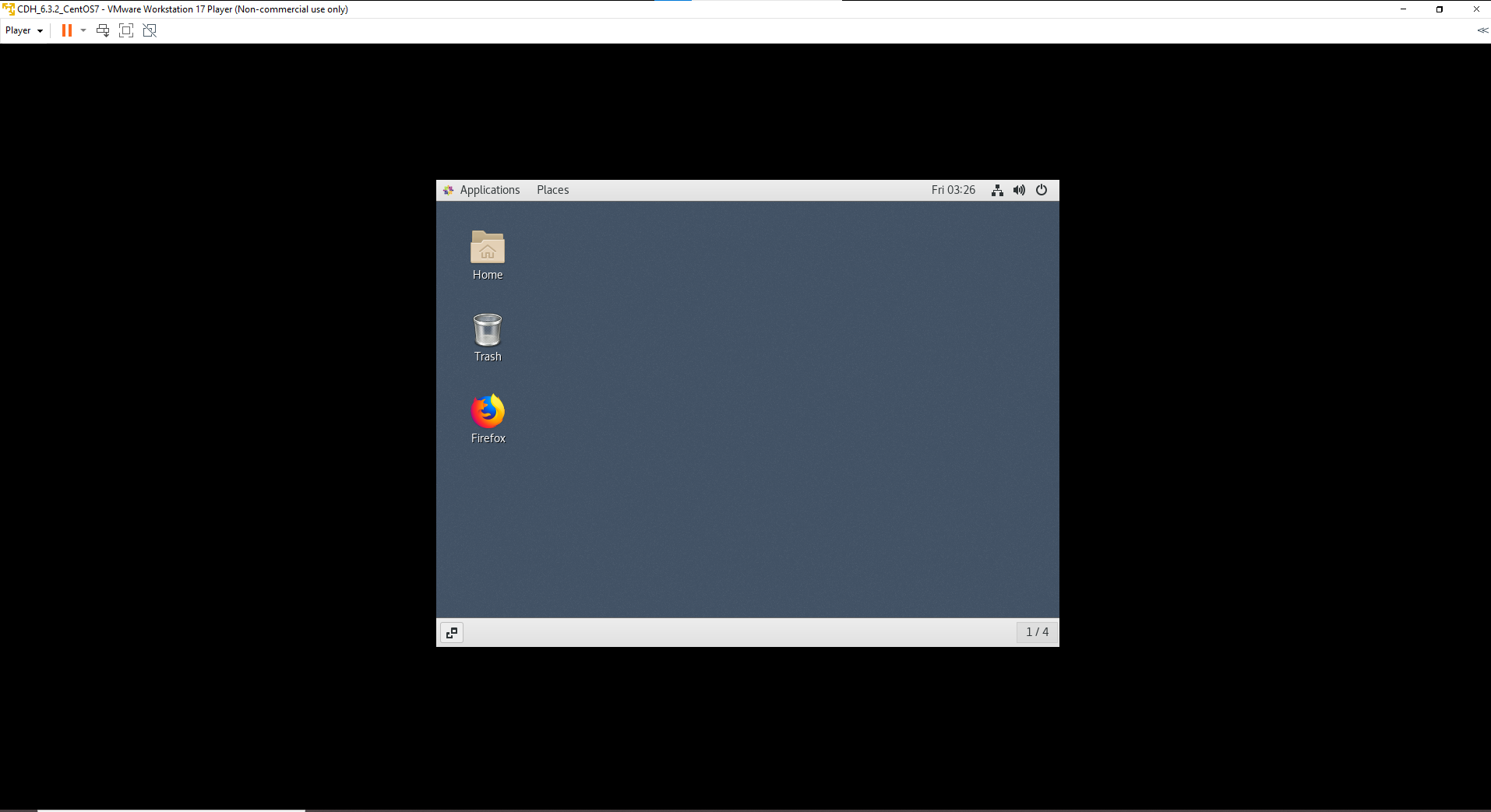
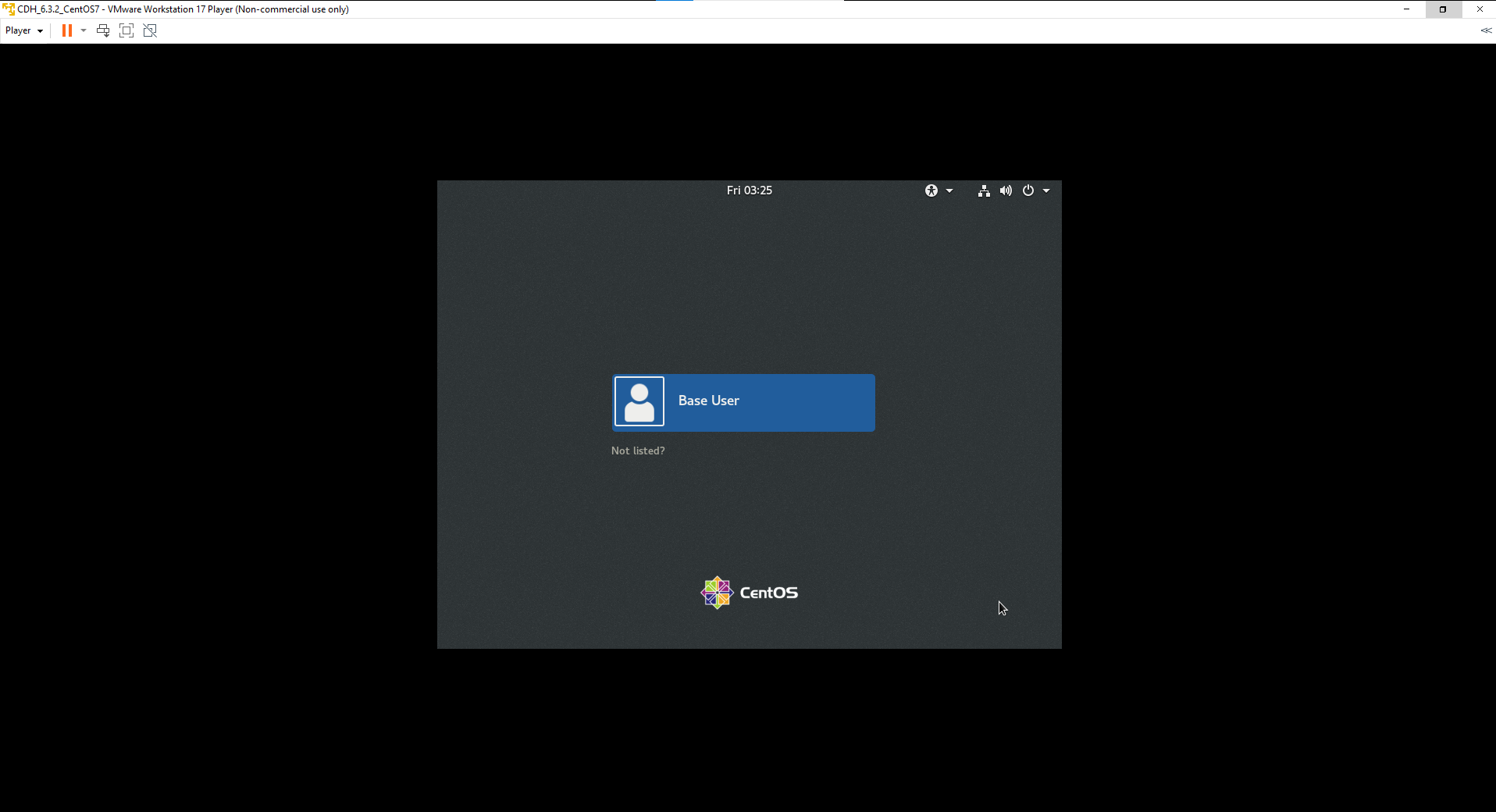
