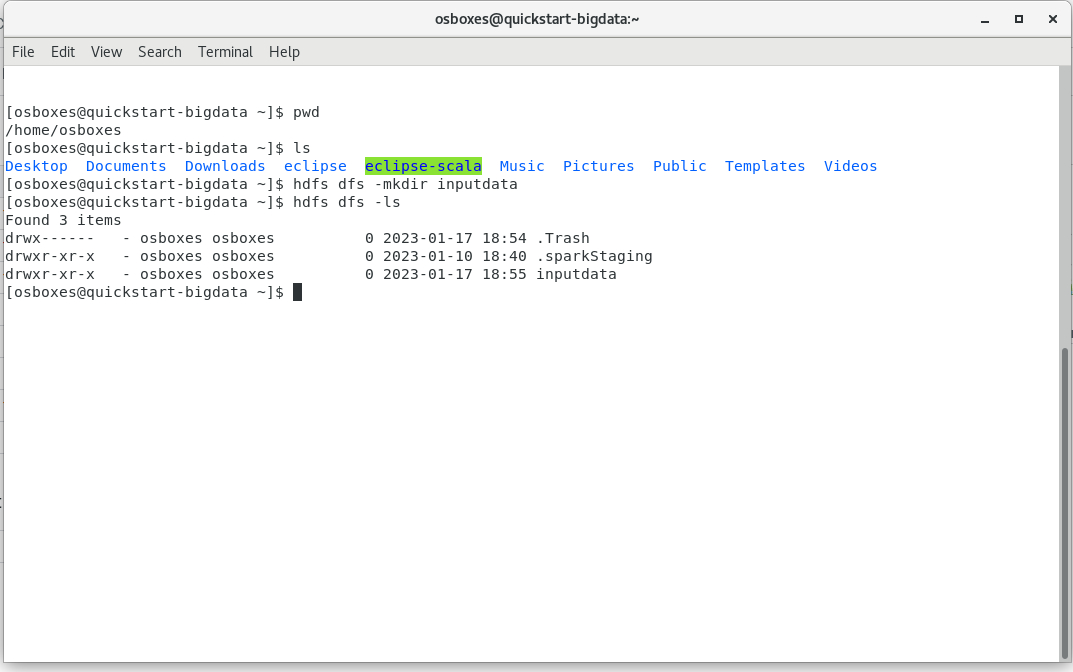
**Part A:**

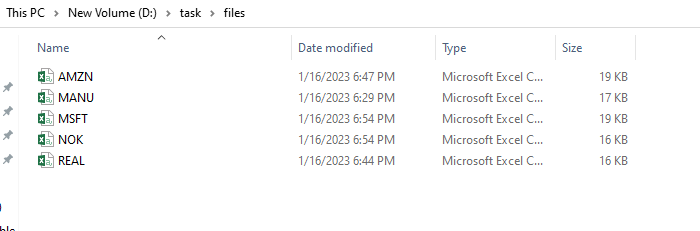




List all the files and directories under /user/osboxes in HDFS and Creating a directory with name inputdata in HDFS and List the folders within hdfs

