TOPIC:

Project on NCDC (National Climate Data Center)

Course: CPSC-651-11-Big Data Systems & Analysis – Fall 2019

Submission to: Prof. Jeongkyu Lee

Submitted by: Srikanth Dabbiru (UB ID 1046112) & Aditya Lavu (UB ID 1066404)

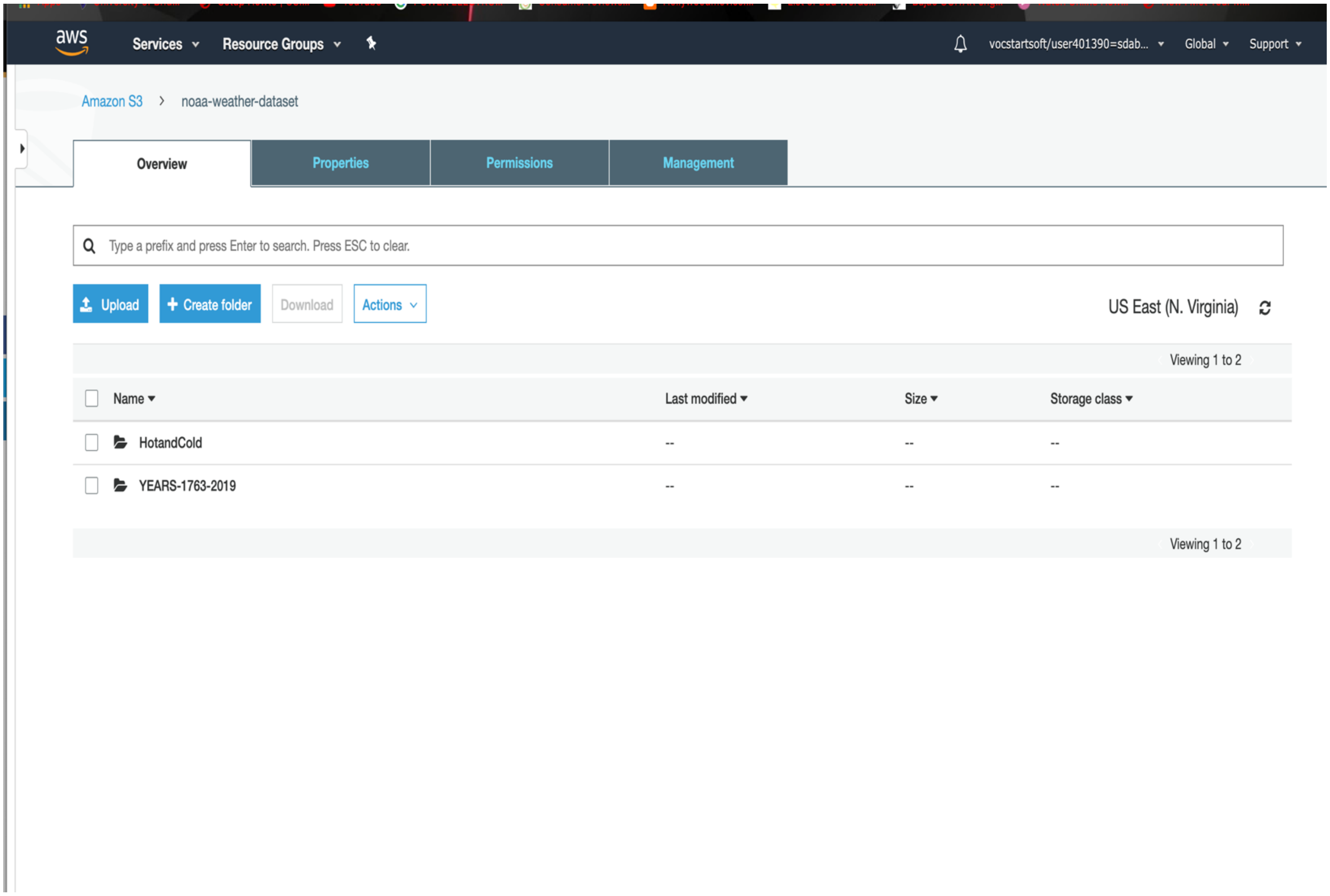
- **<u>Testing the Project on Cloud with AWS:</u>**

1. After testing the sample dataset on Cloudera VM, the project solution is set using AWS.

2. All the dataset files uploaded in .csv format to S3 after creation of a bucket by the name noaa-weather-dataset.

3. Data from years 1763-2019 was uploaded along with the source code jar file.

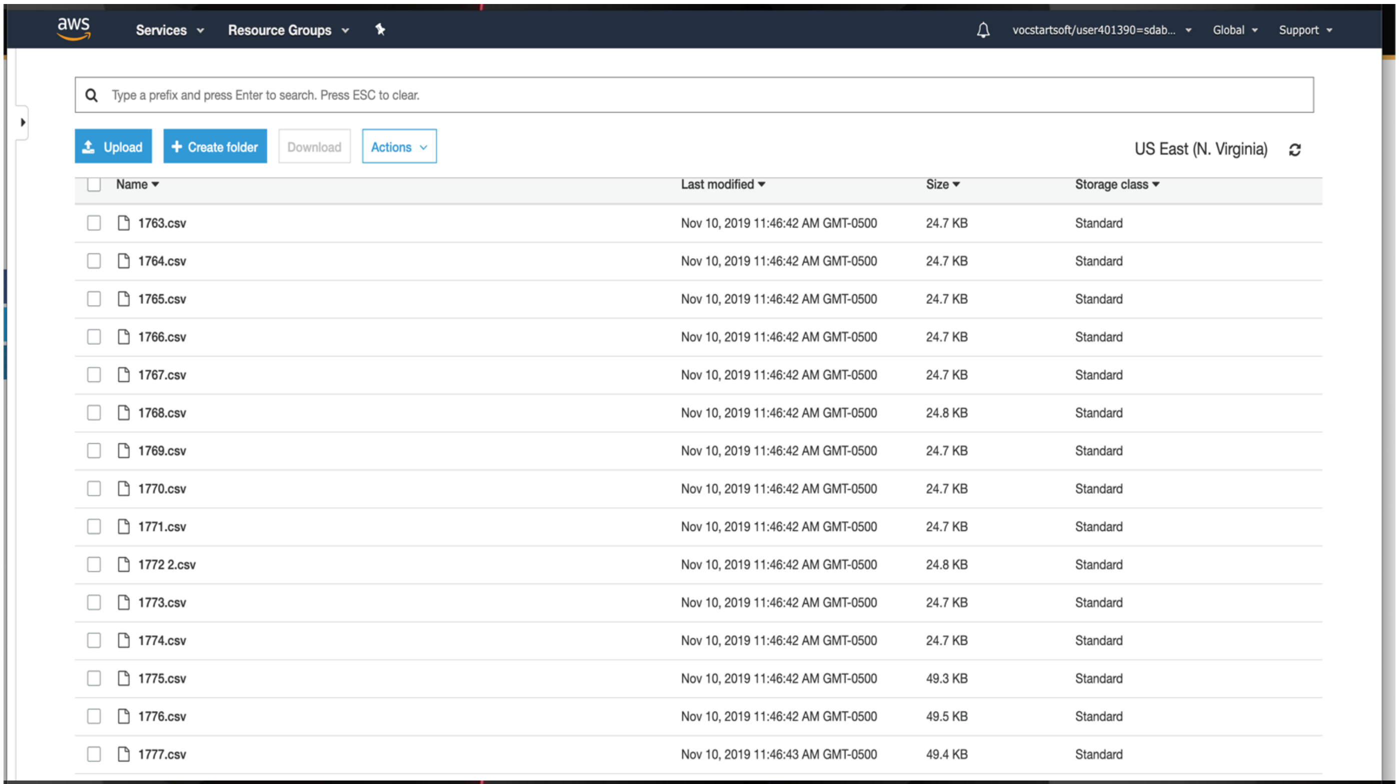
4. Snapshot of the project on AWS :