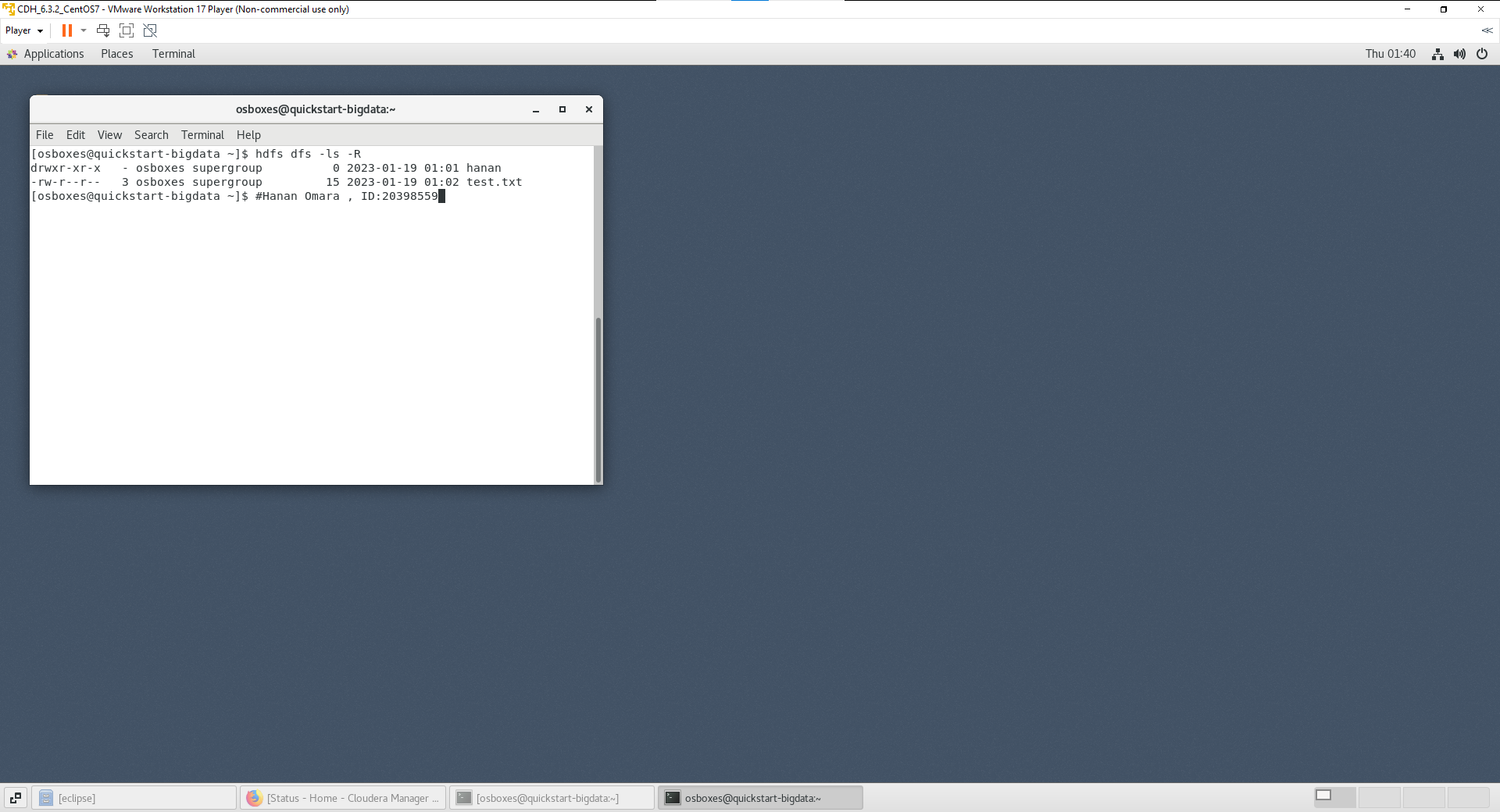
**Part A**