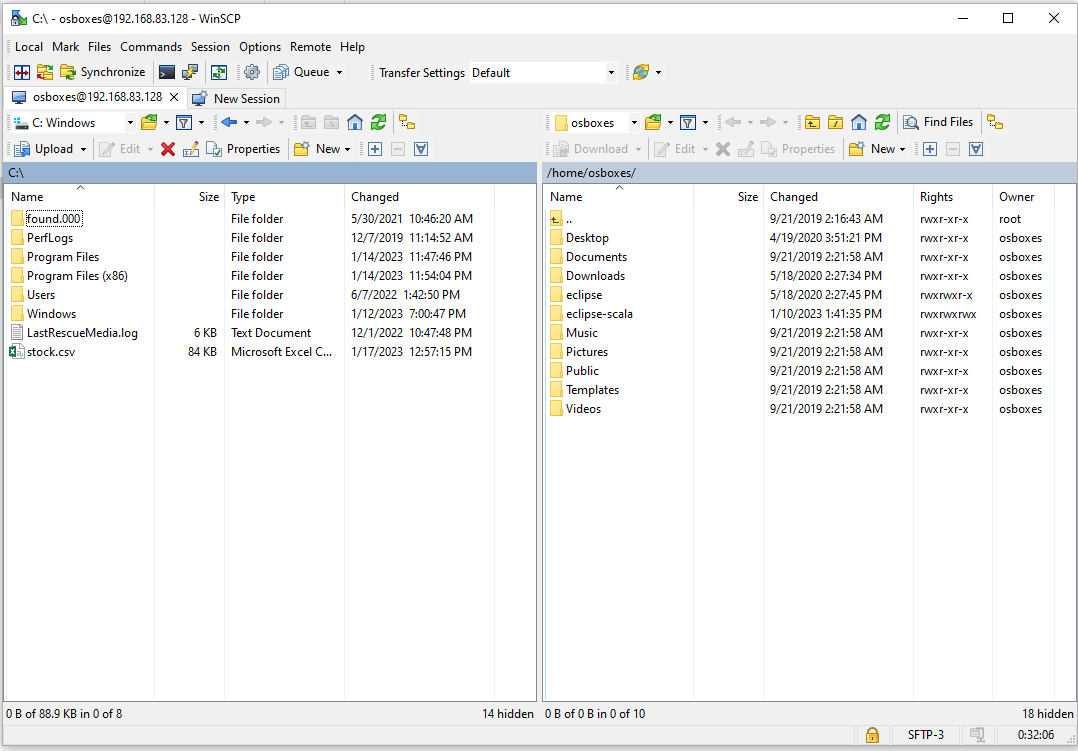


List of all Files that I downloaded and will work on it

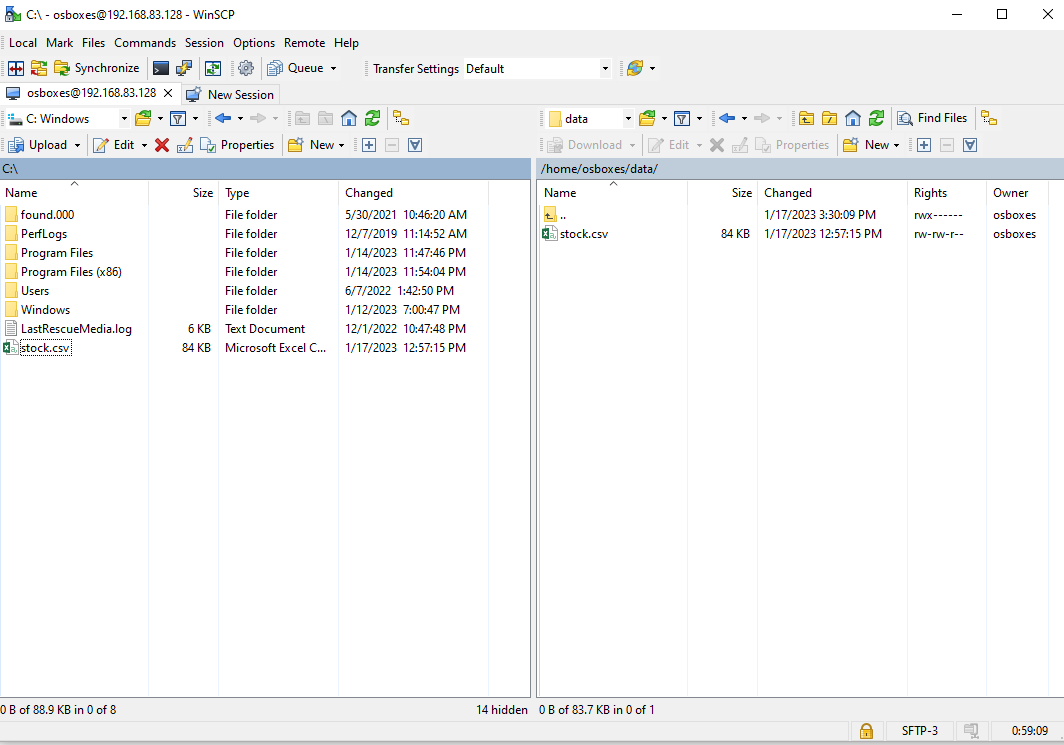


Added the column "Ticker" at the beginning of each file and concatenating them using python and pandas