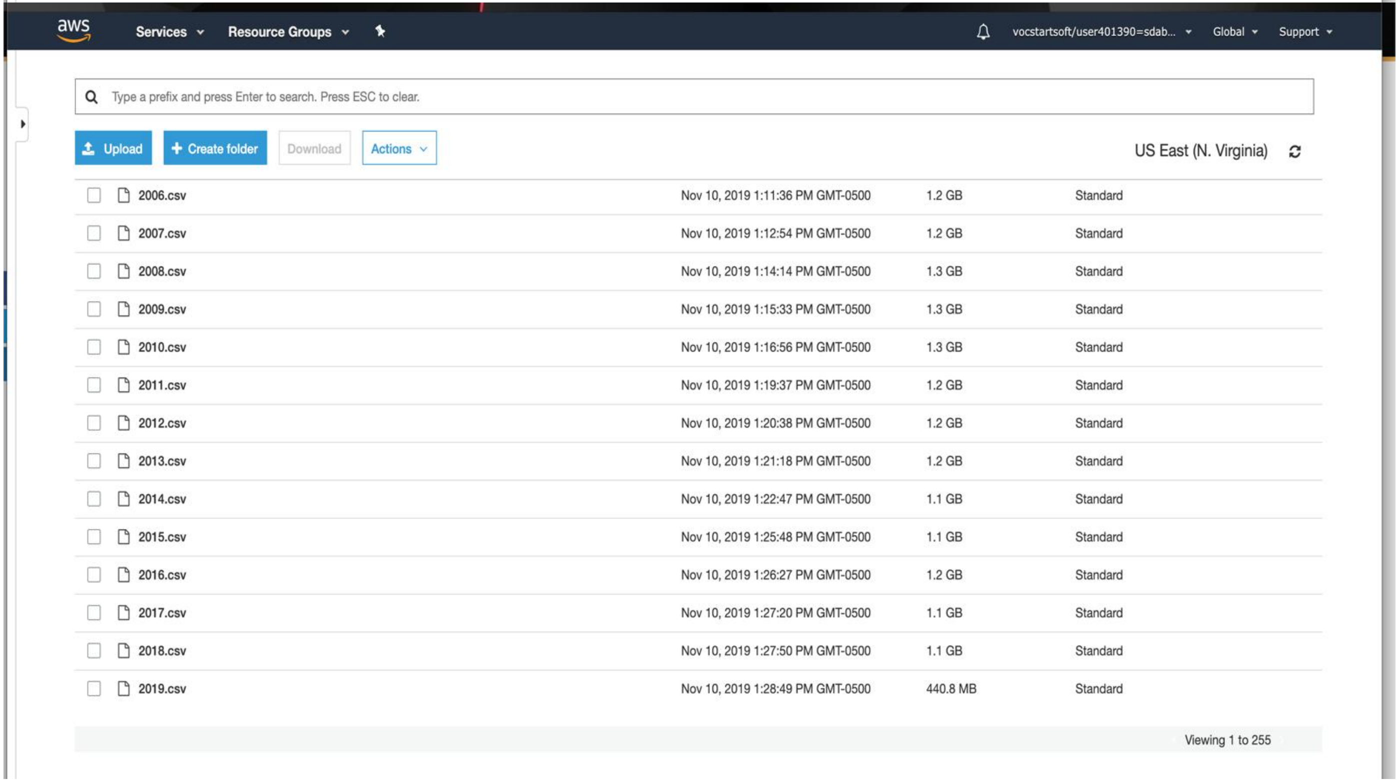
Name	Last modified	Size	Storage class
□ HotandCold	--	--	--
□ YEARS-1763-2019	--	--	--

Viewing 1 to 2

5. Dataset used in almost 17GB in size with more than 20M instances.



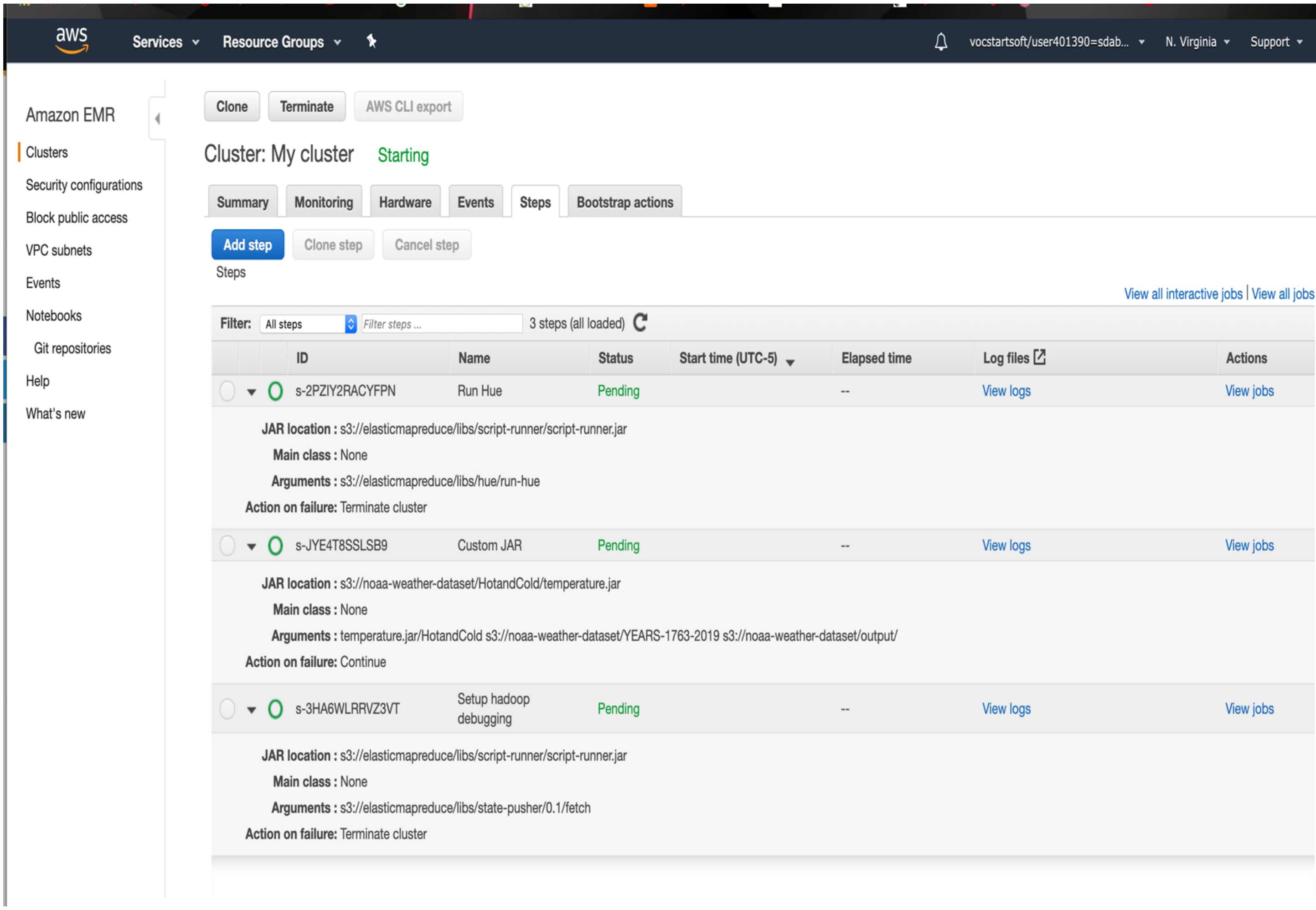
Name	Last modified	Size	Storage class
1763.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1764.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1765.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1766.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1767.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1768.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.8 KB	Standard
1769.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1770.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1771.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1772 2.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.8 KB	Standard
1773.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1774.csv	Nov 10, 2019 11:46:42 AM GMT-0500	24.7 KB	Standard
1775.csv	Nov 10, 2019 11:46:42 AM GMT-0500	49.3 KB	Standard
1776.csv	Nov 10, 2019 11:46:42 AM GMT-0500	49.5 KB	Standard
1777.csv	Nov 10, 2019 11:46:43 AM GMT-0500	49.4 KB	Standard