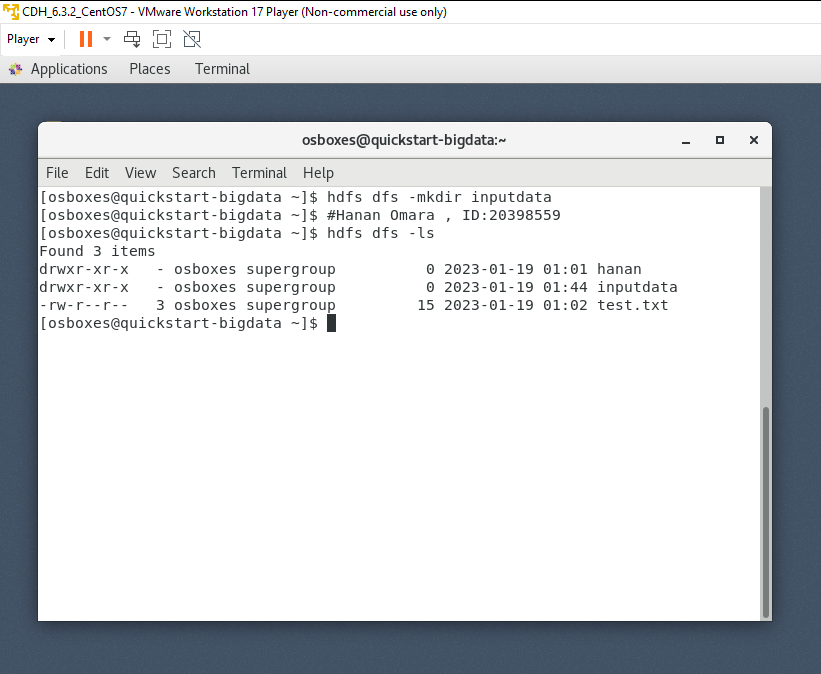
**Q1:**



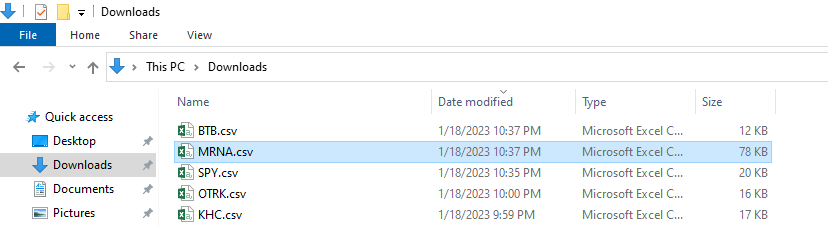
**Q2:**



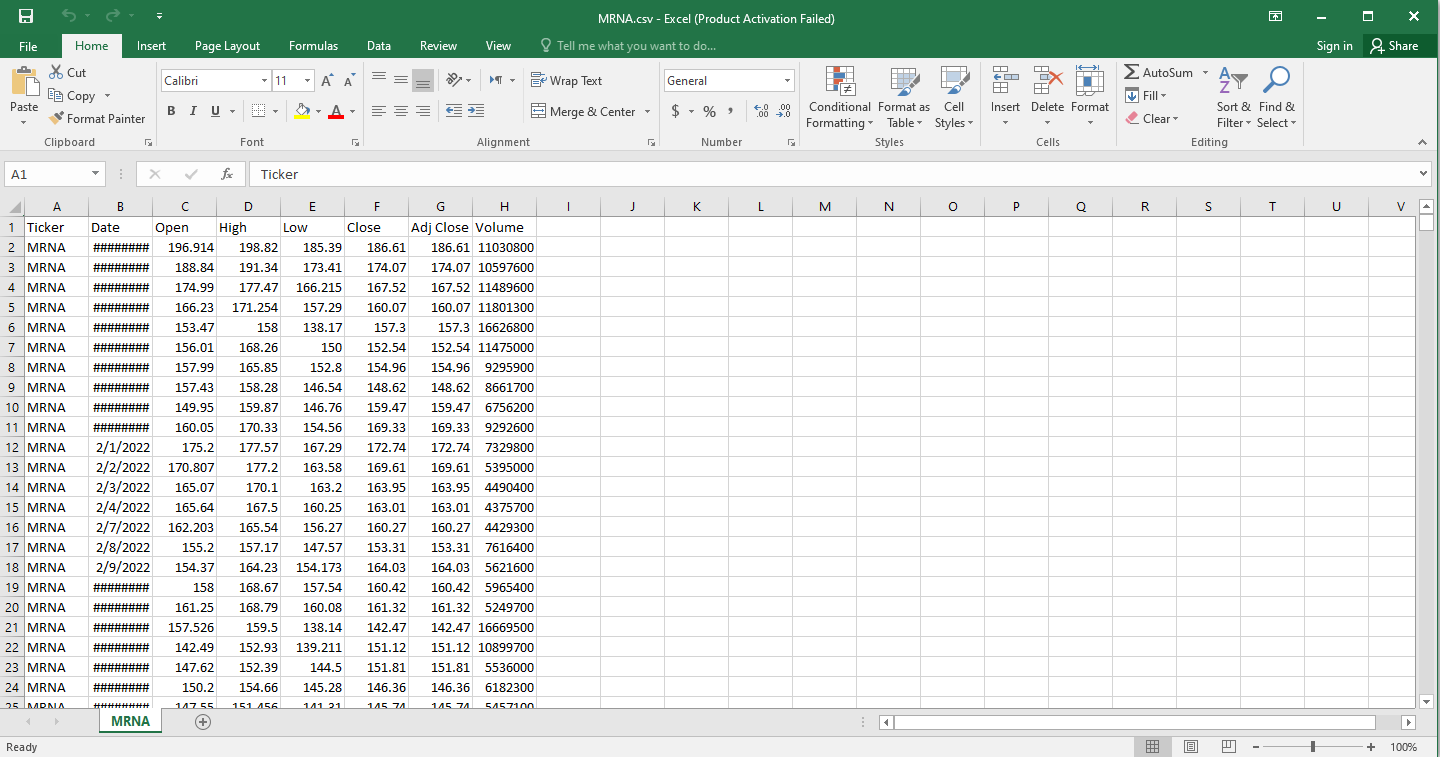