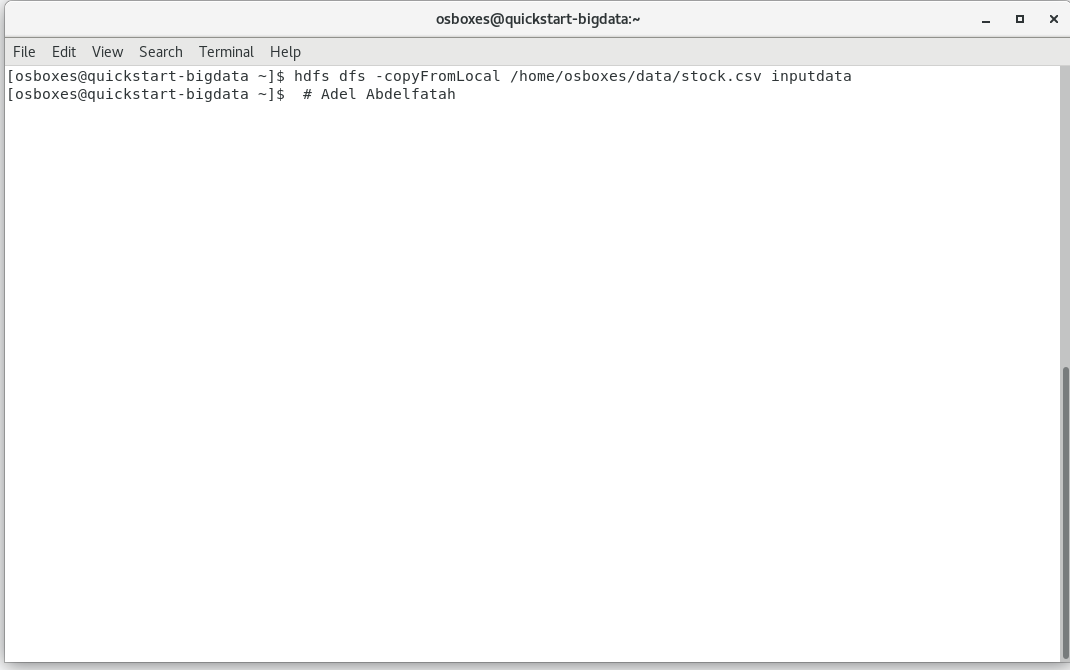
Use Winscp and connect to VM to transfer the stock.csv



