PRACTICAL 4

To determine maximum temperature using Hadoop MapReduce

Map Reduce : works by breaking the processing into two phases: the map phase and the reduce phase. Each phase has key-value pairs as input and output, the types of which may be chosen by the programmer. The programmer also specifies two functions: the map function and the reduce function.

Hadoop MaxTemprature operation occurs in 3 stages –

Mapper Phase

Reducer Phase

Driver code

Dataset – temperature.txt

0067011990999991950051507004+68750+023550FM-12+038299999V0203301N00671220001CN9999999N9+00001+99999999999

0043011990999991950051512004+68750+023550FM-12+038299999V0203201N00671220001CN9999999N9+00221+99999999999

0043011990999991950051518004+68750+023550FM-12+038299999V0203201N00261220001CN9999999N9-00111+99999999999

0043012650999991949032412004+62300+010750FM-12+048599999V0202701N00461220001CN0500001N9+01111+99999999999

0043012650999991949032418004+62300+010750FM-12+048599999V0202701N00461220001CN0500001N9+00781+99999999999

These lines are presented to the map function as the key-value pairs

(0, 0067011990999991950051507004...9999999N9+00001+99999999999...) (106, 0043011990999991950051512004...9999999N9+00221+99999999999...) (212, 0043011990999991950051518004...9999999N9-00111+99999999999...) (318, 0043012650999991949032412004...0500001N9+01111+99999999999...) (424, 0043012650999991949032418004...0500001N9+00781+99999999999...)

The keys are the line offsets within the file, which we ignore in our map function. The map function merely extracts the year and the air temperature (indicated in bold text), and emits them as its output (the temperature values have been interpreted as integers):

(1950, 0)

(1950, 22)

The output from the map function is processed by the MapReduce framework before being sent to the reduce function. This processing sorts and groups the key-value pairs by key. So, continuing the example, our reduce function sees the following input:

(1949, [111, 78])

(1950, [0, 22, −11])

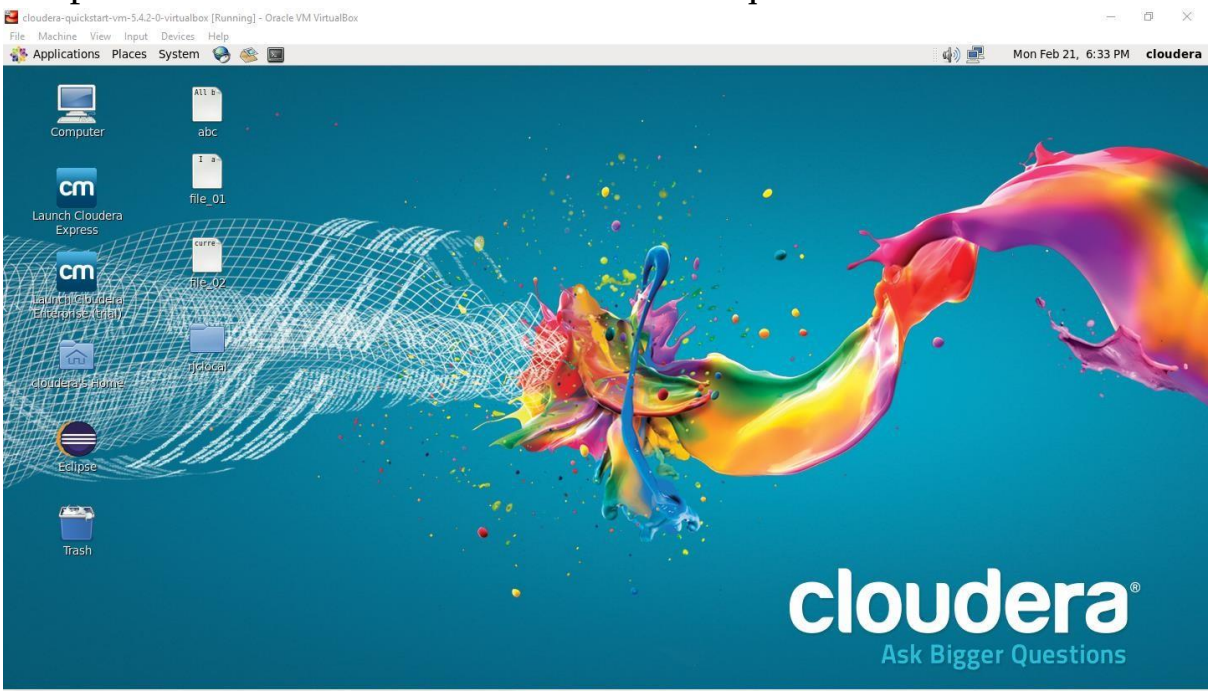
Each year appears with a list of all its air temperature readings. All the reduce function has to do now is iterate through the list and pick up the maximum reading:

(1949, 111)

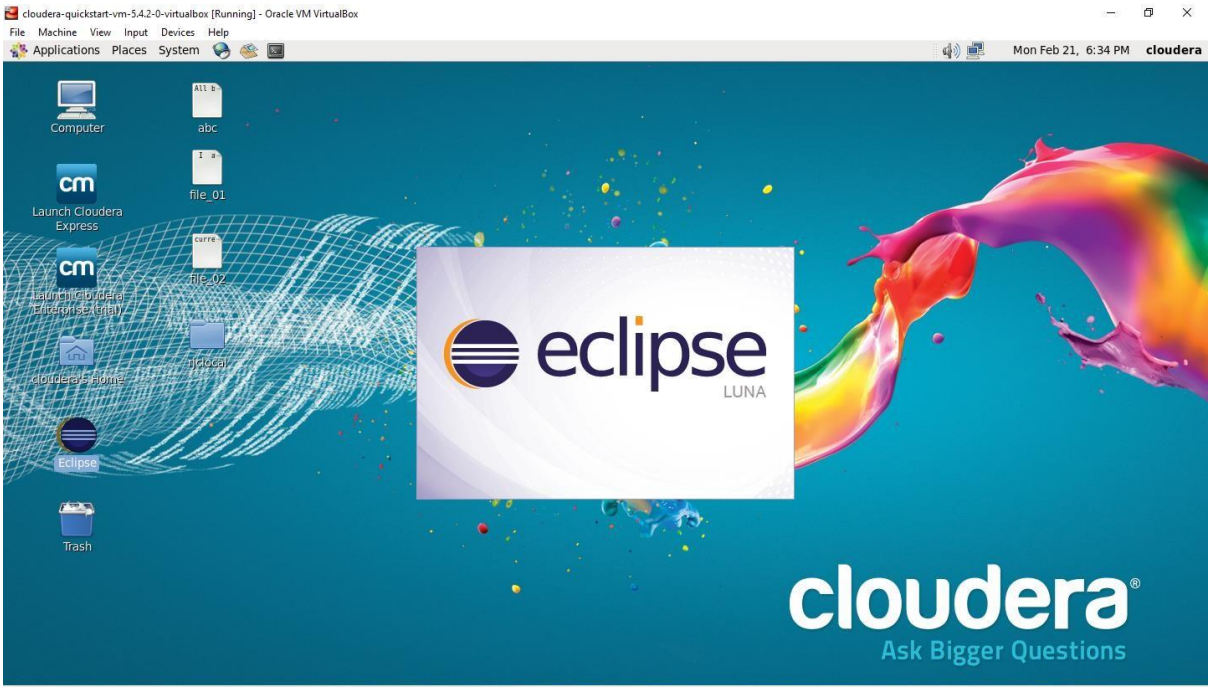
(1950, 22)

- Steps to determine maximum temperature using Hadoop MapReduce in Cloudera:

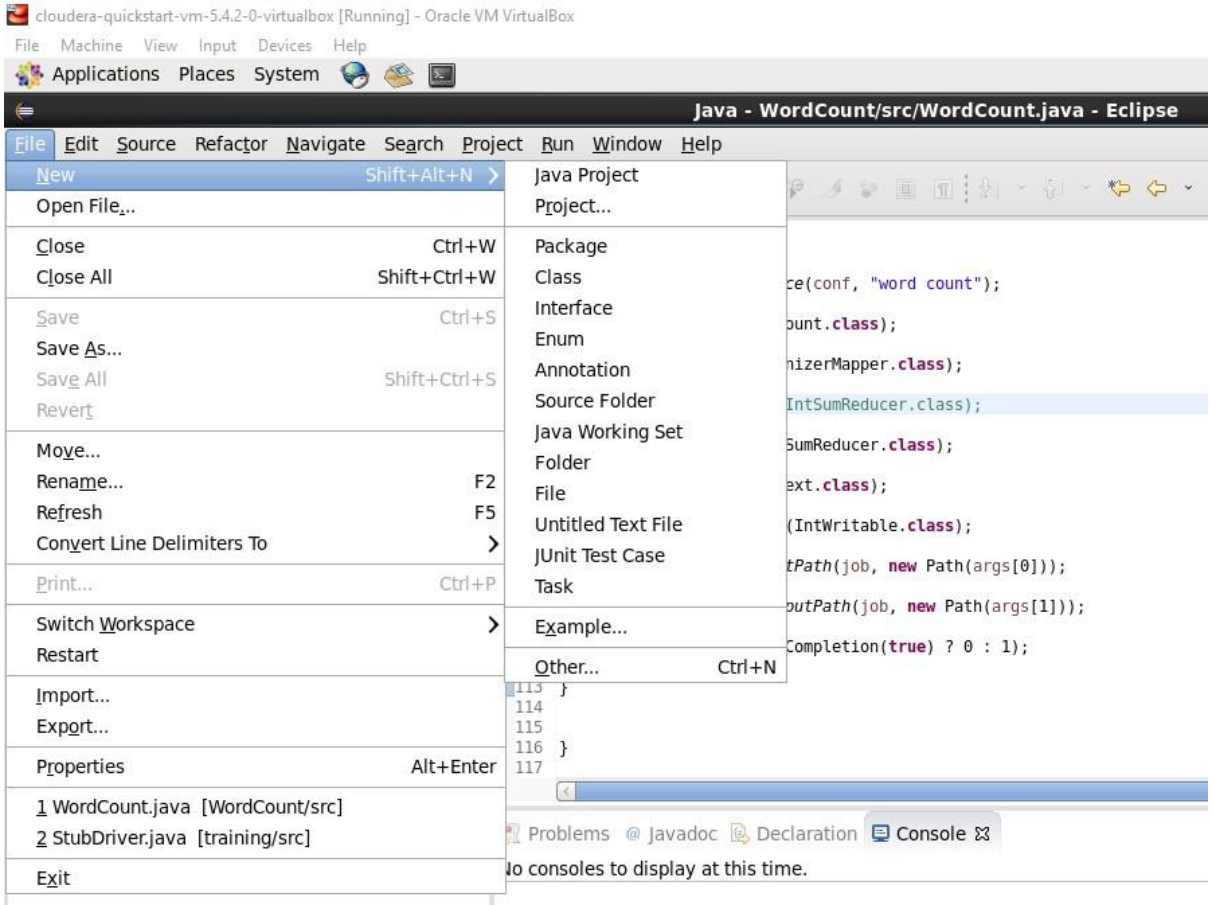
1) Open virtual box and then start cloudera quickstart