**Q3,4:**



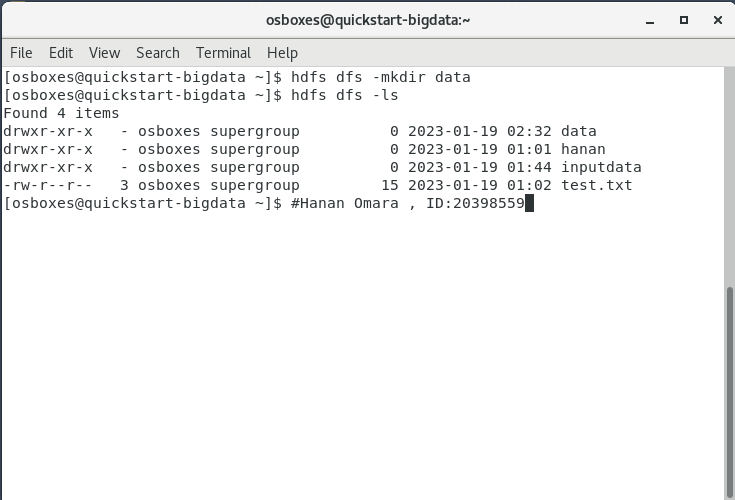
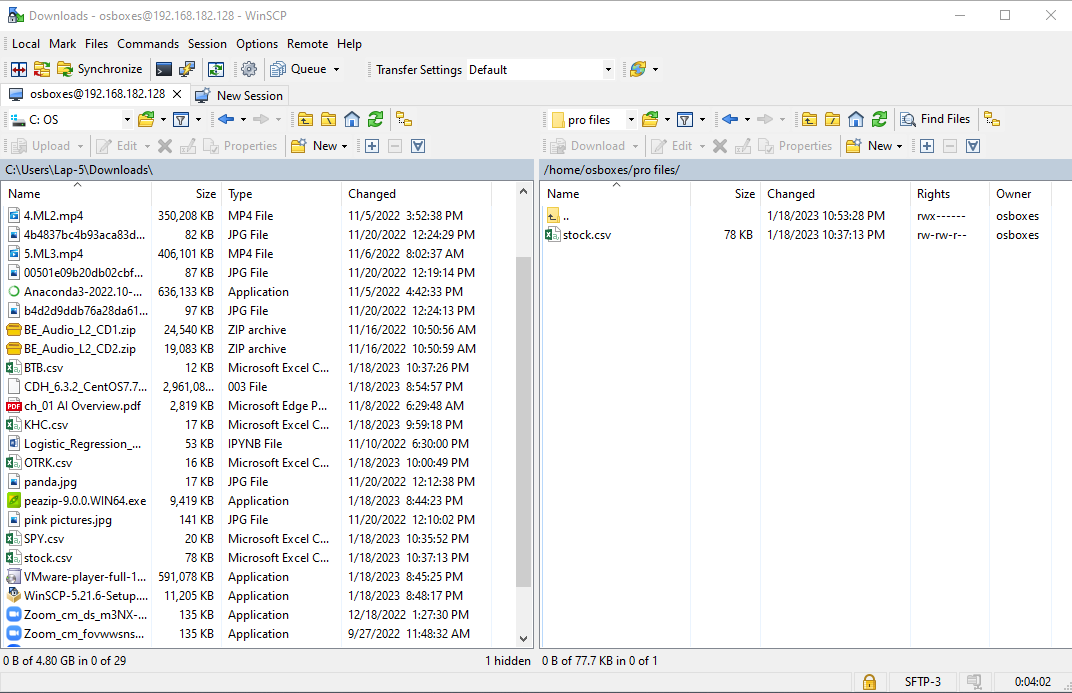
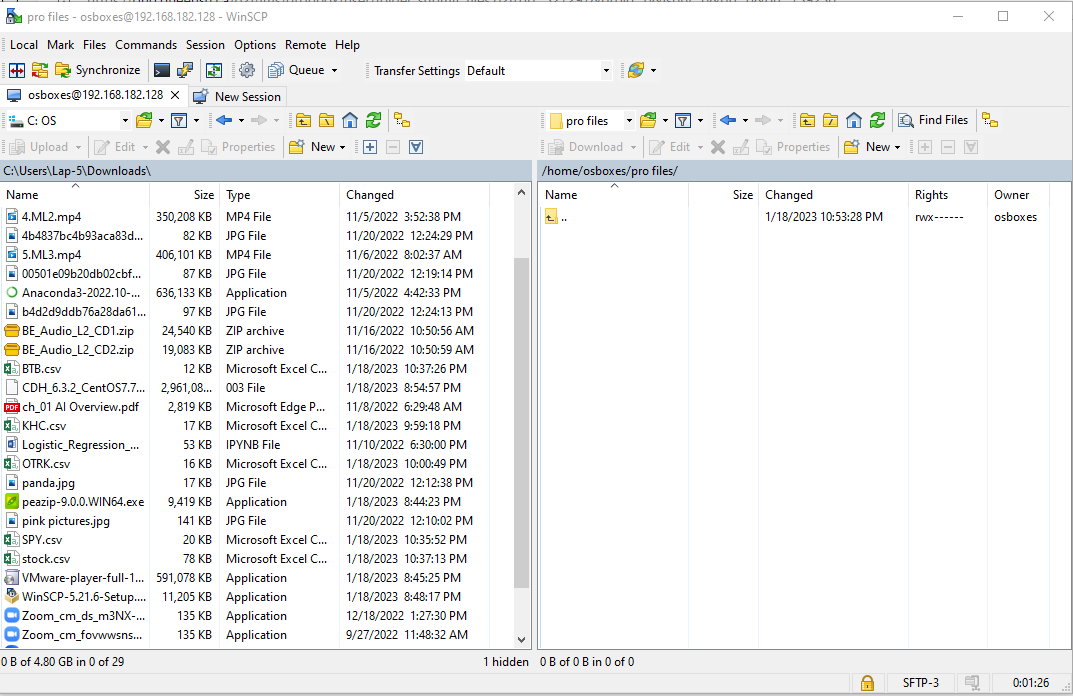
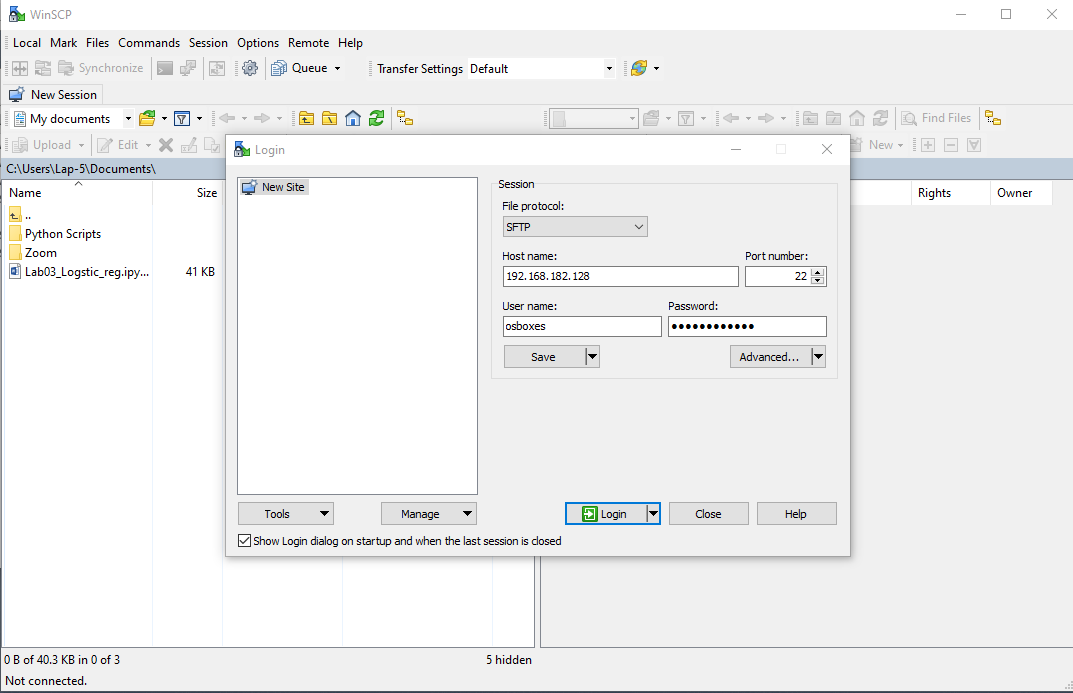
**Q5:**