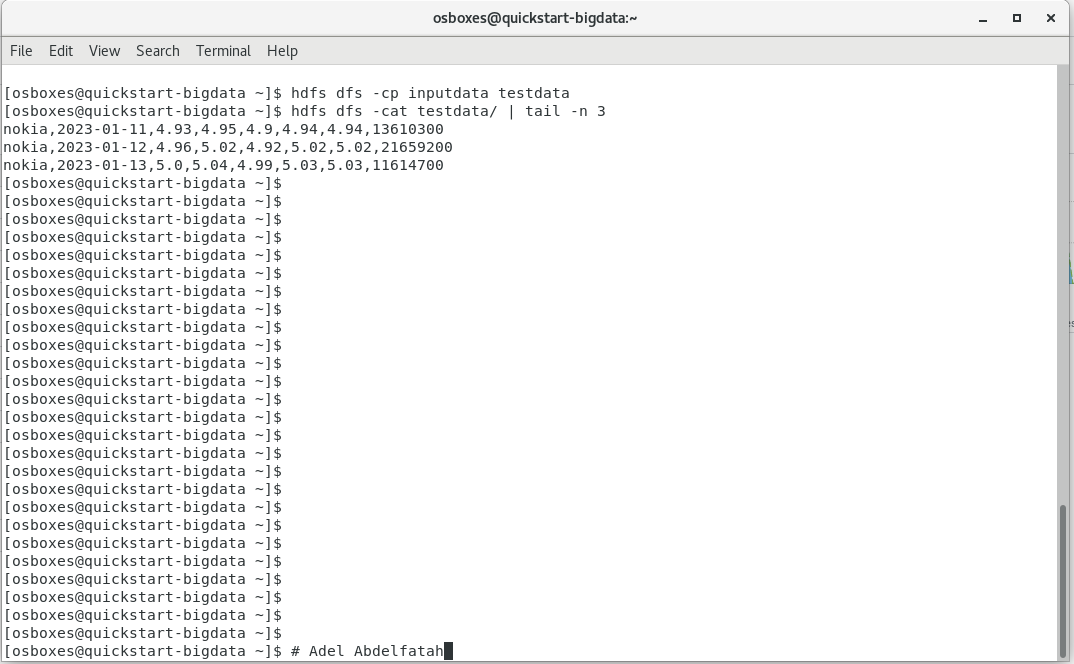
Transfer the file to /home/osboxes/data



Copy the file stock.csv from /home/osboxes/data (local file system) to 'inputdata' directory in HDFS that are created earlier



display the content of the first and last few records in the stock.csv file in HDFS



A copy of the stock.csv file as stock1.csv from the current hdfs location to another folder named 'testdata' in hdfs

**Part B:**

// minstockDriver.java

// Adel Abdelfatah ID: 20398047

**package** minstock;

**import** java.io.IOException;

**import** org.apache.hadoop.conf.Configuration;

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

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

//import org.apache.hadoop.io.IntWritable;

**import** org.apache.hadoop.io.FloatWritable;

**import** org.apache.hadoop.io.Text;

**import** org.apache.hadoop.mapreduce.Job;

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

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

**public** **class** minstockDriver {

**public** **static** **void** main(String[] args) **throws** IOException, ClassNotFoundException, InterruptedException {

Configuration conf = **new** Configuration();

Job job = **new** Job(conf);

job.setJarByClass(minstockDriver.**class**);

job.setMapperClass(minstockMapper.**class**);

job.setNumReduceTasks(1);

job.setReducerClass(minstockReducer.**class**);

//common

job.setOutputKeyClass(Text.**class**);

job.setOutputValueClass(FloatWritable.**class**);

FileInputFormat.*addInputPath*(job, **new** Path(args[0]));

FileOutputFormat.*setOutputPath*(job, **new** Path(args[1]));

//FileSystem fs = FileSystem.get(conf);

//fs.delete(new Path(args[1]));

job.waitForCompletion(**true**);

}

}

// minstockMapper.java

// Adel Abdelfatah ID: 20398047

package minstock;

import java.io.IOException;

import org.apache.hadoop.io.FloatWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class minstockMapper extends Mapper<LongWritable, Text, Text, FloatWritable>{

@Override

protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, FloatWritable>.Context context)

throws IOException, InterruptedException {

String inputLine = value.toString();

String[] Array = inputLine.split(",");

try {

context.write(new Text(Array[0]), new FloatWritable(Float.parseFloat(Array[5])));

}

catch (NumberFormatException e) {}

}

}

// minstockReducer.java

// Adel Abdelfatah ID: 20398047

package minstock;

import java.io.IOException;

import org.apache.hadoop.io.FloatWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class minstockReducer extends Reducer<Text, FloatWritable, Text, FloatWritable>{

@Override

protected void reduce(Text key, Iterable<FloatWritable> values,

Reducer<Text, FloatWritable, Text, FloatWritable>.Context context) throws IOException, InterruptedException {

float minp = Float.MAX\_VALUE;

for(FloatWritable value : values) {

minp = Math.min(minp, value.get());