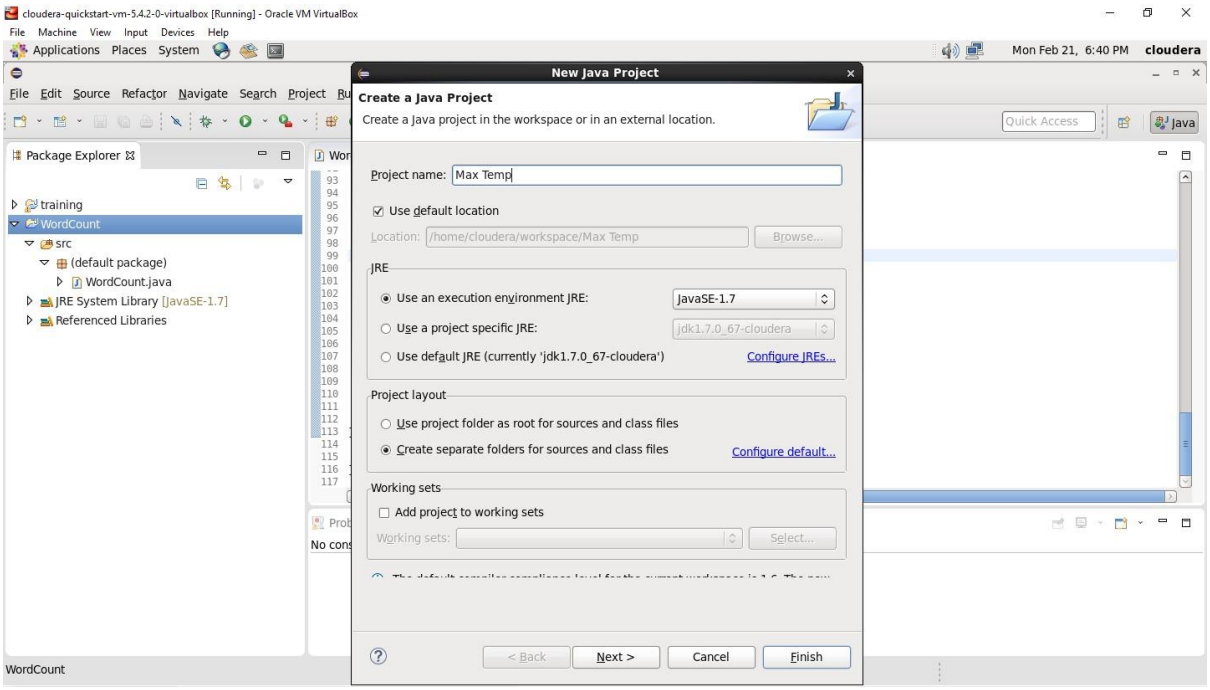


2) Open Eclipse present on the cloudera desktop

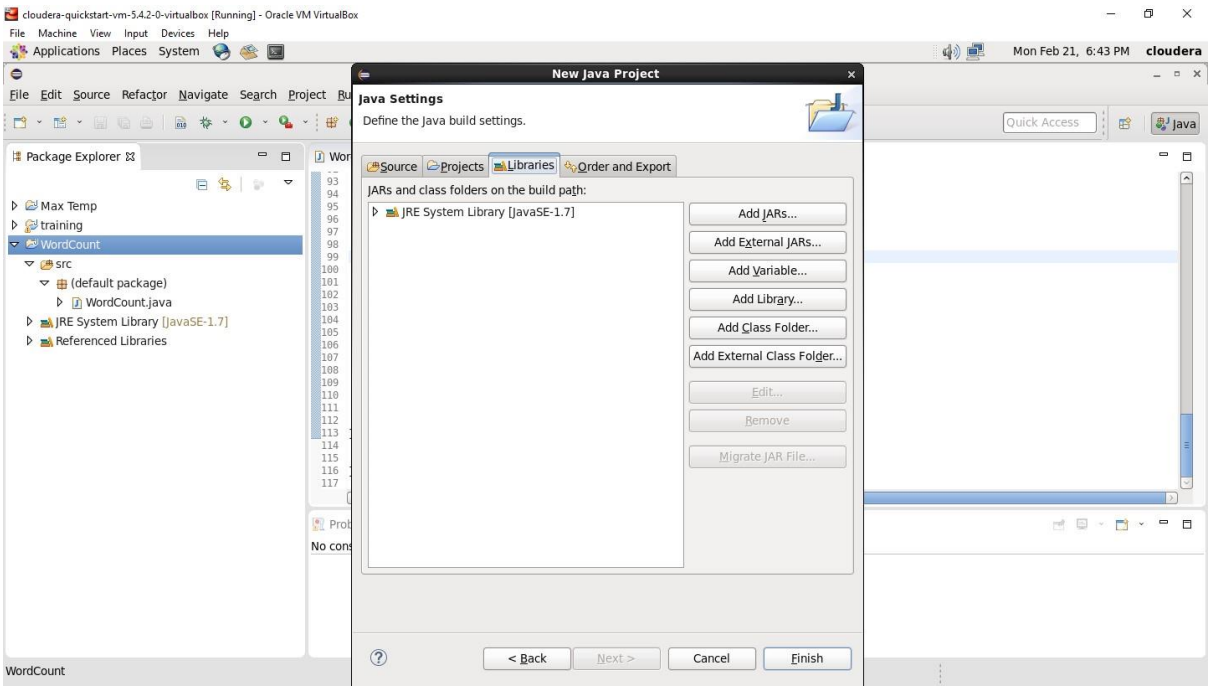


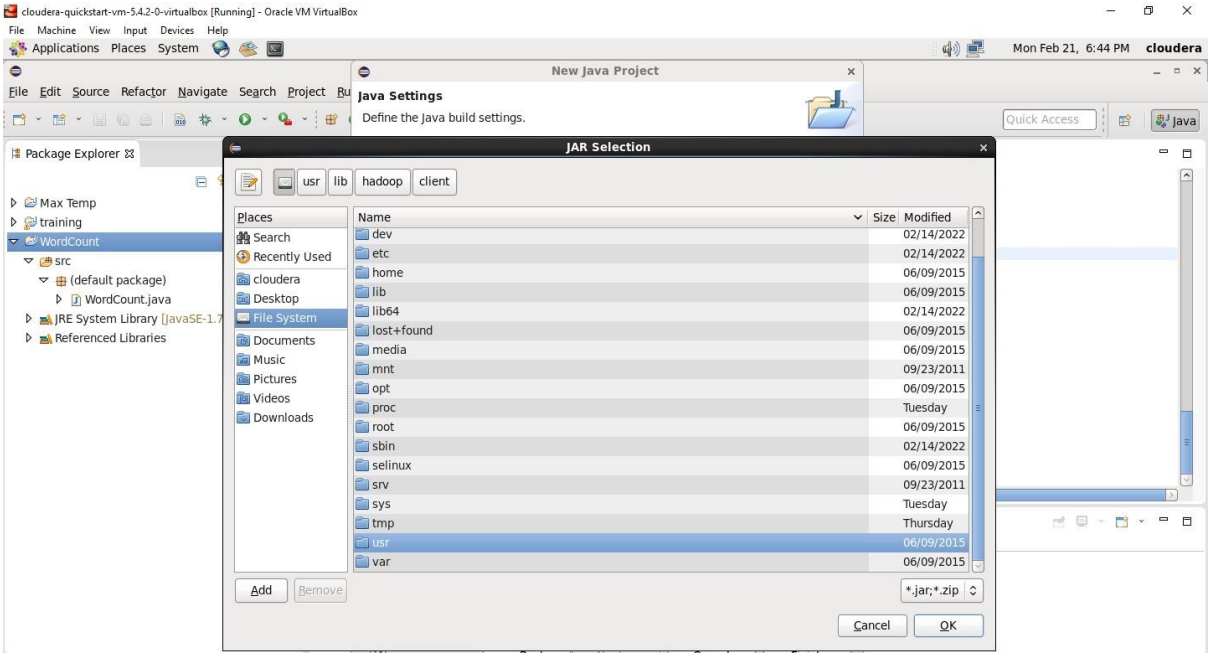
3) Create a new Java project clicking: File -> New -> Project -> Java Project -> Next ("MaxTemp" is the project name).