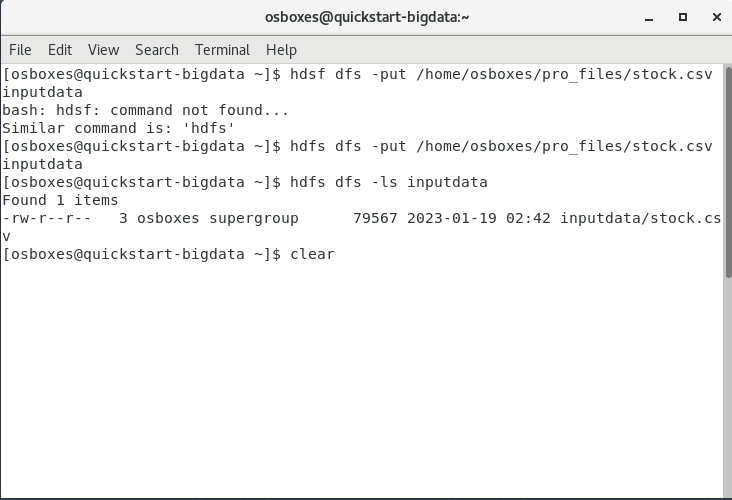
**Q6:**



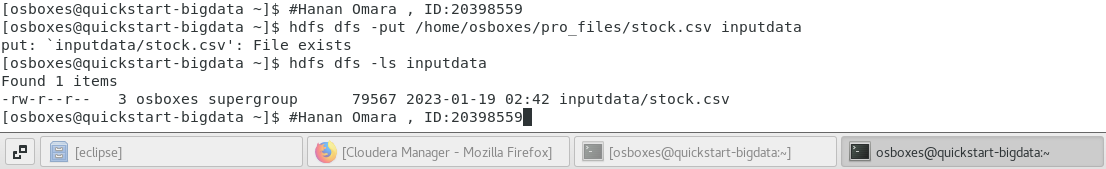
**Q7:**



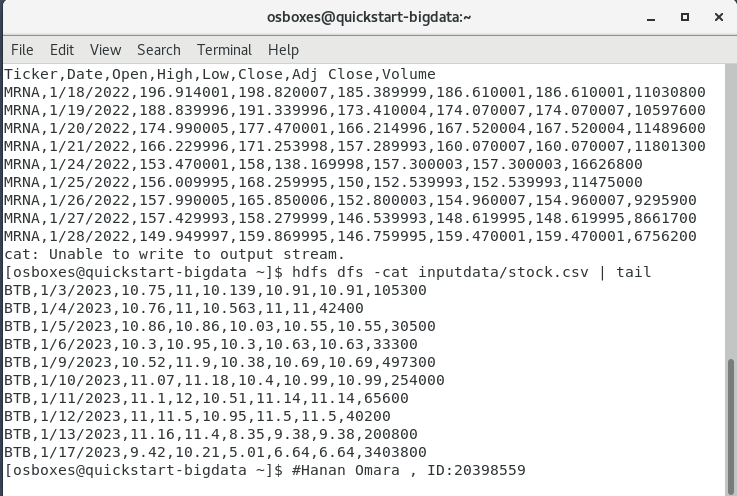
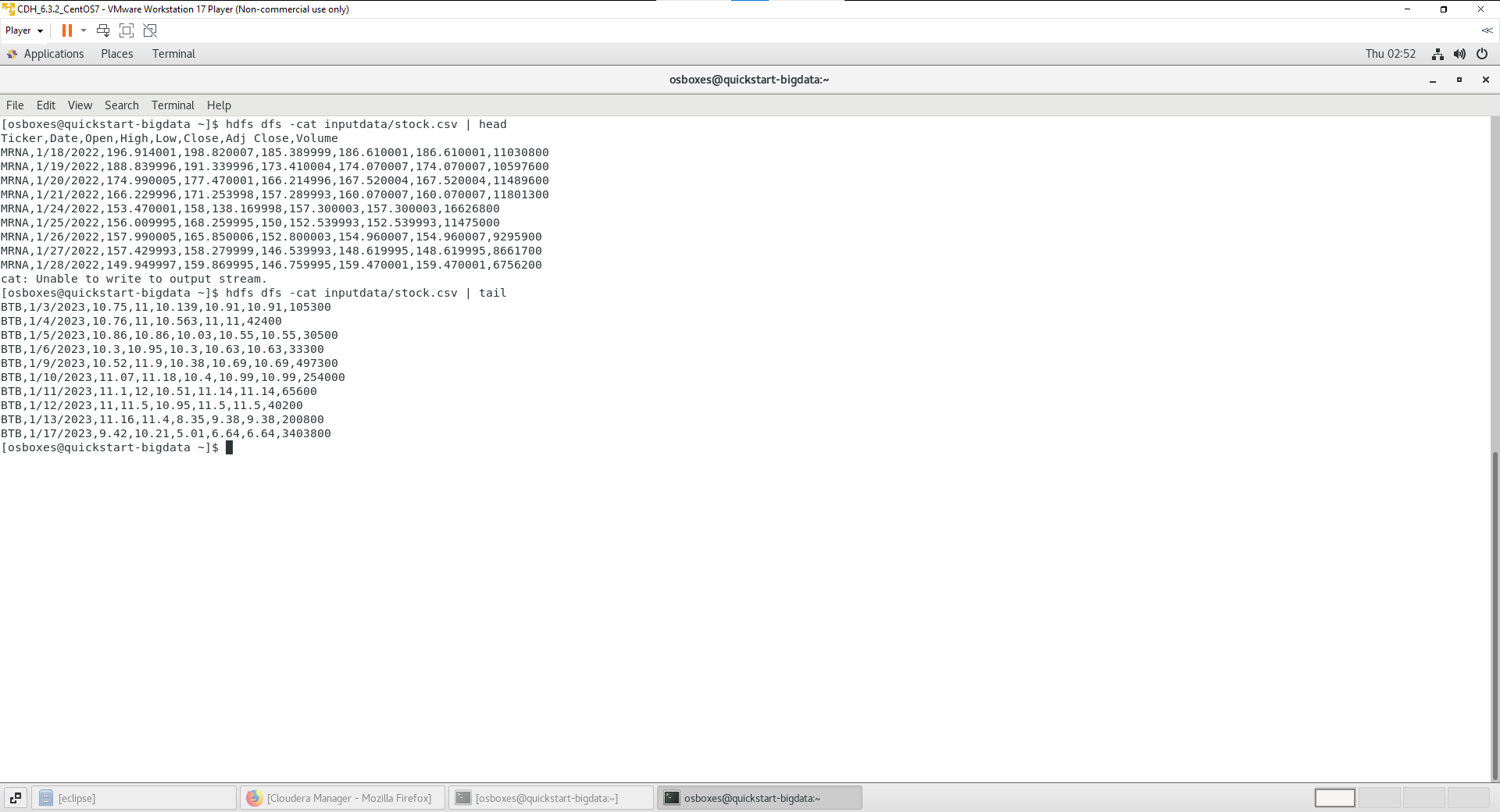
**Q8:**



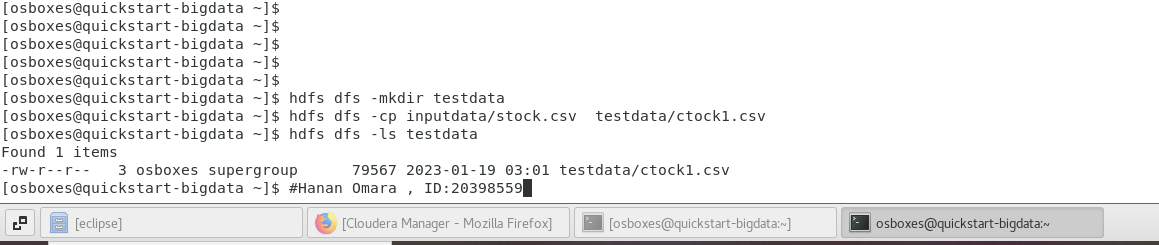
**This is screen with id**



**Q9:**



**Q10:**



**Part B**

-The code of Mapper:

package hanan;