2006.csv	Nov 10, 2019 1:11:36 PM GMT-0500	1.2 GB	Standard
2007.csv	Nov 10, 2019 1:12:54 PM GMT-0500	1.2 GB	Standard
2008.csv	Nov 10, 2019 1:14:14 PM GMT-0500	1.3 GB	Standard
2009.csv	Nov 10, 2019 1:15:33 PM GMT-0500	1.3 GB	Standard
2010.csv	Nov 10, 2019 1:16:56 PM GMT-0500	1.3 GB	Standard
2011.csv	Nov 10, 2019 1:19:37 PM GMT-0500	1.2 GB	Standard
2012.csv	Nov 10, 2019 1:20:38 PM GMT-0500	1.2 GB	Standard
2013.csv	Nov 10, 2019 1:21:18 PM GMT-0500	1.2 GB	Standard
2014.csv	Nov 10, 2019 1:22:47 PM GMT-0500	1.1 GB	Standard
2015.csv	Nov 10, 2019 1:25:48 PM GMT-0500	1.1 GB	Standard
2016.csv	Nov 10, 2019 1:26:27 PM GMT-0500	1.2 GB	Standard
2017.csv	Nov 10, 2019 1:27:20 PM GMT-0500	1.1 GB	Standard
2018.csv	Nov 10, 2019 1:27:50 PM GMT-0500	1.1 GB	Standard
2019.csv	Nov 10, 2019 1:28:49 PM GMT-0500	440.8 MB	Standard

Viewing 1 to 255

6. Clusters were created on EC2 and initiated after making the necessary changes.



Cluster: My cluster Starting

Steps

Filter: All steps Filter steps ... 3 steps (all loaded)

ID	Name	Status	Start time (UTC-5)	Elapsed time	Log files	Actions
s-2PZIY2RACYFPN	Run Hue	Pending		--	View logs	View jobs
s-JYE4T8SSLSB9	Custom JAR	Pending		--	View logs	View jobs
s-3HA6WLRRVZ3VT	Setup hadoop debugging	Pending		--	View logs	View jobs

s-2PZIY2RACYFPN:

JAR location : s3://elasticmapreduce/libs/script-runner/script-runner.jar
Main class : None
Arguments : s3://elasticmapreduce/libs/hue/run-hue
Action on failure: Terminate cluster

s-JYE4T8SSLSB9:

JAR location : s3://noaa-weather-dataset/HotandCold/temperature.jar
Main class : None
Arguments : temperature.jar/HotandCold s3://noaa-weather-dataset/YEARS-1763-2019 s3://noaa-weather-dataset/output/
Action on failure: Continue

s-3HA6WLRRVZ3VT:

JAR location : s3://elasticmapreduce/libs/script-runner/script-runner.jar
Main class : None
Arguments : s3://elasticmapreduce/libs/state-pusher/0.1/fetch
Action on failure: Terminate cluster

- **Mapper-Reducer Code:**

For the mapper, the max temperature class is static, this method takes the input as text data type. Leaving the first five tokens, the 6th token is taken as the temp_max and the 7th as temp_min.

Now temp_max value is set to be >35.0 and the temp_min is set to be <10.0 and are now passed to the reducer step.

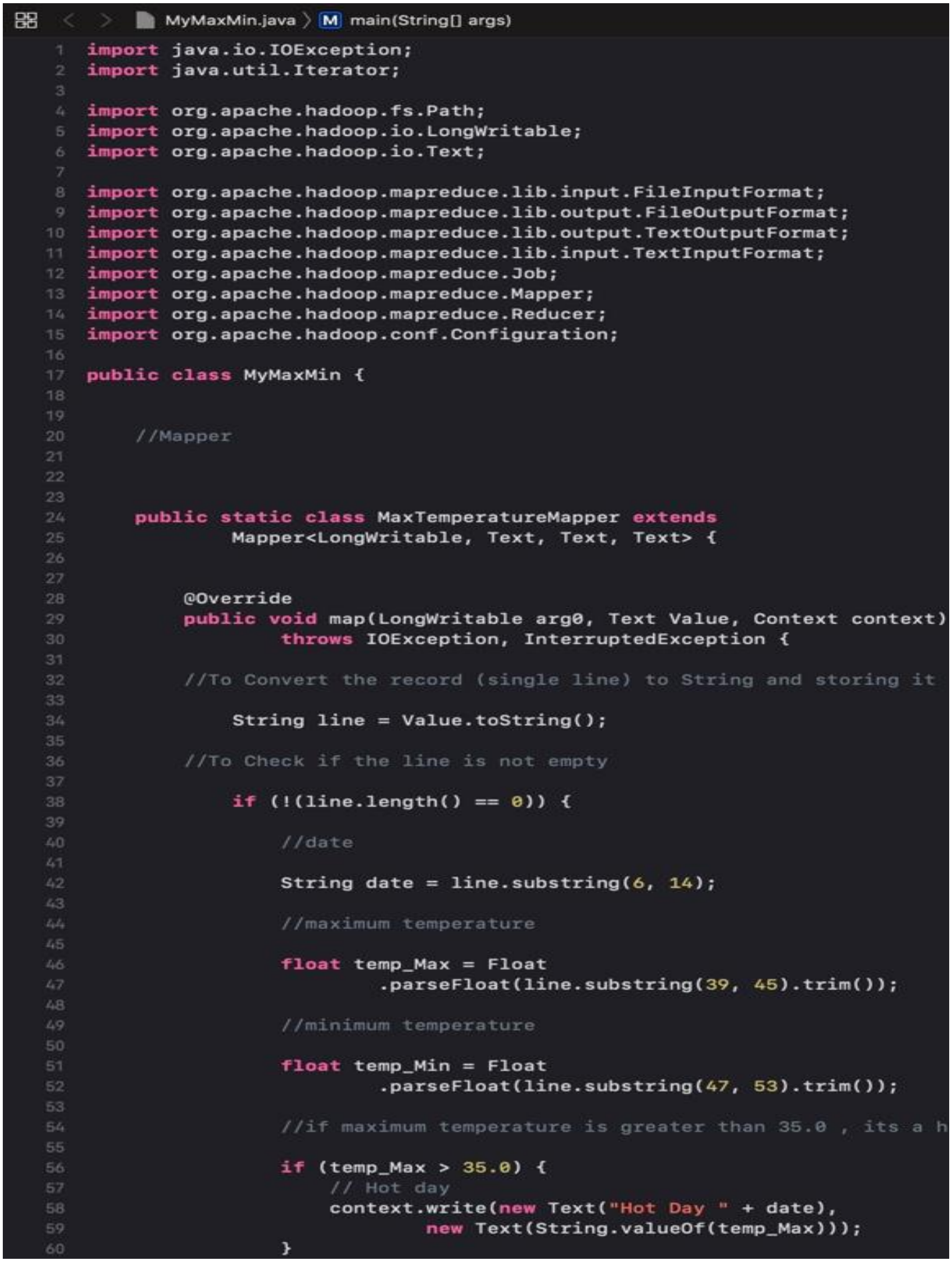
If the temp values for the day are >35.0 output as Hot Day and if <10.0 output as a Cold Day.