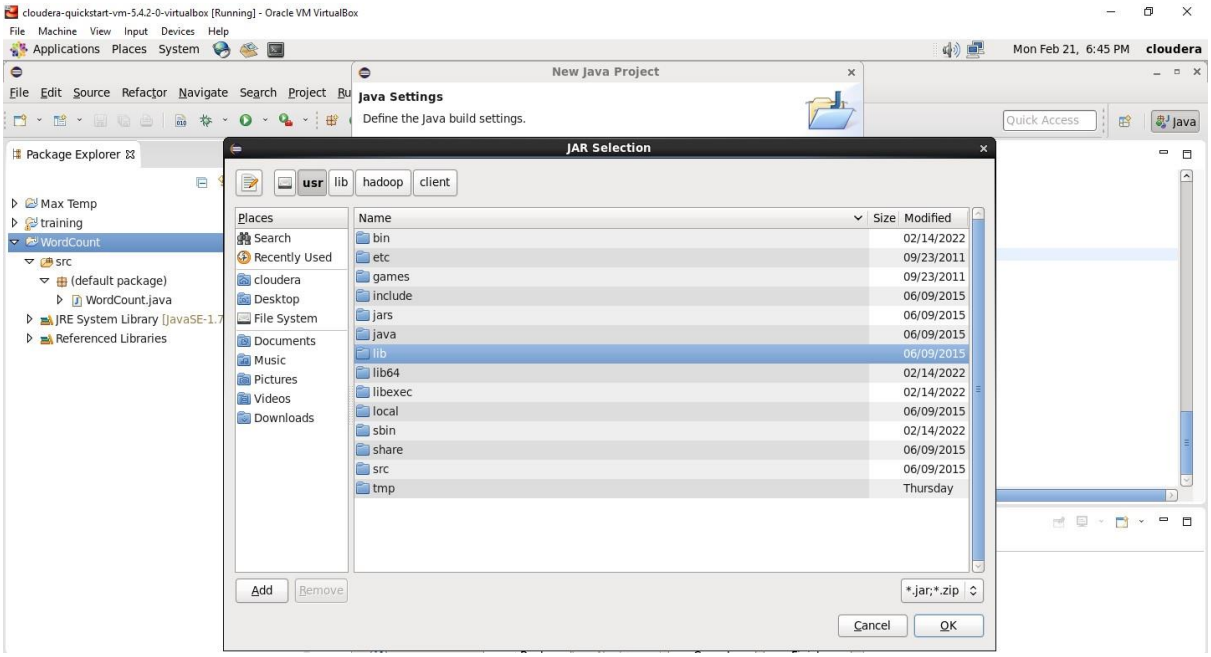


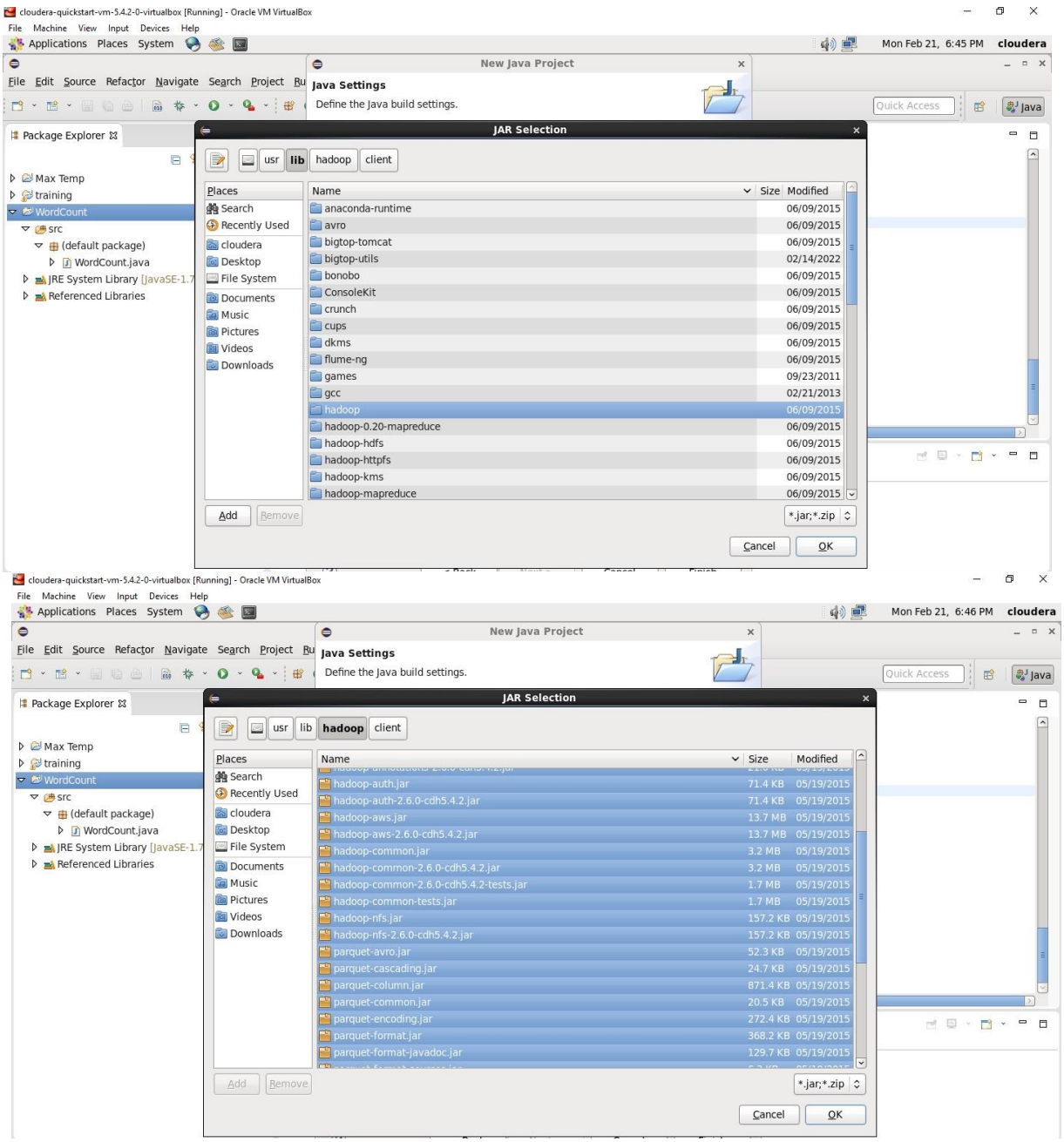


4) Adding the Hadoop libraries to the project Click on Libraries -> Add External JARs Click on File System -> usr -> lib -> hadoop Select all the libraries (JAR Files) -> click OK Click on Add External jars, -> client -> select all jar files -> ok -> Finish

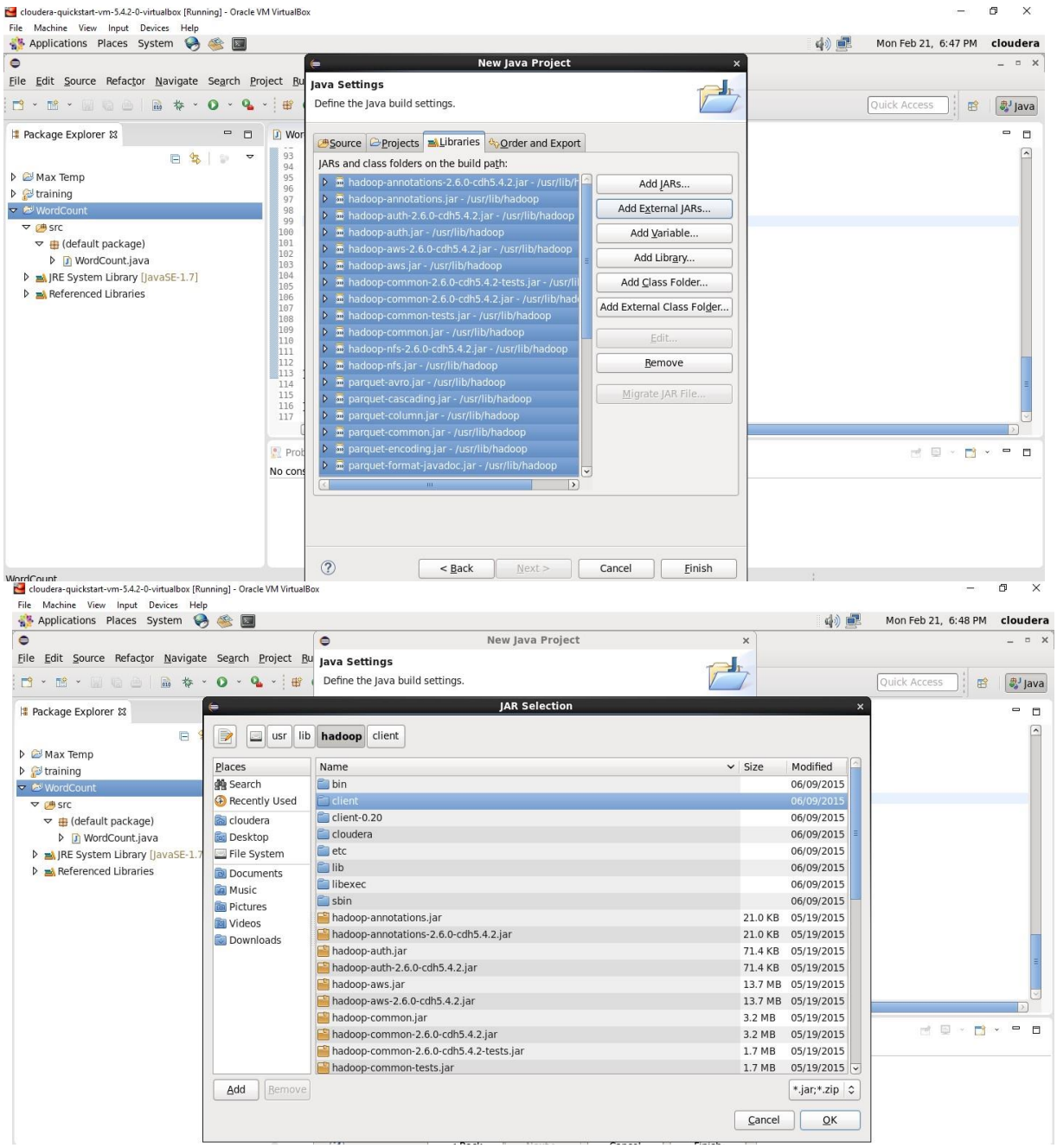



cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Feb 21, 6:44 PM cloudera
New Java Project
Java Settings
Define the Java build settings.
JAR Selection
usr lib hadoop client
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name Size Modified
dev 02/14/2022
etc 02/14/2022
home 06/09/2015
lib 06/09/2015
lib64 02/14/2022
lost+found 06/09/2015
media 06/09/2015
mnt 09/23/2011
opt 06/09/2015
proc Tuesday
root 06/09/2015
sbin 02/14/2022
selinux 06/09/2015
srv 09/23/2011
sys Tuesday
tmp Thursday
usr 06/09/2015
var 06/09/2015
Add Remove
.jar;.zip
Cancel OK
Package Explorer
Max Temp
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7
Referenced Libraries

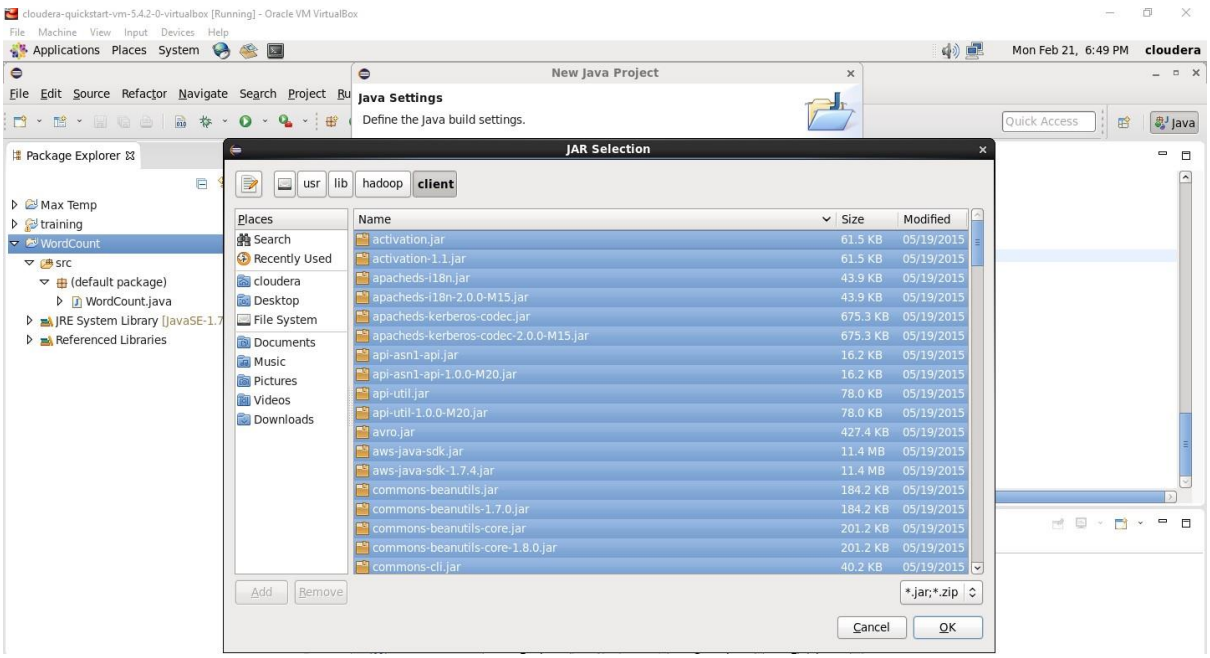

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Feb 21, 6:45 PM cloudera
New Java Project
Java Settings
Define the Java build settings.
JAR Selection
usr lib hadoop client
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name Size Modified
bin 02/14/2022
etc 09/23/2011
games 09/23/2011
include 06/09/2015
jars 06/09/2015
java 06/09/2015
lib 06/09/2015
lib64 02/14/2022
libexec 02/14/2022
local 06/09/2015
sbin 02/14/2022
share 06/09/2015
src 06/09/2015
tmp Thursday
Add Remove
.jar;.zip
Cancel OK
Package Explorer
Max Temp
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7
Referenced Libraries


cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Feb 21, 6:45 PM cloudera
New Java Project
Java Settings
Define the Java build settings.
JAR Selection
usr lib hadoop client
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name Size Modified
anaconda-runtime 06/09/2015
avro 06/09/2015
bigtop-tomcat 06/09/2015
bigtop-utils 02/14/2022
bonobo 06/09/2015
ConsoleKit 06/09/2015
crunch 06/09/2015
cups 06/09/2015
dkms 06/09/2015
flume-ng 06/09/2015
games 09/23/2011
gcc 02/21/2013
hadoop 06/09/2015
hadoop-0.20-mapreduce 06/09/2015
hadoop-hdfs 06/09/2015
hadoop-httpfs 06/09/2015
hadoop-kms 06/09/2015
hadoop-mapreduce 06/09/2015
Add Remove
.jar;.zip
Cancel OK
Package Explorer
Max Temp
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Quick Access
Java

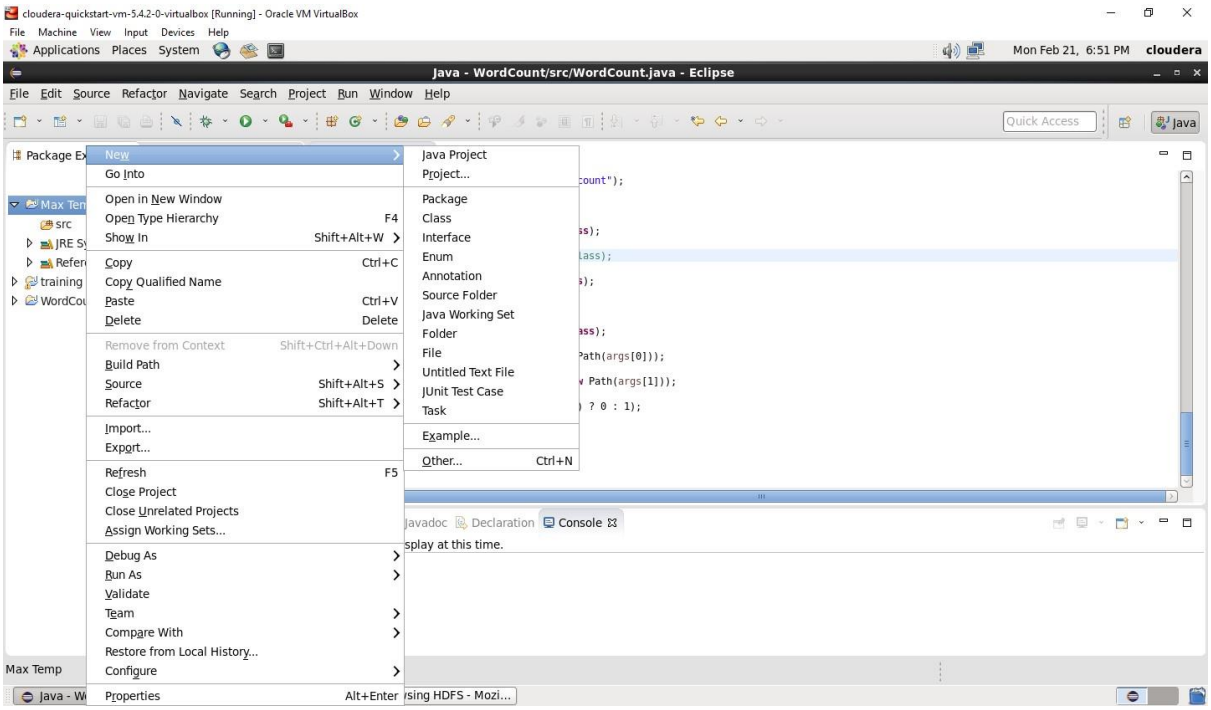
cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Feb 21, 6:46 PM cloudera
New Java Project
Java Settings
Define the Java build settings.
JAR Selection
usr lib hadoop client
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name Size Modified
hadoop-auth.jar 71.4 KB 05/19/2015
hadoop-auth-2.6.0-cdh5.4.2.jar 71.4 KB 05/19/2015
hadoop-aws.jar 13.7 MB 05/19/2015
hadoop-aws-2.6.0-cdh5.4.2.jar 13.7 MB 05/19/2015
hadoop-common.jar 3.2 MB 05/19/2015
hadoop-common-2.6.0-cdh5.4.2.jar 3.2 MB 05/19/2015
hadoop-common-2.6.0-cdh5.4.2-tests.jar 1.7 MB 05/19/2015
hadoop-common-tests.jar 1.7 MB 05/19/2015
hadoop-nfs.jar 157.2 KB 05/19/2015
hadoop-nfs-2.6.0-cdh5.4.2.jar 157.2 KB 05/19/2015
parquet-avro.jar 52.3 KB 05/19/2015
parquet-cascading.jar 24.7 KB 05/19/2015
parquet-column.jar 871.4 KB 05/19/2015
parquet-common.jar 20.5 KB 05/19/2015
parquet-encoding.jar 272.4 KB 05/19/2015
parquet-format.jar 368.2 KB 05/19/2015
parquet-format-javadoc.jar 129.7 KB 05/19/2015
Add Remove
.jar;.zip
Cancel OK
Package Explorer
Max Temp
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Quick Access
Java

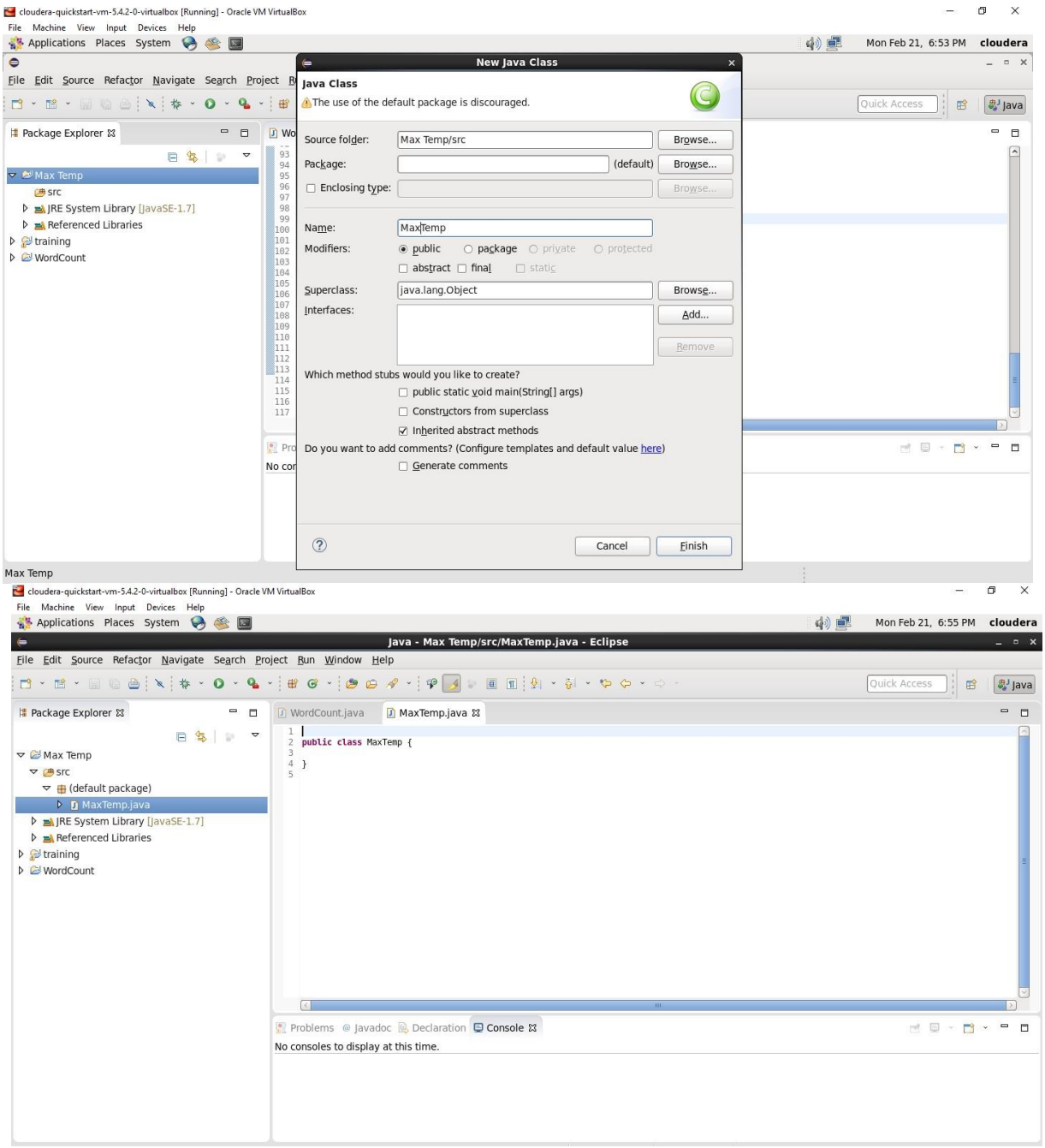

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Feb 21, 6:47 PM cloudera
New Java Project
Java Settings
Define the Java build settings.
Source Projects Libraries Order and Export
JARs and class folders on the build path:
hadoop-annotations-2.6.0-cdh5.4.2.jar - /usr/lib/hadoop
hadoop-annotations.jar - /usr/lib/hadoop
hadoop-auth-2.6.0-cdh5.4.2.jar - /usr/lib/hadoop
hadoop-auth.jar - /usr/lib/hadoop
hadoop-aws-2.6.0-cdh5.4.2.jar - /usr/lib/hadoop
hadoop-aws.jar - /usr/lib/hadoop
hadoop-common-2.6.0-cdh5.4.2-tests.jar - /usr/lib/hadoop
hadoop-common-2.6.0-cdh5.4.2.jar - /usr/lib/hadoop
hadoop-common-tests.jar - /usr/lib/hadoop
hadoop-common.jar - /usr/lib/hadoop
hadoop-nfs-2.6.0-cdh5.4.2.jar - /usr/lib/hadoop
hadoop-nfs.jar - /usr/lib/hadoop
parquet-avro.jar - /usr/lib/hadoop
parquet-cascading.jar - /usr/lib/hadoop
parquet-column.jar - /usr/lib/hadoop
parquet-common.jar - /usr/lib/hadoop
parquet-encoding.jar - /usr/lib/hadoop
parquet-format-javadoc.jar - /usr/lib/hadoop
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
< Back
Next >
Cancel
Finish
Package Explorer
Max Temp
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Quick Access
Java