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;

//omara , ID=20398559

public class StockStudyMapper 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[] stockArray = inputLine.split(",");

try {

context.write(new Text(stockArray[0]), new FloatWritable(Float.parseFloat(stockArray[6])));

} catch (NumberFormatException e) {

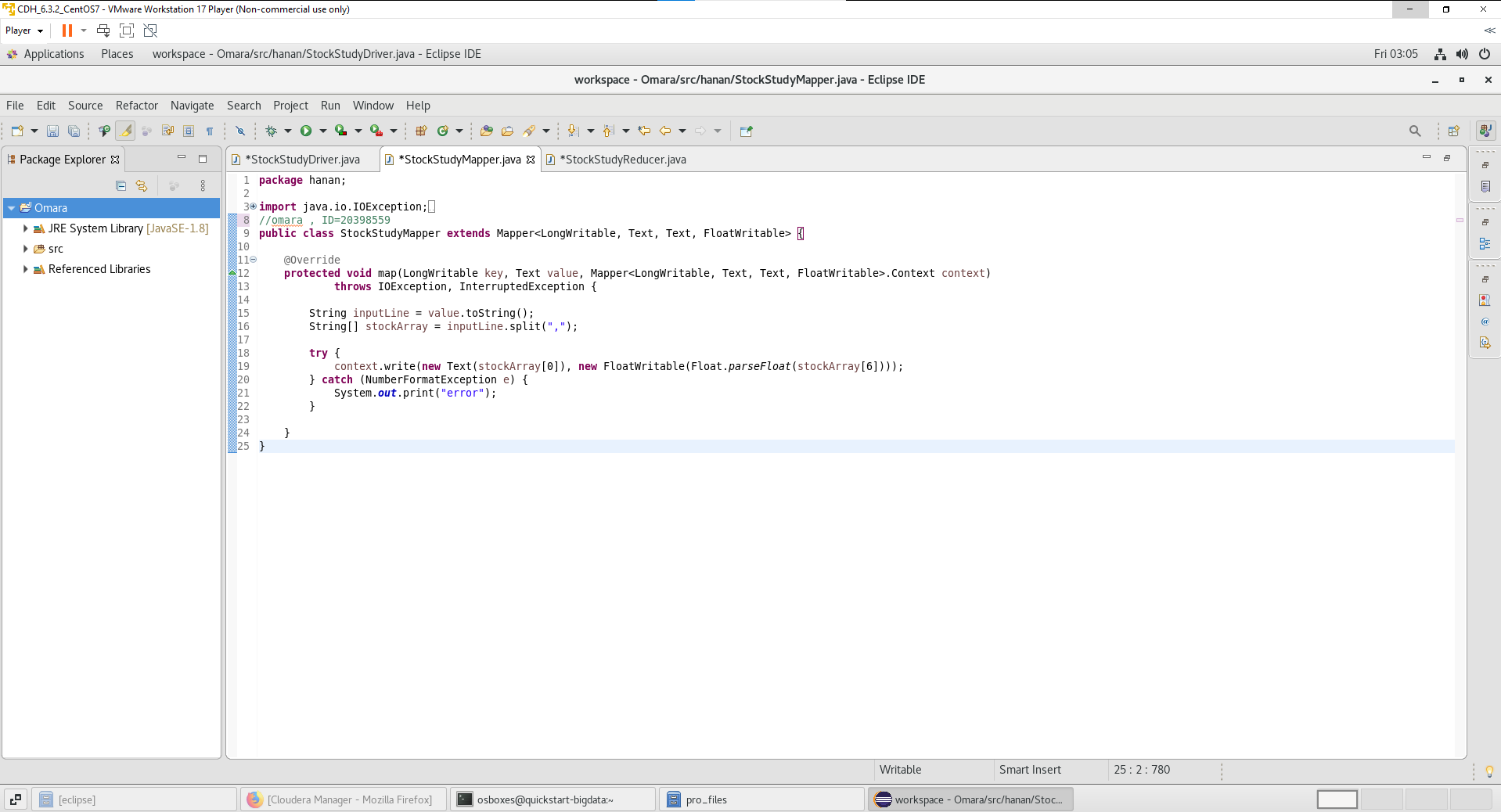
System.out.print("error");

}

}

}

The screenshot for that



-The code of Reducer

package hanan;

import java.io.IOException;

import org.apache.hadoop.io.FloatWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

//omara , ID=20398559

public class StockStudyReducer 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 minValue = (float) 100000;

for (FloatWritable minimum : values) {

minValue = Math.min(minValue, minimum.get());