For the Reducer method, it takes the input as key and the pairs would be the list of values from the Mapper.

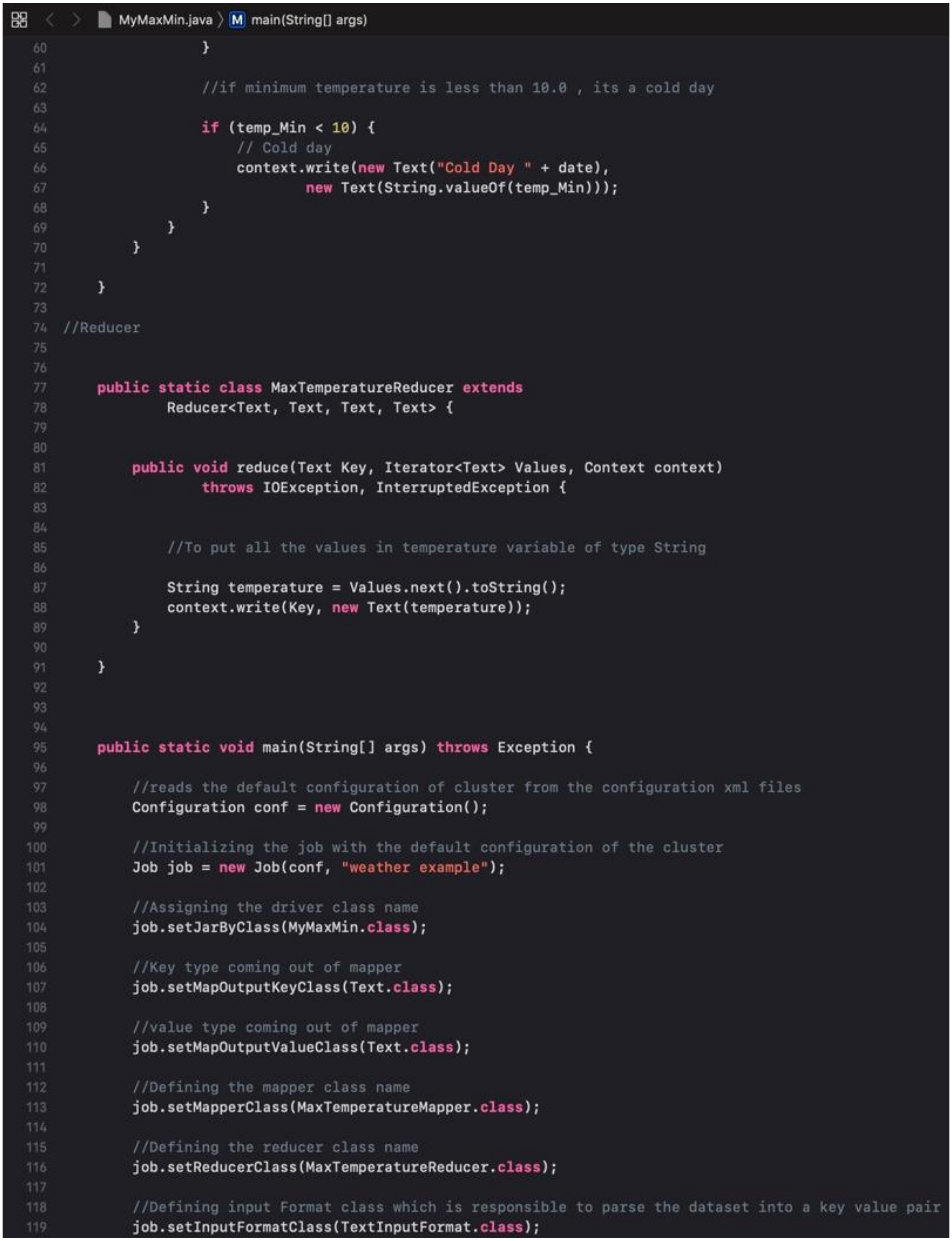
Now **Aggregation** is applied, and it produce the next result.

For the main method, it is used for setting up all the configuration properties. This will be acting as the driver for our Map Reduce code.

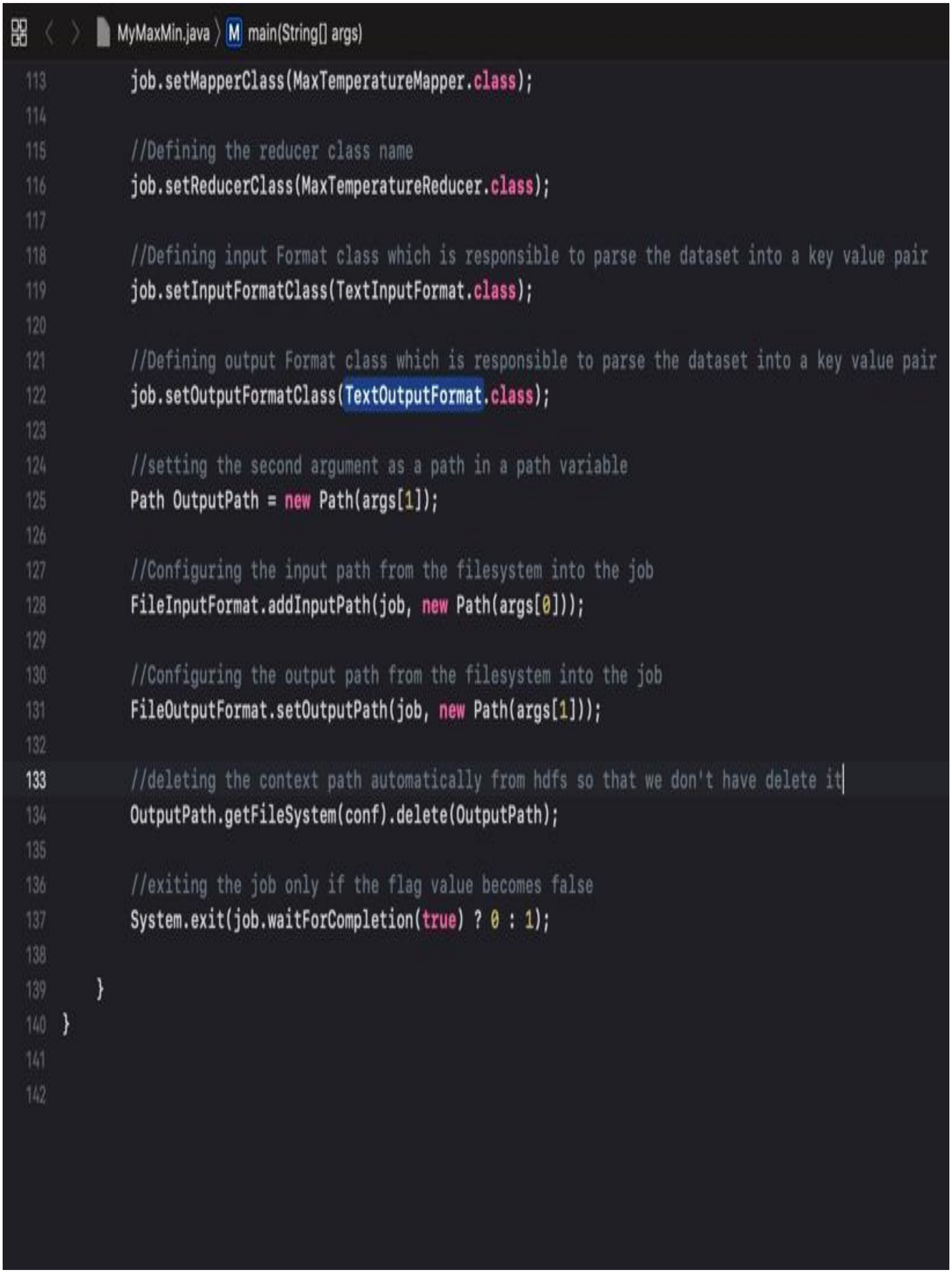
- **Below is the Complete Source Code used:**



```java
import java.io.IOException;
import java.util.Iterator;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;

public class MyMaxMin {


    //Mapper



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


        @Override
        public void map(LongWritable arg0, Text Value, Context context)
                throws IOException, InterruptedException {

        //To Convert the record (single line) to String and storing it

            String line = Value.toString();

        //To Check if the line is not empty

            if (!(line.length() == 0)) {

                //date

                String date = line.substring(6, 14);

                //maximum temperature

                float temp_Max = Float
                        .parseFloat(line.substring(39, 45).trim());

                //minimum temperature

                float temp_Min = Float
                        .parseFloat(line.substring(47, 53).trim());

                //if maximum temperature is greater than 35.0 , its a h

                if (temp_Max > 35.0) {
                    // Hot day
                    context.write(new Text("Hot Day " + date),
                            new Text(String.valueOf(temp_Max)));
                }
```



```java
                }

                //if minimum temperature is less than 10.0 , its a cold day

                if (temp_Min < 10) {
                    // Cold day
                    context.write(new Text("Cold Day " + date),
                            new Text(String.valueOf(temp_Min)));
                }
            }
        }

    }

//Reducer


    public static class MaxTemperatureReducer extends
            Reducer<Text, Text, Text, Text> {


        public void reduce(Text Key, Iterator<Text> Values, Context context)
                throws IOException, InterruptedException {


            //To put all the values in temperature variable of type String

            String temperature = Values.next().toString();
            context.write(Key, new Text(temperature));
        }

    }



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

        //reads the default configuration of cluster from the configuration xml files
        Configuration conf = new Configuration();

        //Initializing the job with the default configuration of the cluster
        Job job = new Job(conf, "weather example");

        //Assigning the driver class name
        job.setJarByClass(MyMaxMin.class);

        //Key type coming out of mapper
        job.setMapOutputKeyClass(Text.class);

        //value type coming out of mapper
        job.setMapOutputValueClass(Text.class);

        //Defining the mapper class name
        job.setMapperClass(MaxTemperatureMapper.class);

        //Defining the reducer class name
        job.setReducerClass(MaxTemperatureReducer.class);

        //Defining input Format class which is responsible to parse the dataset into a key value pair
        job.setInputFormatClass(TextInputFormat.class);
```