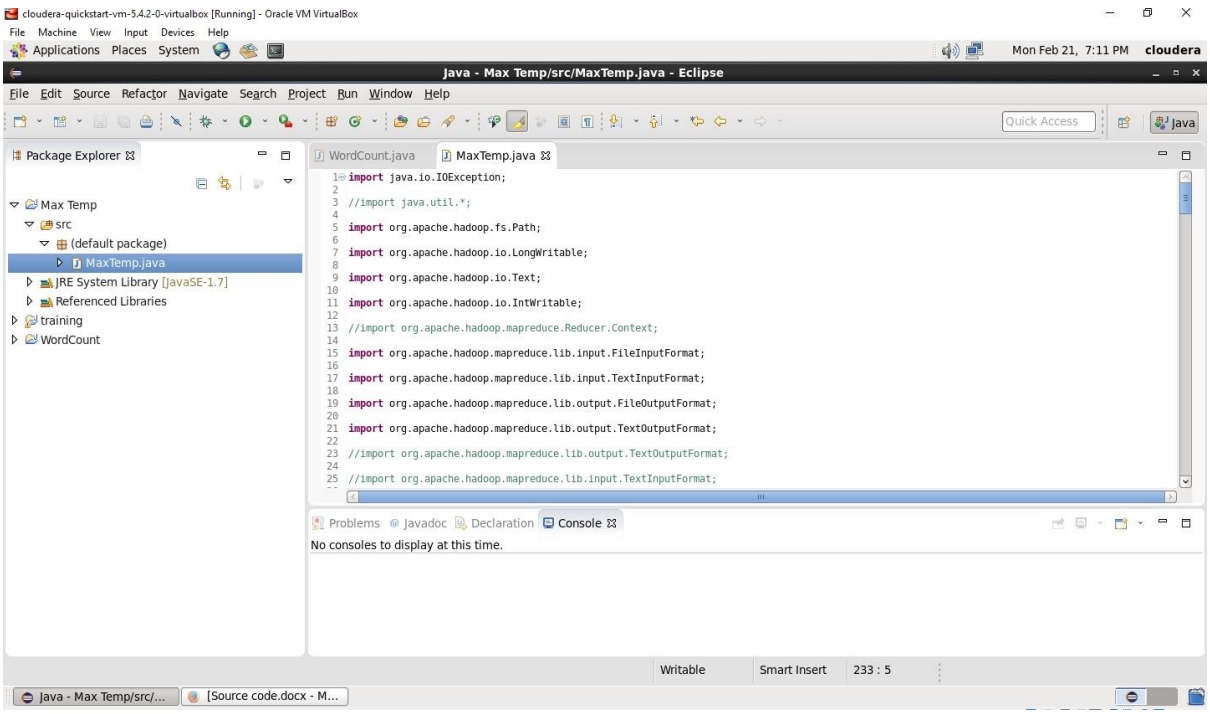
cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Feb 21, 6:48 PM cloudera
New Java Project
Java Settings
Define the Java build settings.
JAR Selection
usr lib hadoop client
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name Size Modified
bin 06/09/2015
client 06/09/2015
client-0.20 06/09/2015
cloudera 06/09/2015
etc 06/09/2015
lib 06/09/2015
libexec 06/09/2015
sbin 06/09/2015
hadoop-annotations.jar 21.0 KB 05/19/2015
hadoop-annotations-2.6.0-cdh5.4.2.jar 21.0 KB 05/19/2015
hadoop-auth.jar 71.4 KB 05/19/2015
hadoop-auth-2.6.0-cdh5.4.2.jar 71.4 KB 05/19/2015
hadoop-aws.jar 13.7 MB 05/19/2015
hadoop-aws-2.6.0-cdh5.4.2.jar 13.7 MB 05/19/2015
hadoop-common.jar 3.2 MB 05/19/2015
hadoop-common-2.6.0-cdh5.4.2.jar 3.2 MB 05/19/2015
hadoop-common-2.6.0-cdh5.4.2-tests.jar 1.7 MB 05/19/2015
hadoop-common-tests.jar 1.7 MB 05/19/2015
Add
Remove
.jar;.zip
Cancel
OK
Package Explorer
Max Temp
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Quick Access
Java



5) Right Click on the name of Project “MaxTemp” -> New -> class “MaxTemp” -> Finish Then MaxTemp.java window will pop up




cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Mon Feb 21, 6:53 PM cloudera
New Java Class
Java Class
The use of the default package is discouraged.
Source folder: Max Temp/src
Package: (default)
Enclosing type:
Name: MaxTemp
Modifiers: public package private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Cancel
Finish
Package Explorer
Max Temp
src
JRE System Library [JavaSE-1.7]
Referenced Libraries
training
WordCount
Max Temp
cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
Mon Feb 21, 6:55 PM cloudera
Java - Max Temp/src/MaxTemp.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
WordCount.java
MaxTemp.java
public class MaxTemp {
}
(default package)
MaxTemp.java
Problems Javadoc Declaration Console
No consoles to display at this time.



Source code:

```
import java.io.IOException;
//import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.IntWritable;
//import org.apache.hadoop.mapreduce.Reducer.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
//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;
//import MaximumTemp.MaxTemperatureMapper;
//import MaxTemp.MaxTemperatureReducer;
public class MaxTemp {
```

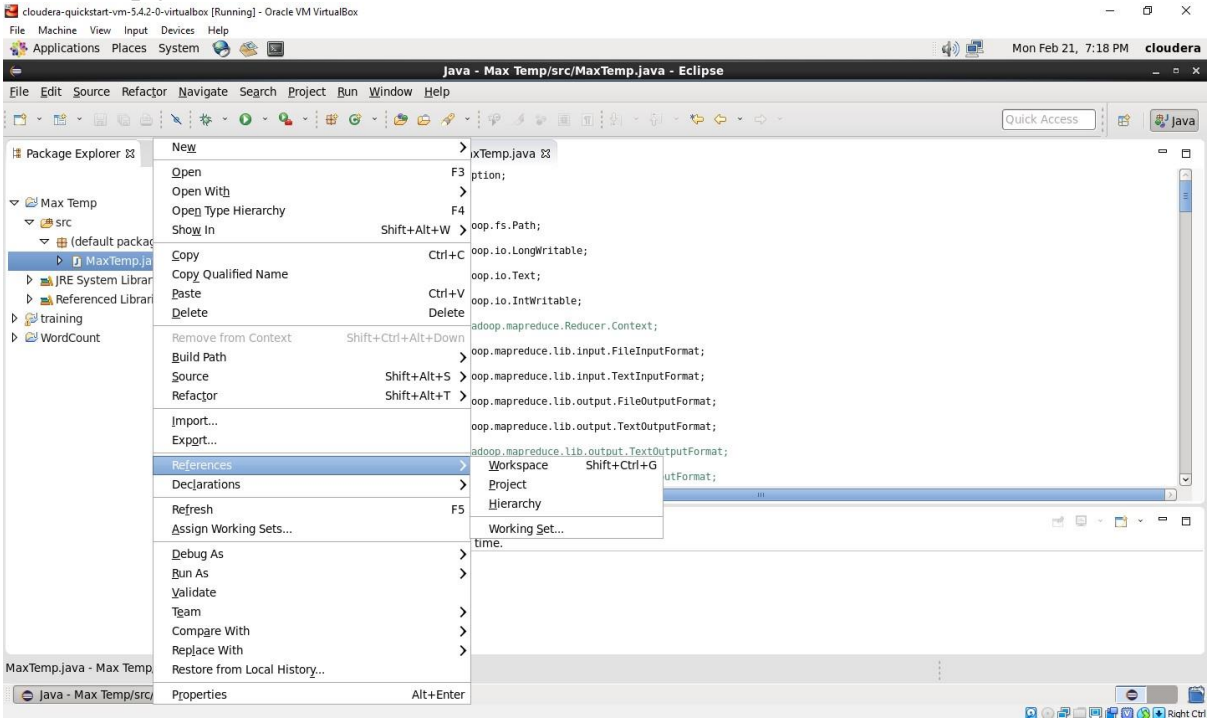
```
// Mapper
/*MaxTemperatureMapper class is static
* And extends Mapper abstract class
* having four Hadoop generics type
* Long Writable, Text, Text, Text.
*/
public static class MaxTemperatureMapper extends
Mapper<LongWritable, Text, Text, IntWritable> {
// the data in our data set with
// this value is inconsistent data
//public static final int MISSING = 9999;

public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
String line=value.toString();
String year=line.substring(15, 19);
int airtemp;
if(line.charAt(87)=='+')
{
airtemp=Integer.parseInt(line.substring(88,92));
}
else
airtemp=Integer.parseInt(line.substring(87,92));
String q=line.substring(92,93);
if(airtemp!=9999 && q.matches("[01459]"))
{

context.write(new Text(year), new IntWritable(airtemp));
}
}
}
// Reducer
/*MaxTemperatureReducer class is static
and extends Reducer abstract class
```