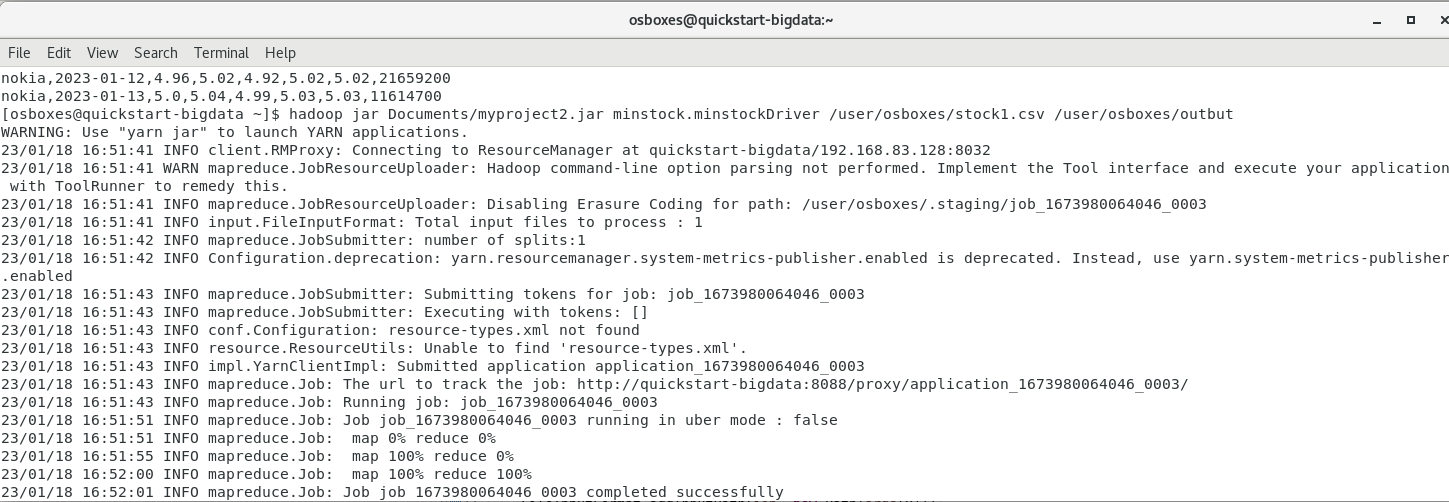
}

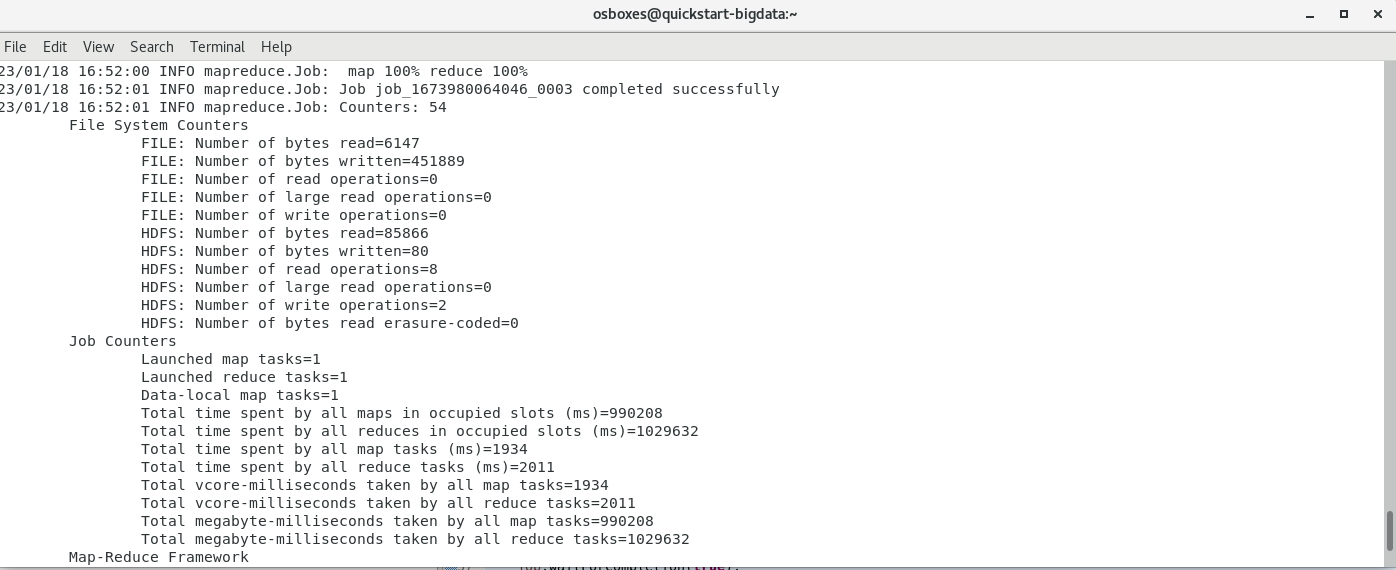
context.write(key, new FloatWritable(minp));