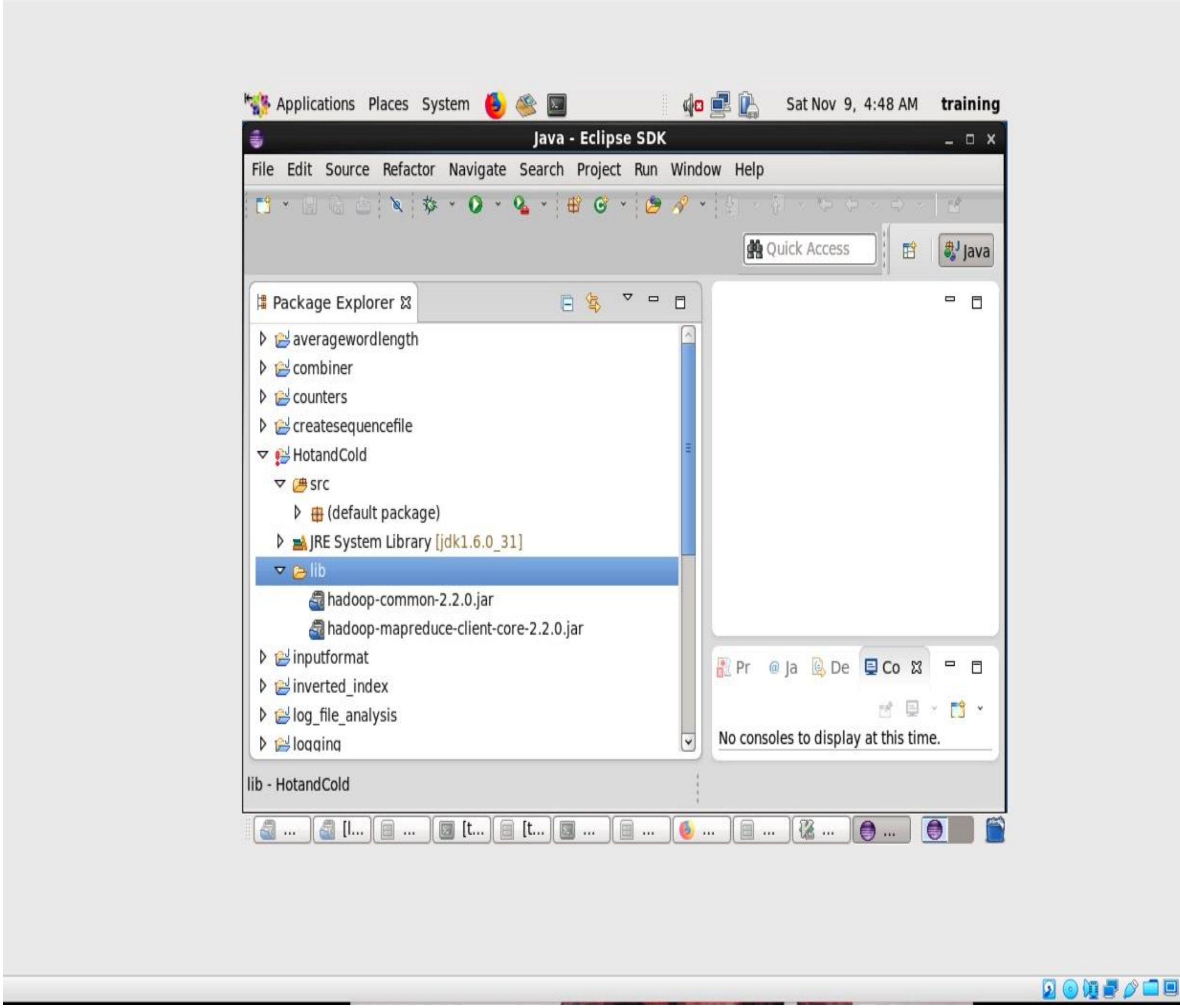


```
        job.setMapperClass(MaxTemperatureMapper.class);

        //Defining the reducer class name
        job.setReducerClass(MaxTemperatureReducer.class);

        //Defining input Format class which is responsible to parse the dataset into a key value pair
        job.setInputFormatClass(TextInputFormat.class);

        //Defining output Format class which is responsible to parse the dataset into a key value pair
        job.setOutputFormatClass(TextOutputFormat.class);

        //setting the second argument as a path in a path variable
        Path OutputPath = new Path(args[1]);

        //Configuring the input path from the filesystem into the job
        FileInputFormat.addInputPath(job, new Path(args[0]));

        //Configuring the output path from the filesystem into the job
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        //deleting the context path automatically from hdfs so that we don't have delete it
        OutputPath.getFileSystem(conf).delete(OutputPath);

        //exiting the job only if the flag value becomes false
        System.exit(job.waitForCompletion(true) ? 0 : 1);

    }
}
```

- **Eclipse IDE:**

Now the project is created in the Eclipse IDE to analyze the sample dataset.

The jar file is then exported after having no issues with the project files.