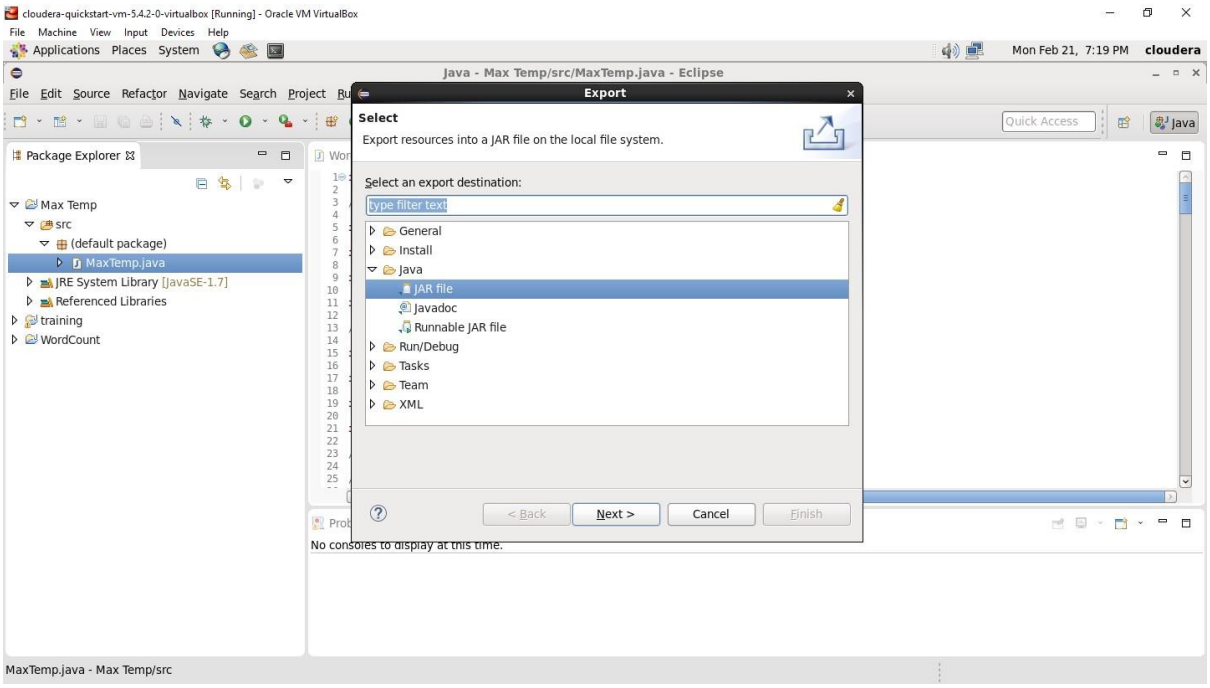
```
having four Hadoop generics type
Text, Text, Text, Text.
*/
public static class MaxTemperatureReducer extends
Reducer<Text, IntWritable, Text, IntWritable> {
/**
* @method reduce
* This method takes the input as key and
* list of values pair from the mapper,
* it does aggregation based on keys and
* produces the final context.
*/
public void reduce(Text key,Iterable<IntWritable> values, Context
context)
throws IOException, InterruptedException {
int maxvalue= Integer.MIN_VALUE;
for (IntWritable value : values) {
maxvalue=Math.max(maxvalue, value.get());
}
context.write(key, new IntWritable(maxvalue));
}
}

/**
* @method main
* This method is used for setting
* all the configuration properties.
* It acts as a driver for map-reduce
* code.
*/
public static void main(String[] args) throws Exception {
// reads the default configuration of the
// cluster from the configuration XML files
Configuration conf = new Configuration();
```

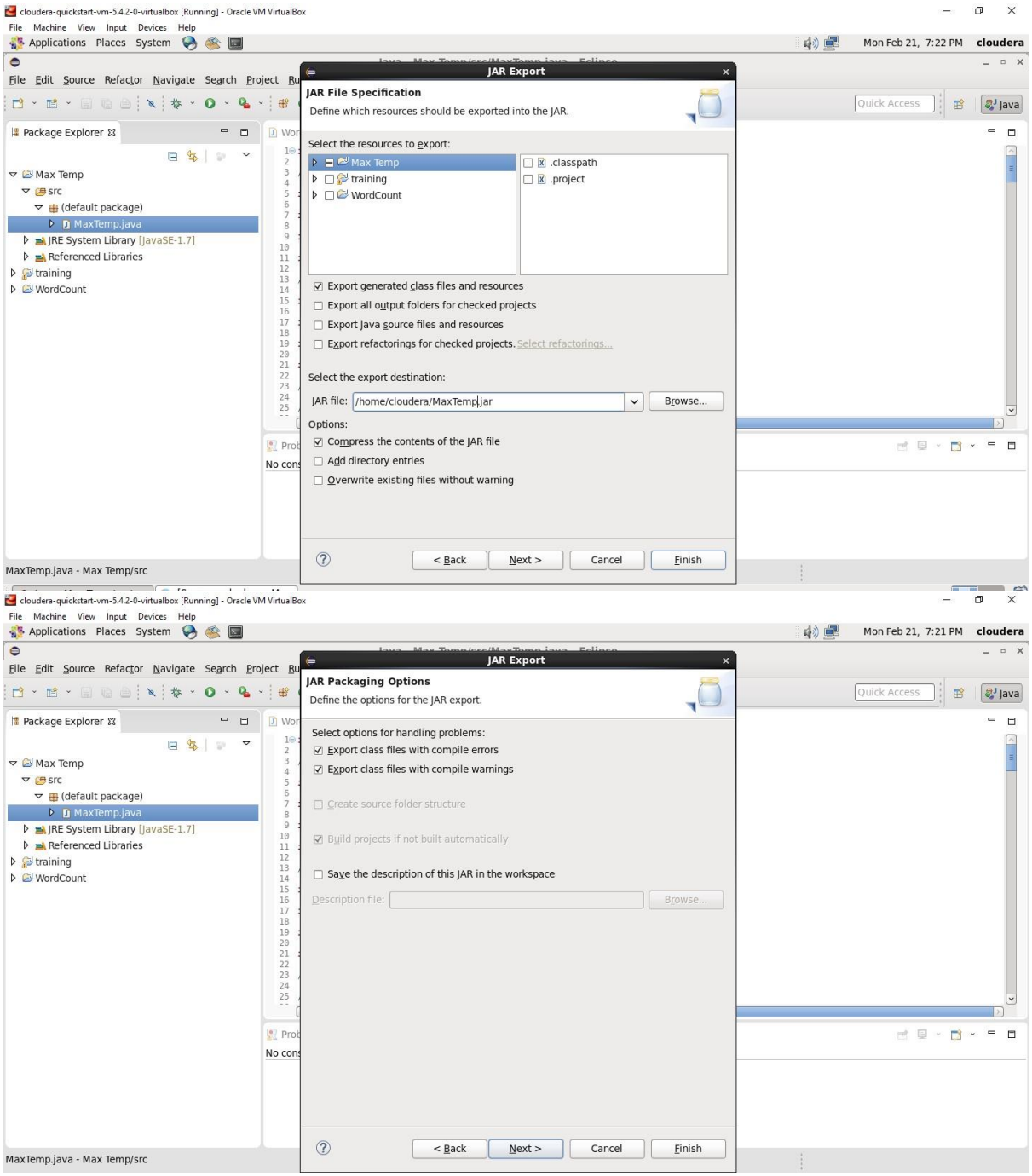
```
// initializing the job with the
// default configuration of the cluster
// Job job = new Job(conf, "weather example");
Job job = Job.getInstance(conf, "weather example");
// Assigning the driver class name
job.setJarByClass(MaxTemp.class);
// Key type coming out of mapper
// job.setMapOutputKeyClass(Text.class);
// 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);
// Defining output Format class which is
// responsible to parse the dataset
// into a key value pair
job.setOutputFormatClass(TextOutputFormat.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);
}
}
```

6) Right Click on the project name MaxTemp -> Export -> Java -> JAR File -> Next -> for select the export destination for JAR file: browse -> Name :

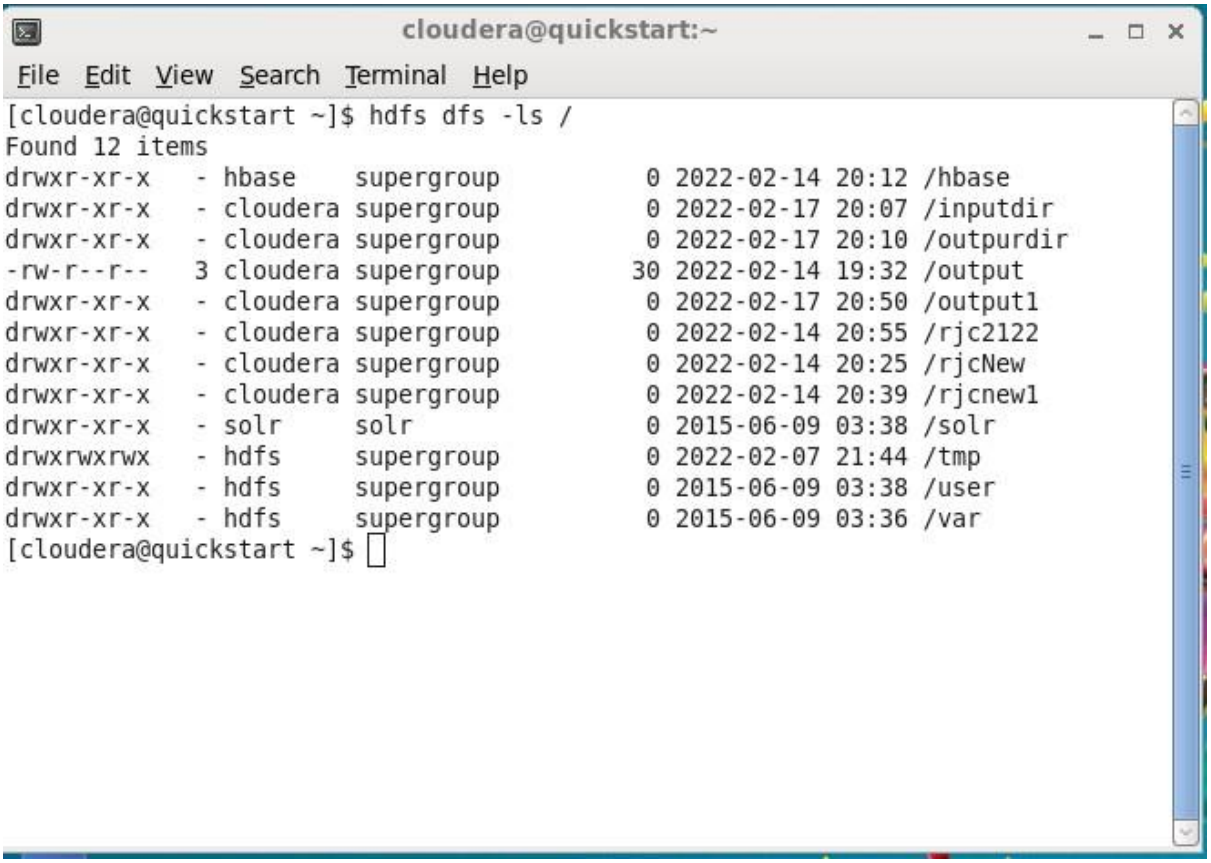
MaxTemp.jar -> save in folder -> cloudera -> Finish -> OK