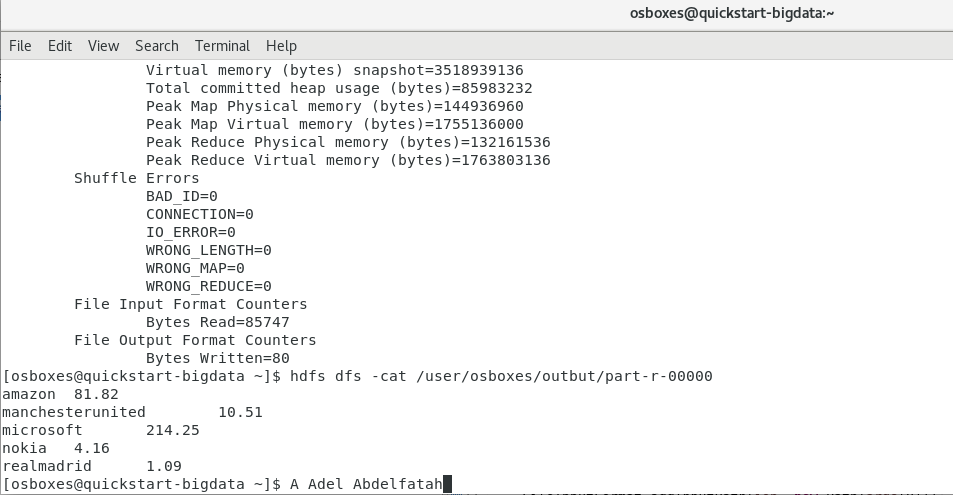
}

}

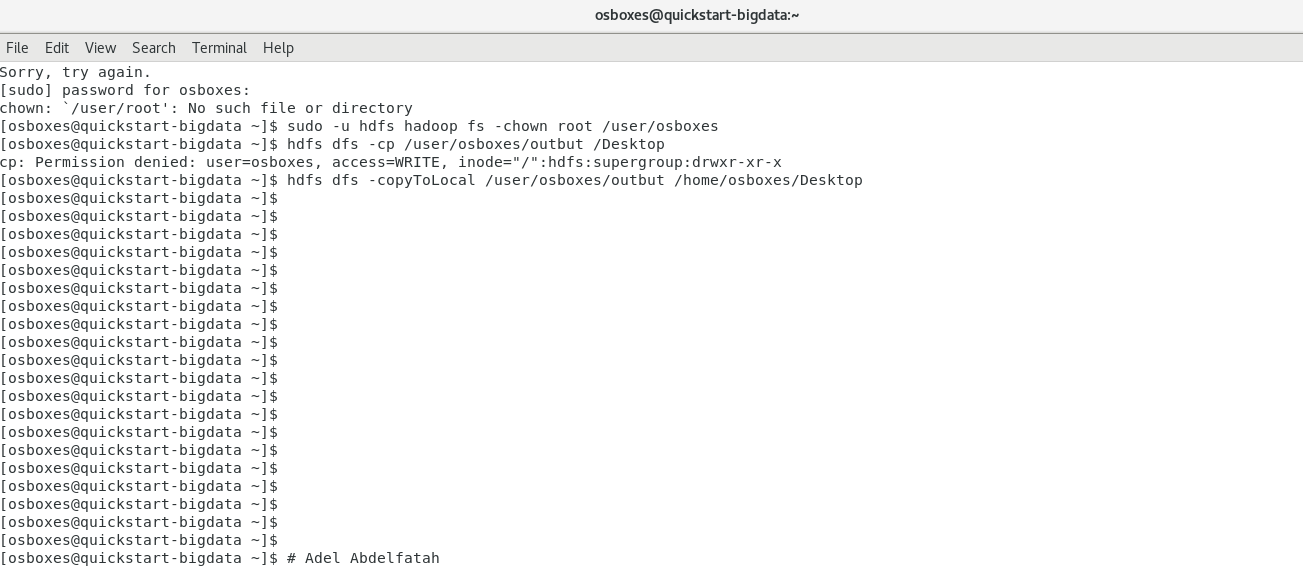
**Completed Successfully**