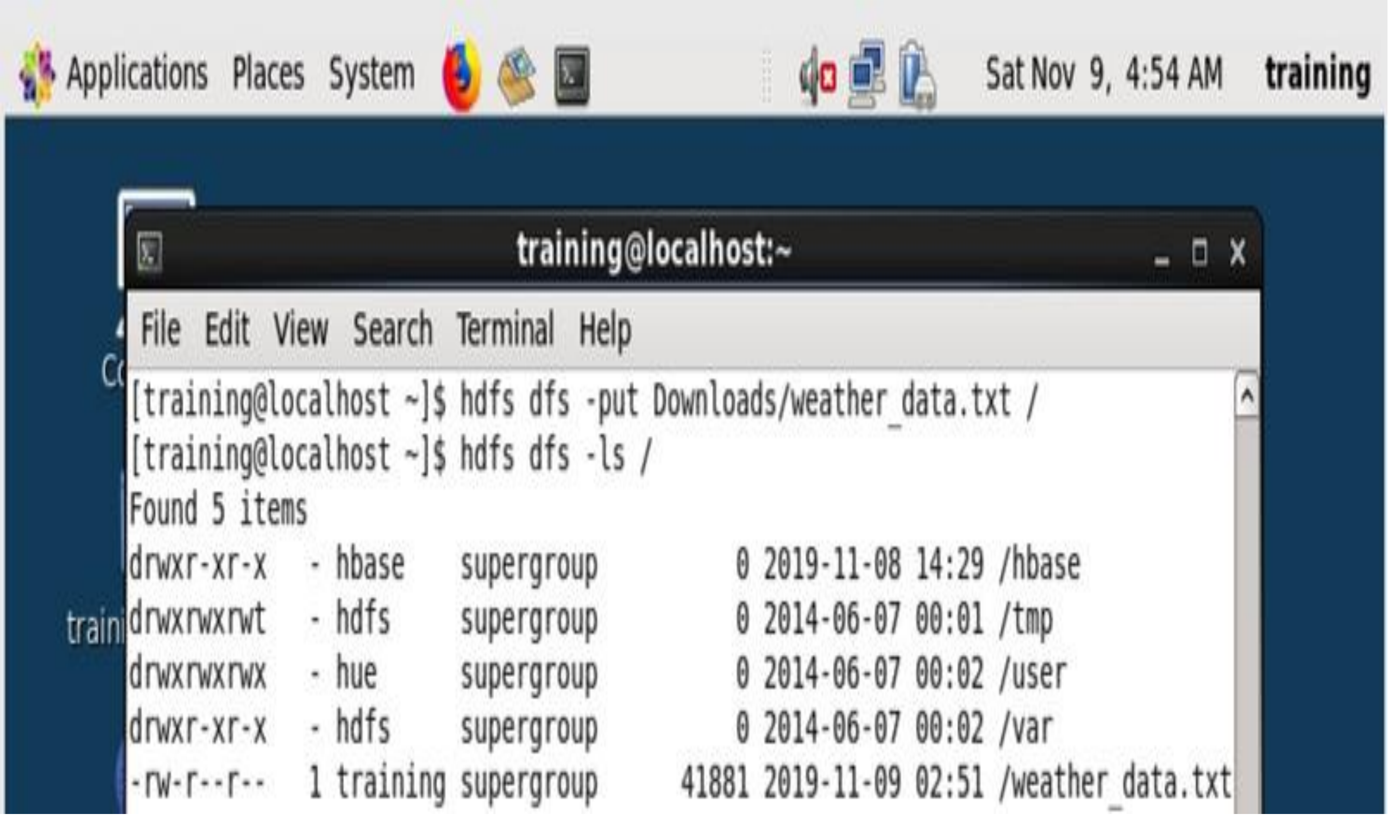


No consoles to display at this time.

lib - HotandCold

The next step was to send the sample dataset onto HDFS.

Hdfs dfs -put Downloads/Noaa_Weather_data.txt /

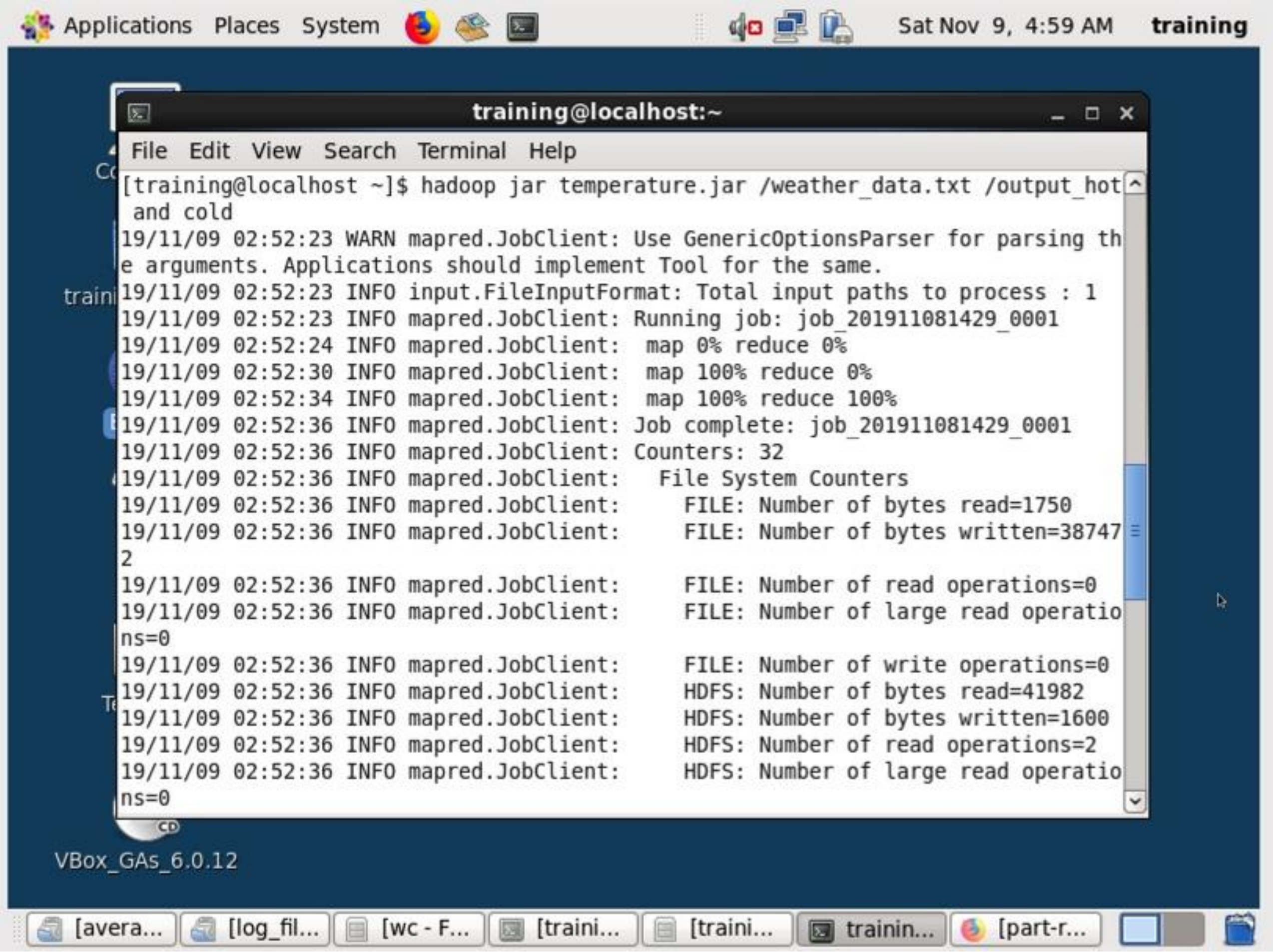


Applications Places System
Sat Nov 9, 4:54 AM training
training@localhost:~
File Edit View Search Terminal Help

```
[training@localhost ~]$ hdfs dfs -put Downloads/weather_data.txt /
[training@localhost ~]$ hdfs dfs -ls /
Found 5 items
drwxr-xr-x   - hbase    supergroup          0 2019-11-08 14:29 /hbase
drwxrwxrwt   - hdfs     supergroup          0 2014-06-07 00:01 /tmp
drwxrwxrwx   - hue      supergroup          0 2014-06-07 00:02 /user
drwxr-xr-x   - hdfs     supergroup          0 2014-06-07 00:02 /var
-rw-r--r--   1 training supergroup      41881 2019-11-09 02:51 /weather_data.txt
```