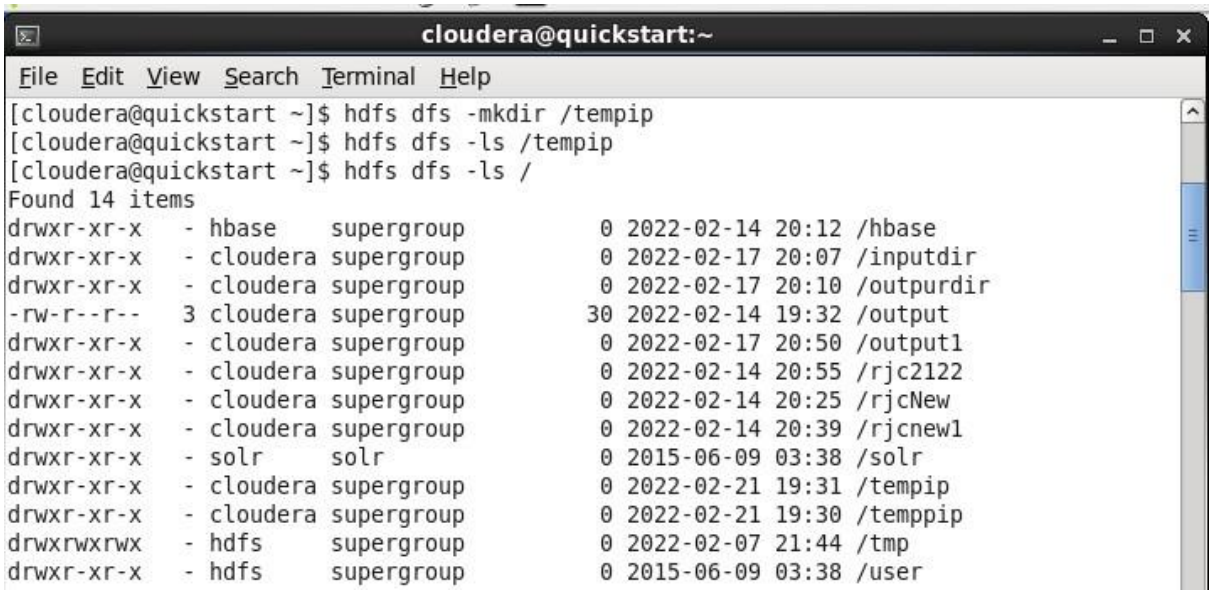


7) Open terminal and type hdfs dfs -ls/ command Here listing all the directory present in hdfs.



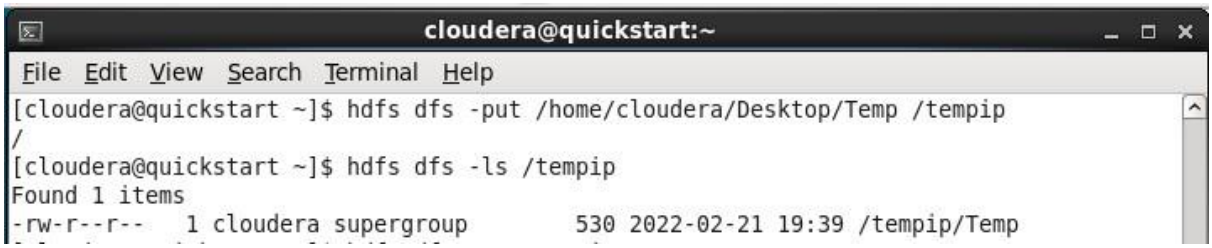
8) Input file named as Temp which is present on desktop i.e. in local file system.

9) Now we have to move this input file to hdfs. For this we create a direcory on hdfs using command **hdfs dfs -mkdir /tempip**.

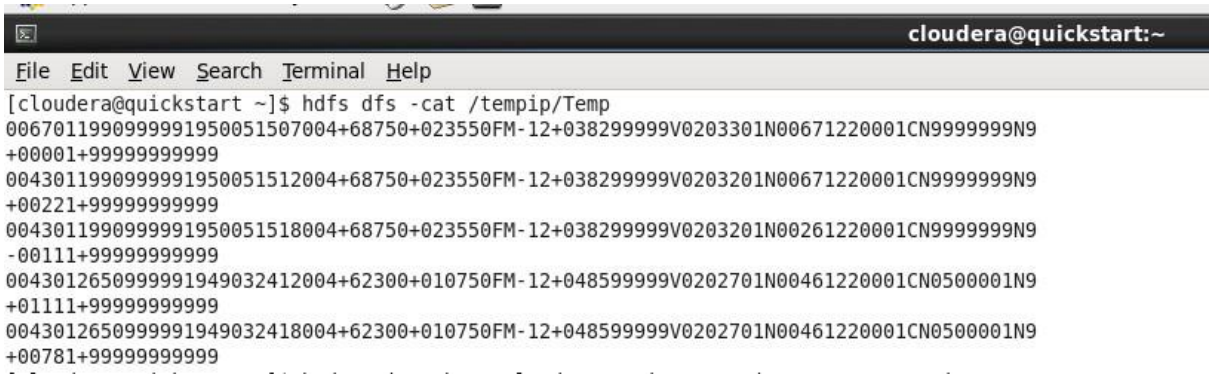


10) Move the input file i.e. temperature to this directory created in hdfs by using either put command or copyFromLocal command.

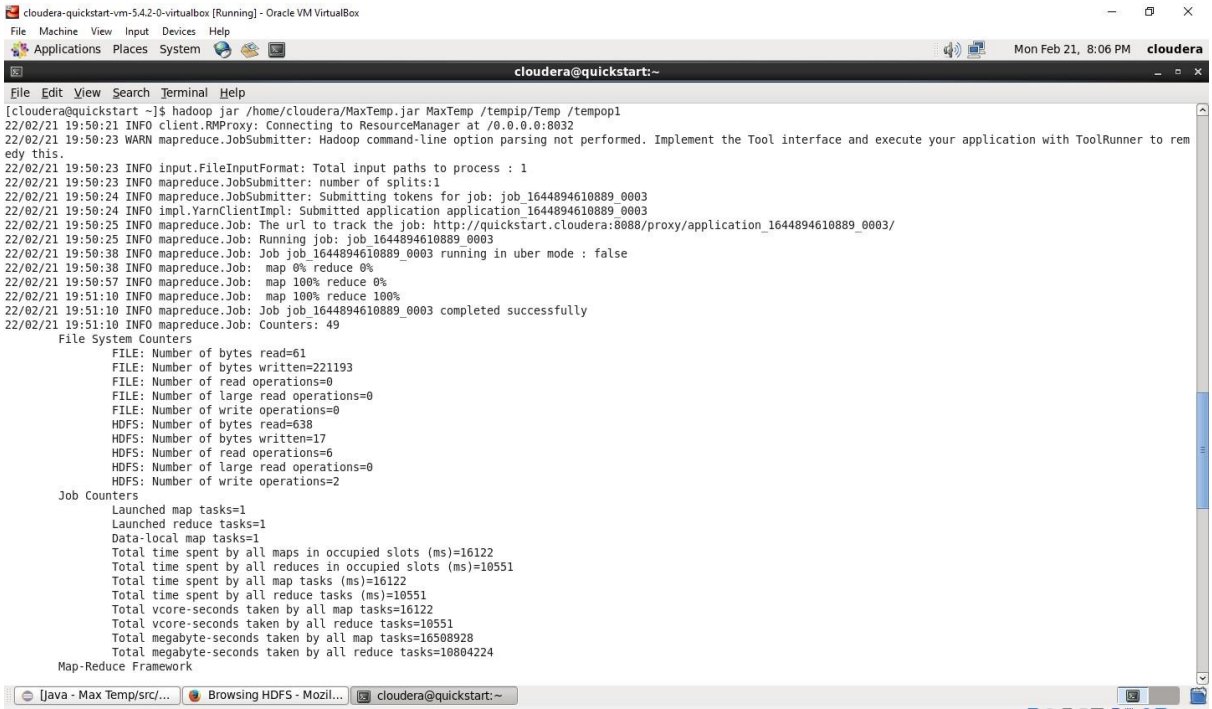
11) Now checking whether the **"Temp"** present in /tempip directory of hdfs or not using **hdfs dfs -ls /tempip** command



12) As we can see **"Temp"** file is present in /tempip directory of hdfs. Now we will see the content of this file using **hdfs dfs –cat /tempip/temperature** command



13) Running Mapreduce Program on Hadoop, syntax is hadoop jar jarFileName.jar ClassName /InputFileAddress /outputdir



Map-Reduce Framework

14) Then we can verify the content of tempop1 directory and in that part-r file has the actual output by using the command Hdfs dfs -cat /tempop1/part-r-00000 This will give us final output. The same file can also be accessed using a browser. For every execution of this program we need to delete the output directory or give a new name to the output directory every time.1st we are checking whether the tempop1 directory is created in hdfs or not using command **hdfs dfs -ls /**

15) Now let's check what we have inside this **tempop1** directory using command as **hdfs dfs -ls /tempop1**

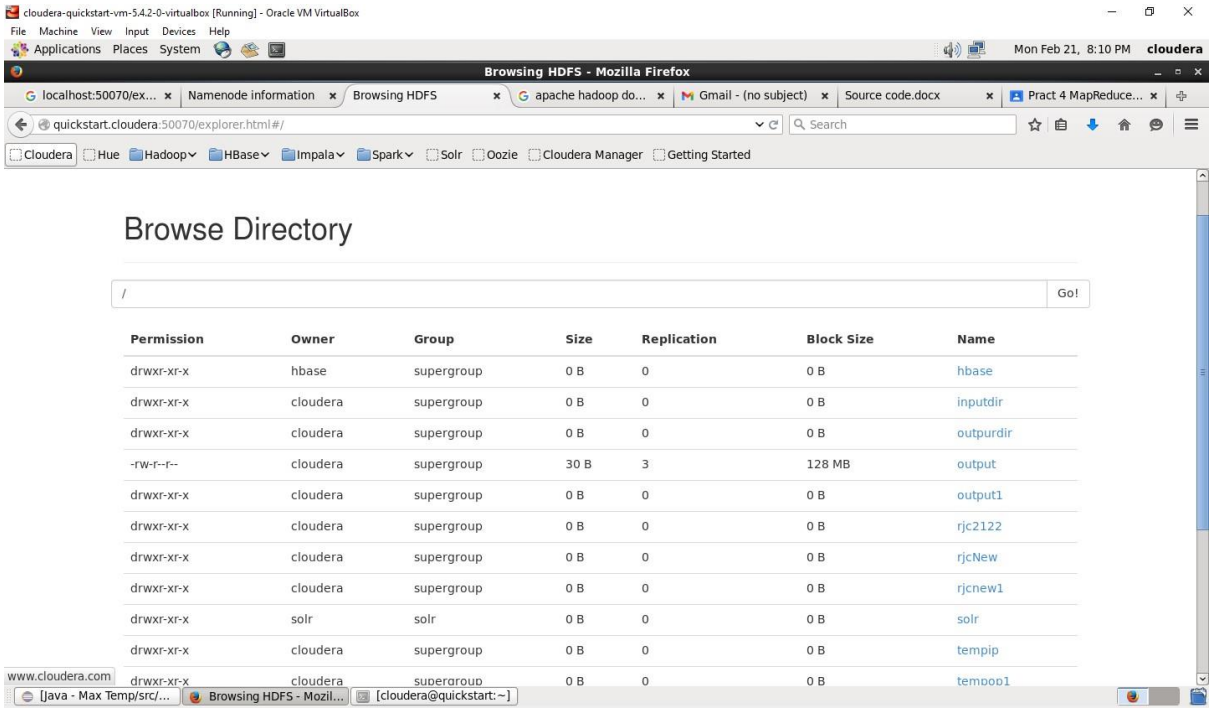
16) Now we want to read the content of the **part-r-00000** file which present inside the **tempop1** using command **hdfs dfs -cat /tempop1/part-r-00000**