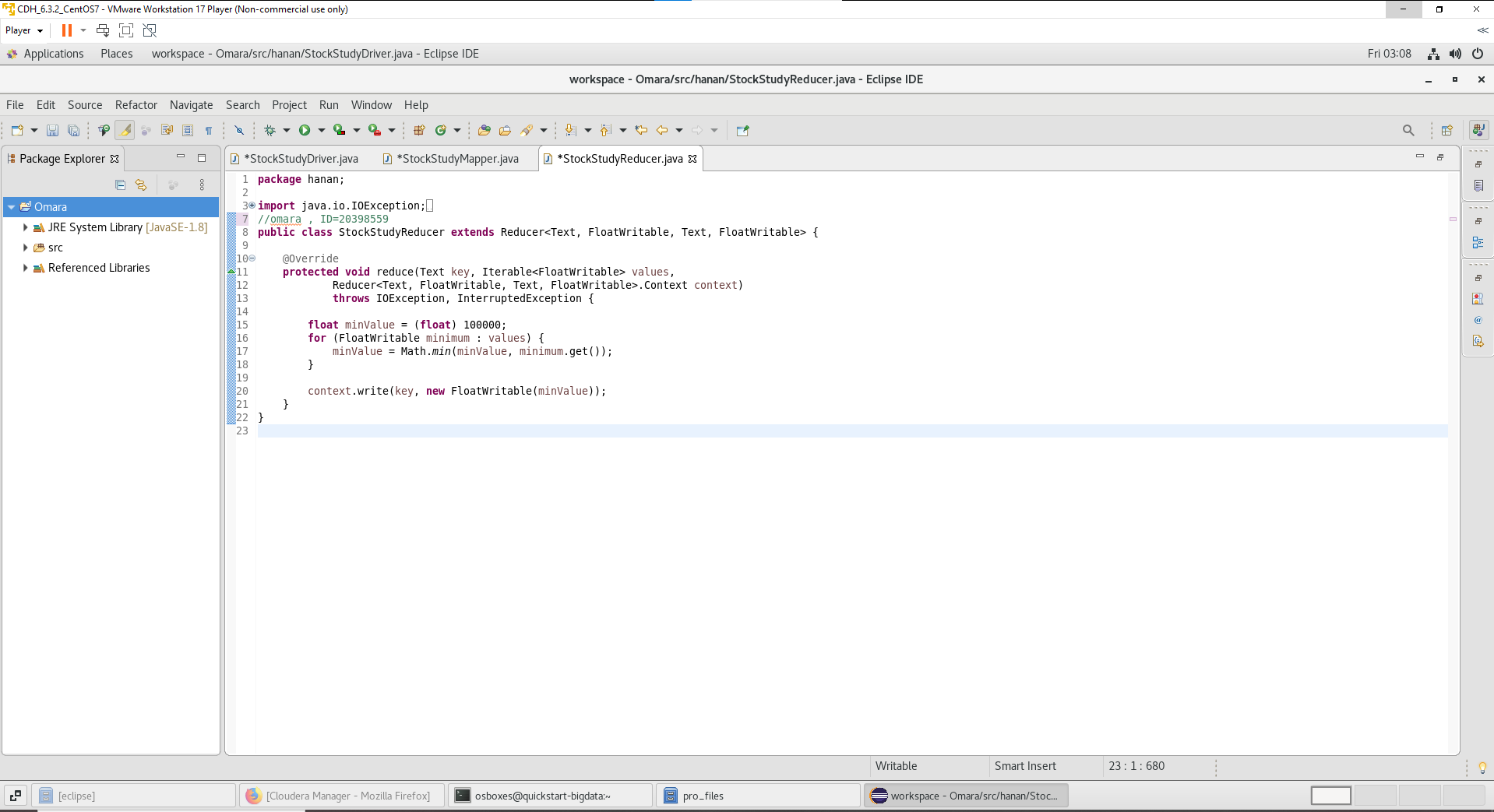
}

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

}

}

The screenshot for that



-The code of Driver

package hanan;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

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 StockStudyDriver {

//omara , ID=20398559

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

Configuration conf = new Configuration();

Job job = new Job(conf);

job.setJarByClass(StockStudyDriver.class);

job.setMapperClass(StockStudyMapper.class);

job.setNumReduceTasks(1);

job.setReducerClass(StockStudyReducer.class);

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]));

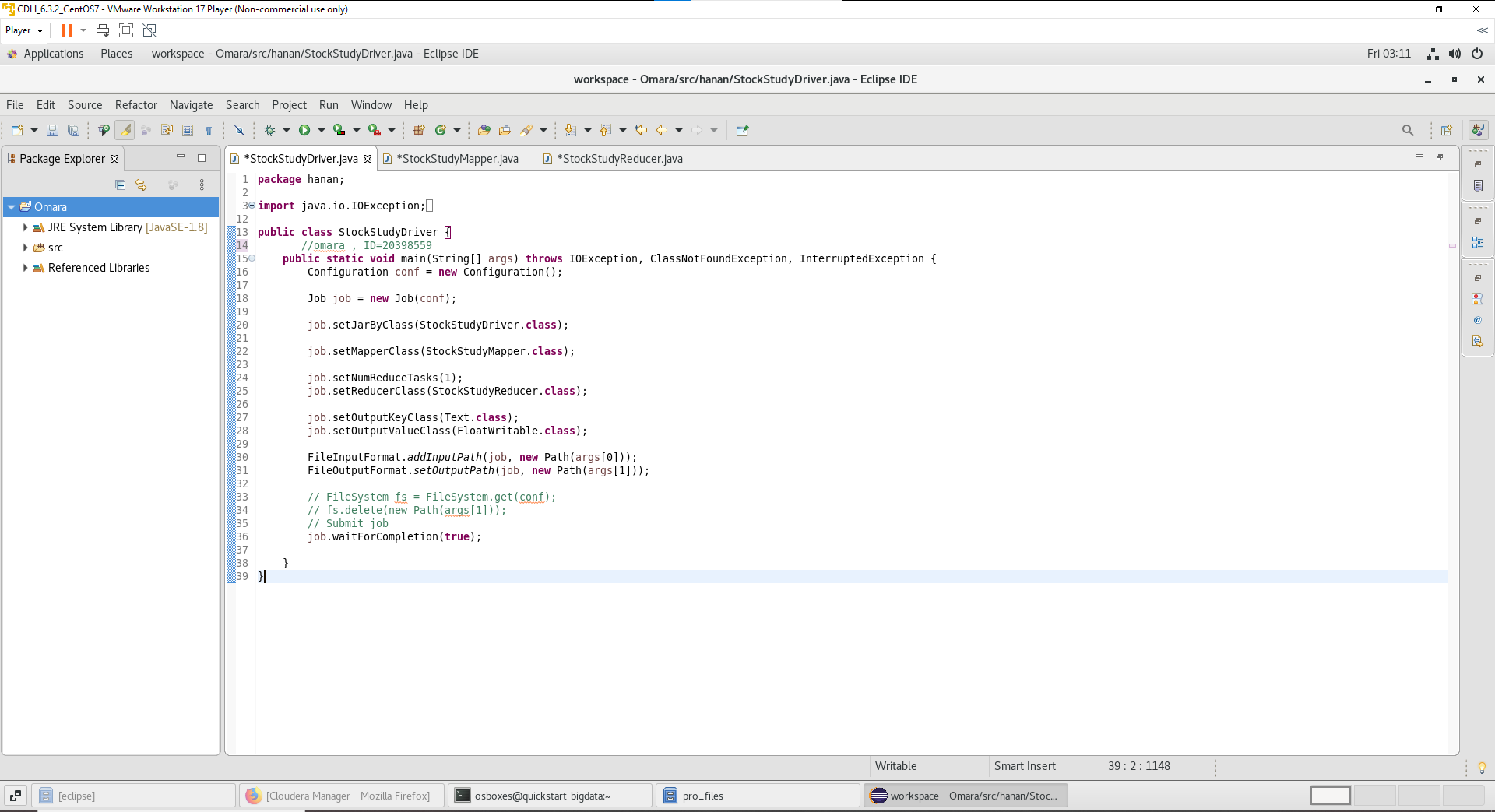
// Submit job

job.waitForCompletion(true);