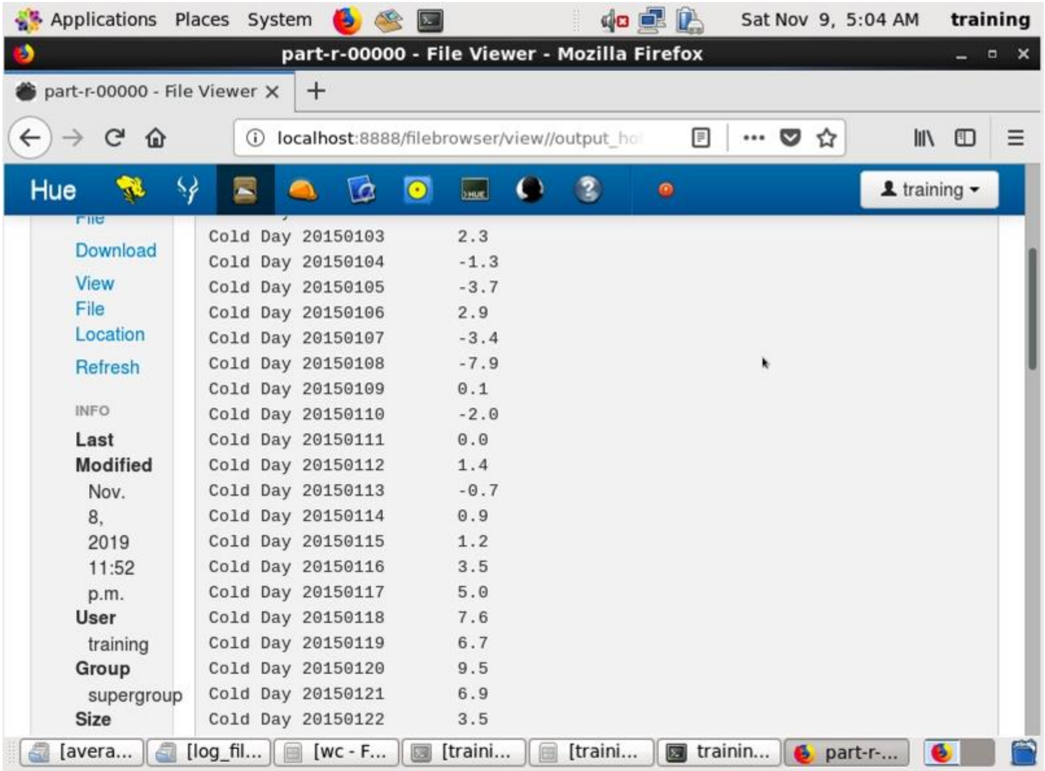
Run the Jar file for output.

Hadoop jar temperature.jar /Noaa_weather_data.txt /output_hotandcold



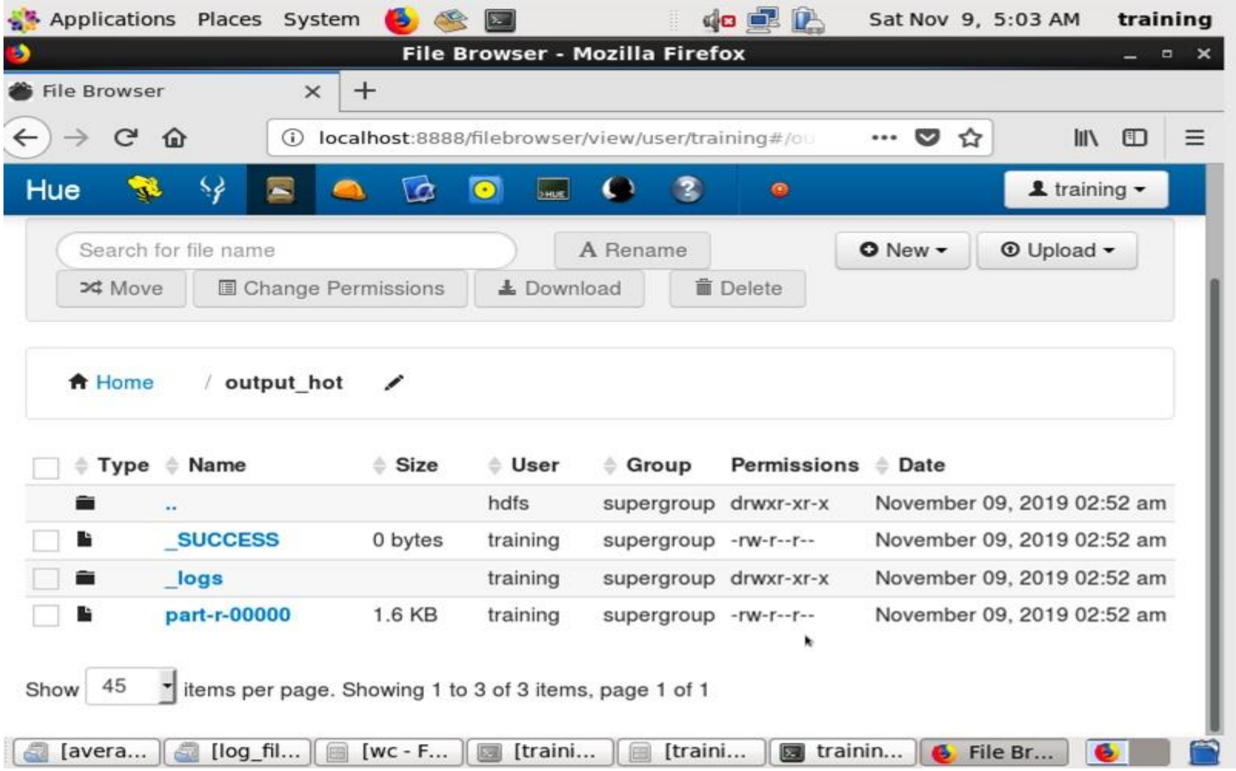
```
[training@localhost ~]$ hadoop jar temperature.jar /weather_data.txt /output_hot
 and cold
19/11/09 02:52:23 WARN mapred.JobClient: Use GenericOptionsParser for parsing th
e arguments. Applications should implement Tool for the same.
19/11/09 02:52:23 INFO input.FileInputFormat: Total input paths to process : 1
19/11/09 02:52:23 INFO mapred.JobClient: Running job: job_201911081429_0001
19/11/09 02:52:24 INFO mapred.JobClient:  map 0% reduce 0%
19/11/09 02:52:30 INFO mapred.JobClient:  map 100% reduce 0%
19/11/09 02:52:34 INFO mapred.JobClient:  map 100% reduce 100%
19/11/09 02:52:36 INFO mapred.JobClient: Job complete: job_201911081429_0001
19/11/09 02:52:36 INFO mapred.JobClient: Counters: 32
19/11/09 02:52:36 INFO mapred.JobClient:   File System Counters
19/11/09 02:52:36 INFO mapred.JobClient:     FILE: Number of bytes read=1750
19/11/09 02:52:36 INFO mapred.JobClient:     FILE: Number of bytes written=38747
2
19/11/09 02:52:36 INFO mapred.JobClient:     FILE: Number of read operations=0
19/11/09 02:52:36 INFO mapred.JobClient:     FILE: Number of large read operatio
ns=0
19/11/09 02:52:36 INFO mapred.JobClient:     FILE: Number of write operations=0
19/11/09 02:52:36 INFO mapred.JobClient:     HDFS: Number of bytes read=41982
19/11/09 02:52:36 INFO mapred.JobClient:     HDFS: Number of bytes written=1600
19/11/09 02:52:36 INFO mapred.JobClient:     HDFS: Number of read operations=2
19/11/09 02:52:36 INFO mapred.JobClient:     HDFS: Number of large read operatio
ns=0
```

Check the Output directory in the HDFS.