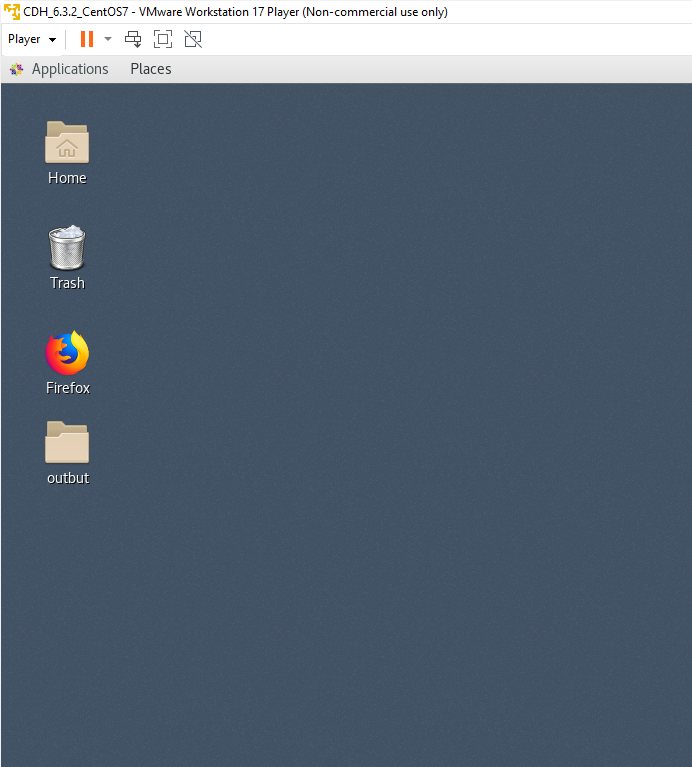




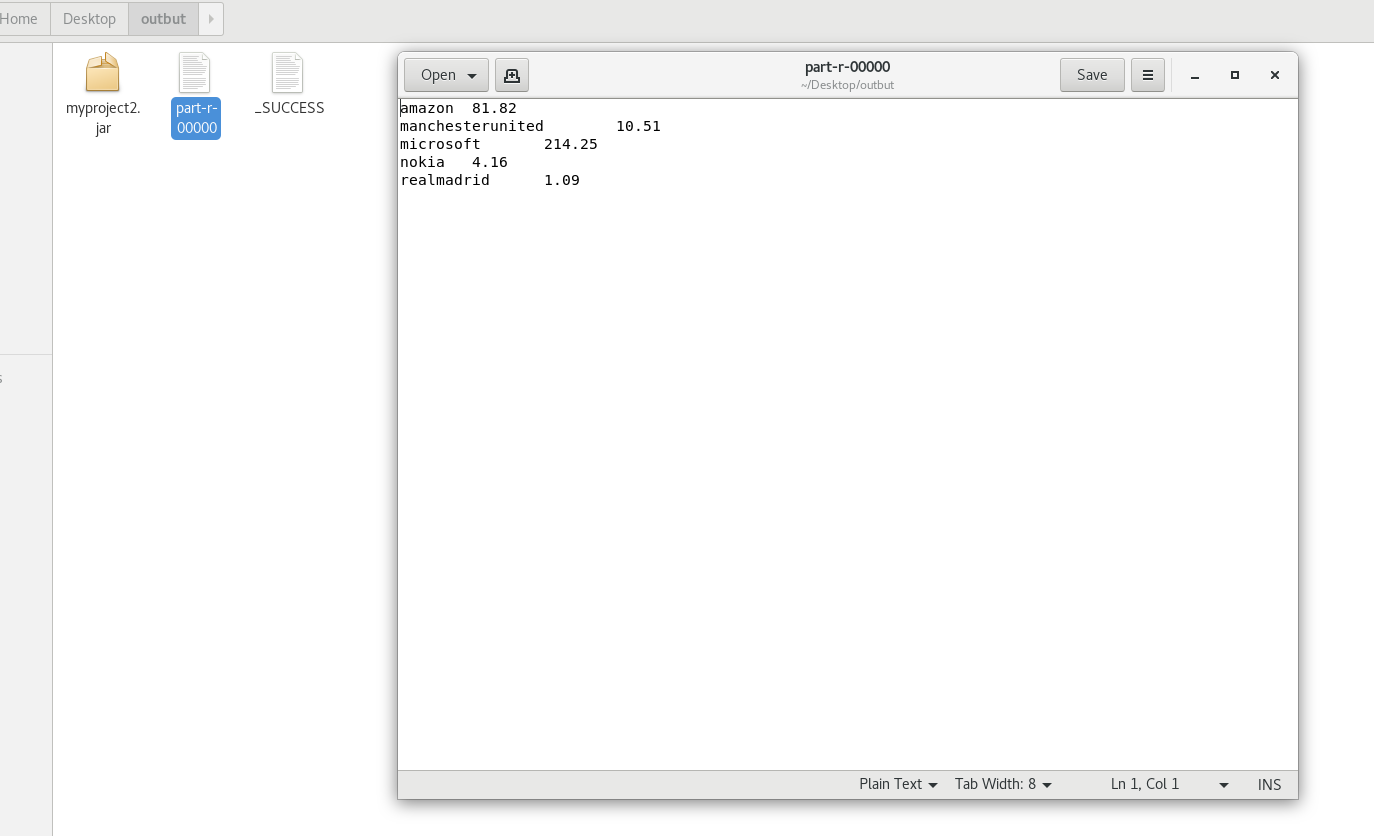
The O/P file:

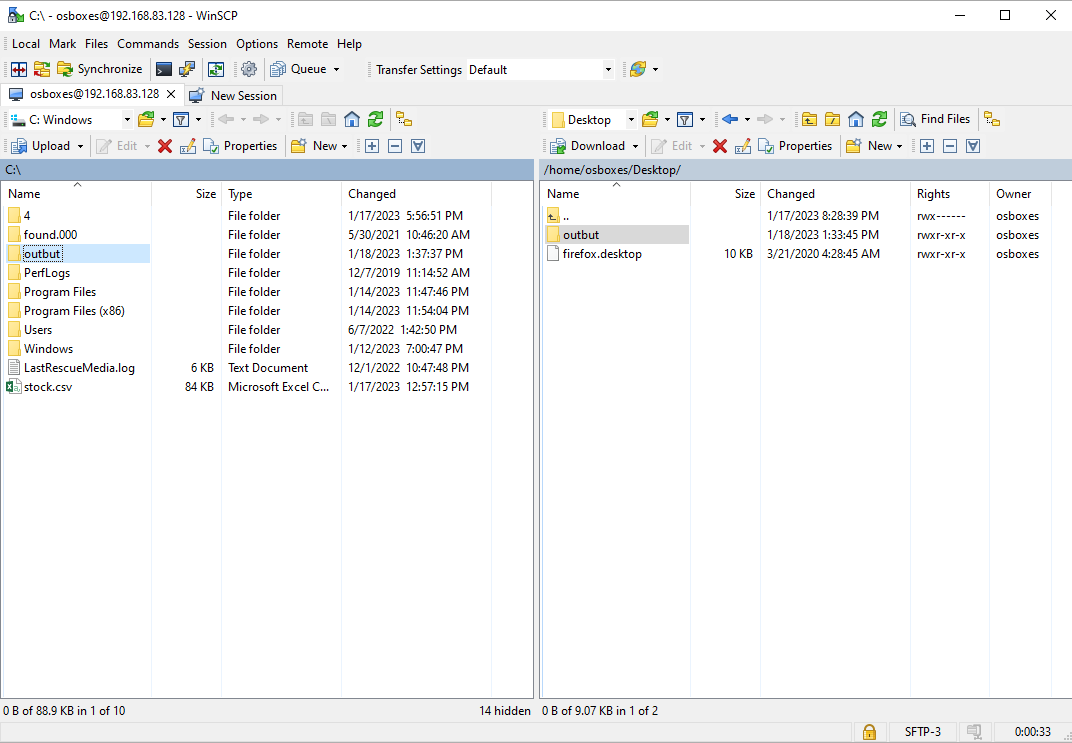
Copy My output file from distributed file system to VM local





I moved the jar file too to the output folder to move all of them to my machine using WinSCP



Move the file from the VM to my local Machine

The output:

amazon 81.82

manchesterunited 10.51

microsoft 214.25

nokia 4.16

realmadrid 1.09