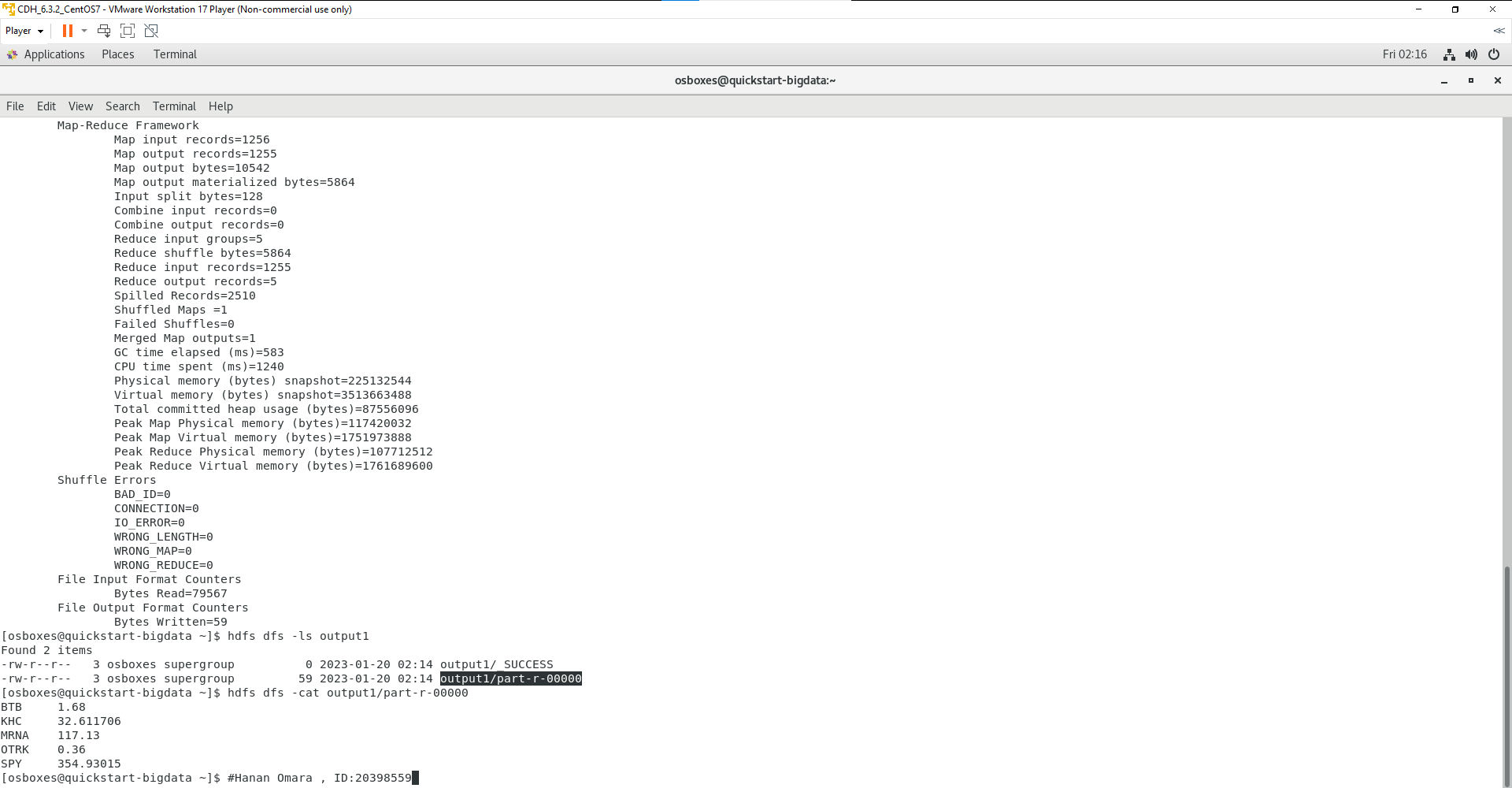
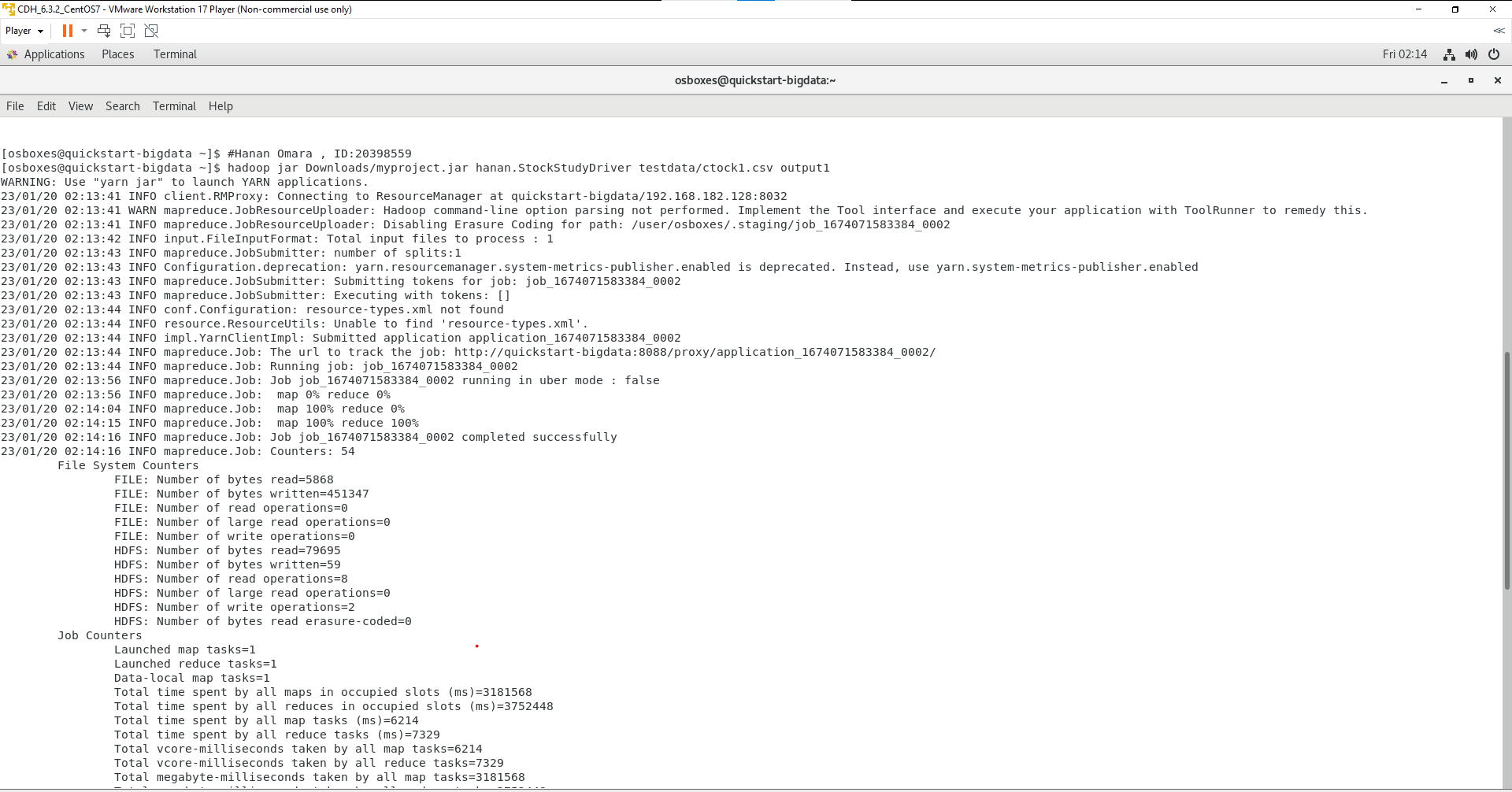
}

}

The screenshot of that



The Output:



**The output file content**

BTB 1.68

KHC 32.611706

MRNA 117.13

OTRK 0.36

SPY 354.93015