Applications Places System
Sat Nov 9, 5:04 AM training
part-r-00000 - File Viewer - Mozilla Firefox
part-r-00000 - File Viewer
localhost:8888/filebrowser/view//output_hot
Hue
training
File
Download
View
File
Location
Refresh
INFO
[avera... [log_fil... [wc - F... [traini... [traini... trainin... part-r-...

Last Modified
Nov. 8, 2019 11:52 p.m.
User
training
Group
supergroup
Size

```
Cold Day 20150103       2.3
Cold Day 20150104       -1.3
Cold Day 20150105       -3.7
Cold Day 20150106       2.9
Cold Day 20150107       -3.4
Cold Day 20150108       -7.9
Cold Day 20150109       0.1
Cold Day 20150110       -2.0
Cold Day 20150111       0.0
Cold Day 20150112       1.4
Cold Day 20150113       -0.7
Cold Day 20150114       0.9
Cold Day 20150115       1.2
Cold Day 20150116       3.5
Cold Day 20150117       5.0
Cold Day 20150118       7.6
Cold Day 20150119       6.7
Cold Day 20150120       9.5
Cold Day 20150121       6.9
Cold Day 20150122       3.5
```

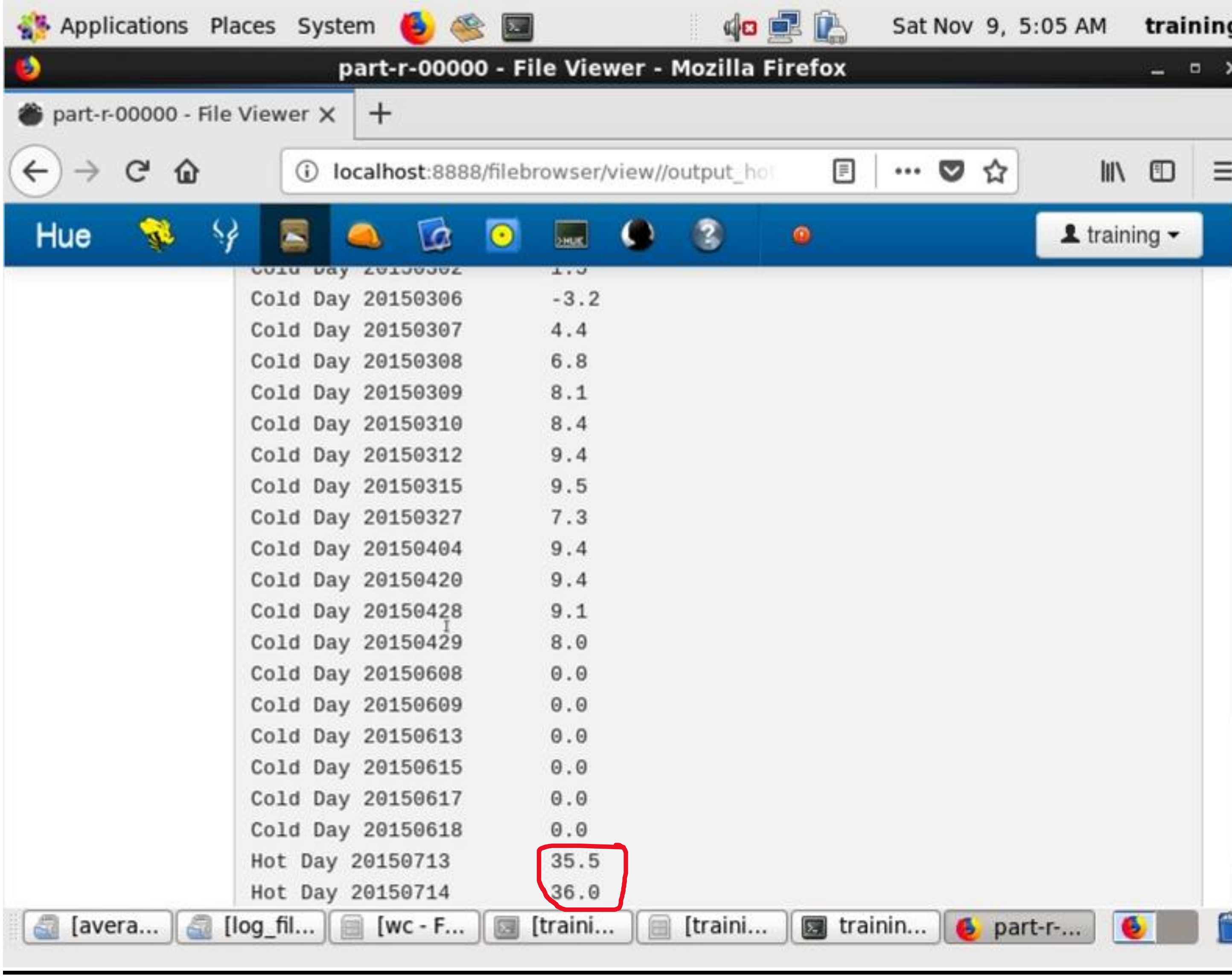

Applications Places System
Sat Nov 9, 5:03 AM training
File Browser - Mozilla Firefox
File Browser
localhost:8888/filebrowser/view/user/training#/ou
Hue
training
Search for file name
A Rename
New
Upload
Move
Change Permissions
Download
Delete
Home / output_hot
[avera... [log_fil... [wc - F... [traini... [traini... trainin... File Br...

☐	Type	Name	Size	User	Group	Permissions	Date
	📁	..		hdfs	supergroup	drwxr-xr-x	November 09, 2019 02:52 am
☐	📄	_SUCCESS	0 bytes	training	supergroup	-rw-r--r--	November 09, 2019 02:52 am
☐	📁	_logs		training	supergroup	drwxr-xr-x	November 09, 2019 02:52 am
☐	📄	part-r-00000	1.6 KB	training	supergroup	-rw-r--r--	November 09, 2019 02:52 am

Show 45 items per page. Showing 1 to 3 of 3 items, page 1 of 1

Results analysis:

Depending on the 2015 sample dataset only two days above 35.0 recorded.



```
Cold Day 20150306       -3.2
Cold Day 20150307       4.4
Cold Day 20150308       6.8
Cold Day 20150309       8.1
Cold Day 20150310       8.4
Cold Day 20150312       9.4
Cold Day 20150315       9.5
Cold Day 20150327       7.3
Cold Day 20150404       9.4
Cold Day 20150420       9.4
Cold Day 20150428       9.1
Cold Day 20150429       8.0
Cold Day 20150608       0.0
Cold Day 20150609       0.0
Cold Day 20150613       0.0
Cold Day 20150615       0.0
Cold Day 20150617       0.0
Cold Day 20150618       0.0
Hot Day 20150713        35.5
Hot Day 20150714        36.0
```