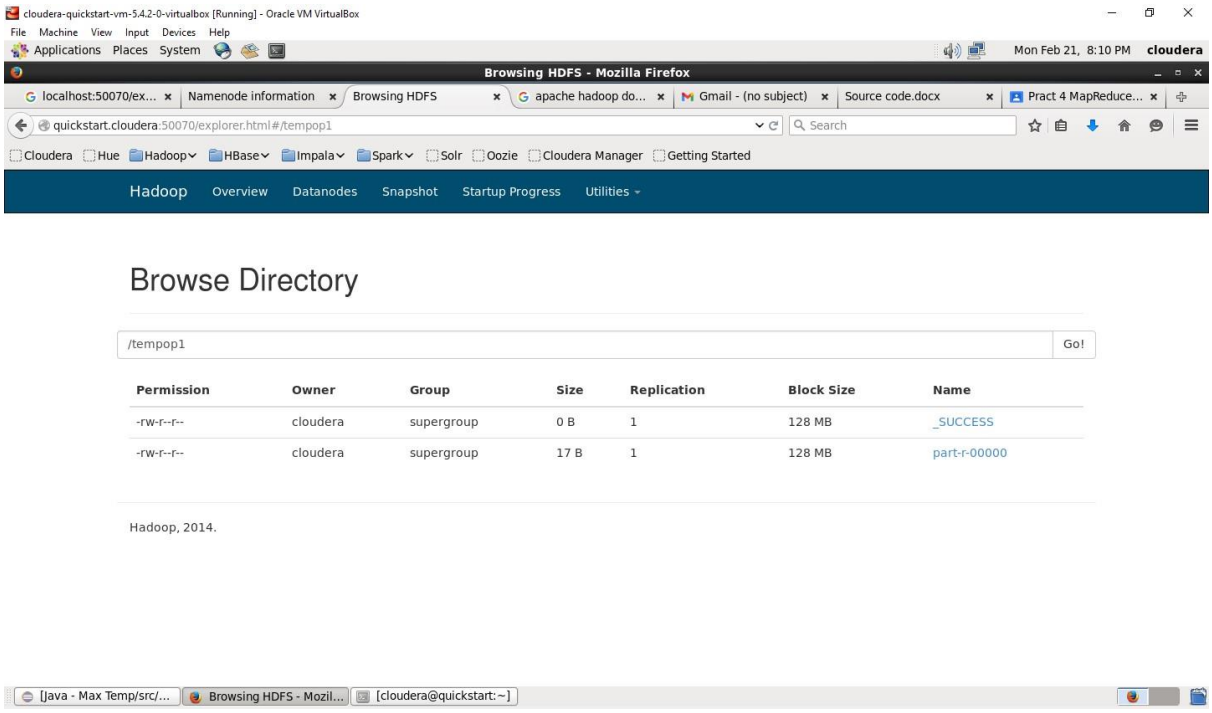
```
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 15 items
drwxr-xr-x   - hbase    supergroup          0 2022-02-14 20:12 /hbase
drwxr-xr-x   - cloudera supergroup          0 2022-02-17 20:07 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2022-02-17 20:10 /outpurdir
-rw-r--r--   3 cloudera supergroup         30 2022-02-14 19:32 /output
drwxr-xr-x   - cloudera supergroup          0 2022-02-17 20:50 /output1
drwxr-xr-x   - cloudera supergroup          0 2022-02-14 20:55 /rjc2122
drwxr-xr-x   - cloudera supergroup          0 2022-02-14 20:25 /rjcNew
drwxr-xr-x   - cloudera supergroup          0 2022-02-14 20:39 /rjcnew1
drwxr-xr-x   - solr     solr                0 2015-06-09 03:38 /solr
drwxr-xr-x   - cloudera supergroup          0 2022-02-21 19:39 /tempip
drwxr-xr-x   - cloudera supergroup          0 2022-02-21 19:51 /tempop1
drwxr-xr-x   - cloudera supergroup          0 2022-02-21 19:30 /temppip
drwxrwxrwx   - hdfs     supergroup          0 2022-02-07 21:44 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2015-06-09 03:38 /user
drwxr-xr-x   - hdfs     supergroup          0 2015-06-09 03:36 /var
[cloudera@quickstart ~]$ hdfs dfs -ls /tempop1
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2022-02-21 19:51 /tempop1/_SUCCESS
-rw-r--r--   1 cloudera supergroup         17 2022-02-21 19:51 /tempop1/part-r-00000
[cloudera@quickstart ~]$ hdfs dfs -cat /tempop1/part-r-00000
1949    111
1950    22
[cloudera@quickstart ~]$
```

So the maximum temperature for the year 1949 is 111 and for the year 1950 is 22.

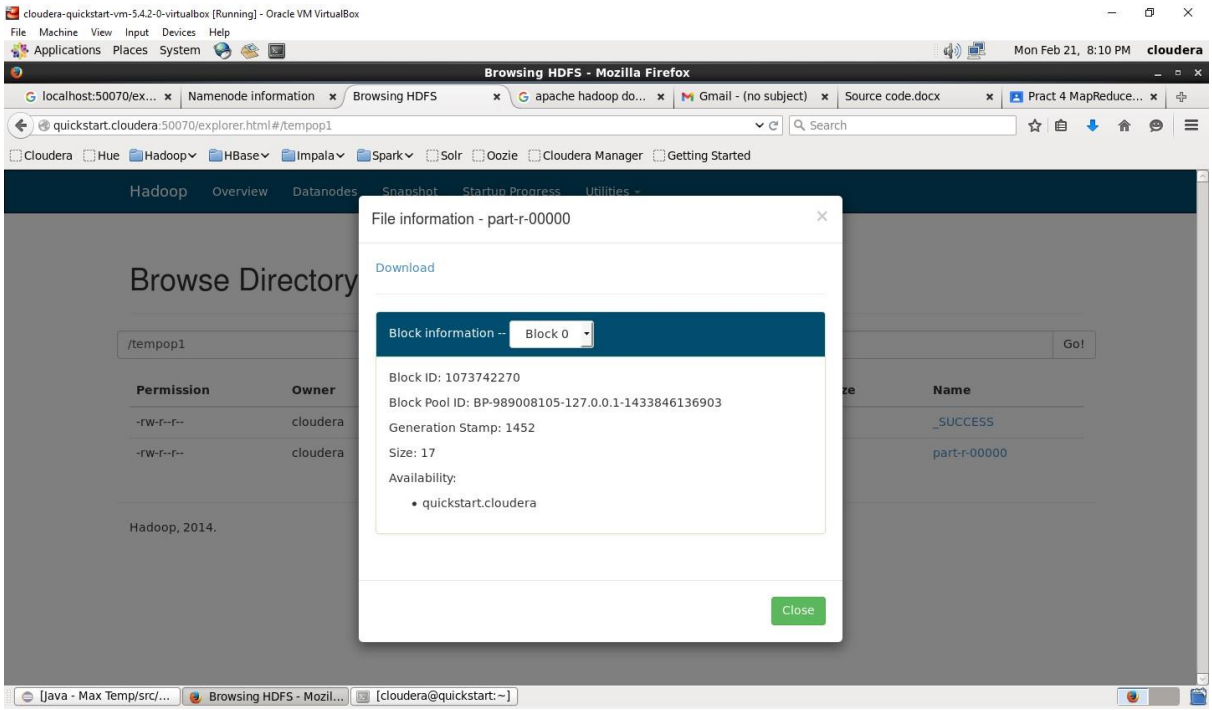
18) Browse the Directory by

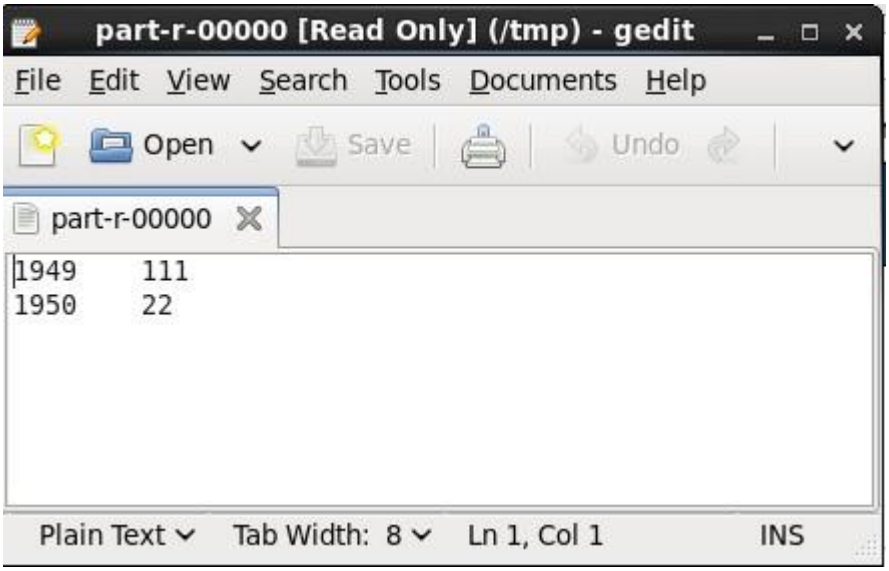
Hadoop->HDFS Namenode->Ultilities ->Browse the file system





Now downloading the **part-r-00000** file**.**




part-r-00000 [Read Only] (/tmp) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
part-r-00000
1949 111
1950 22
Plain Text
Tab Width: 8
Ln 1, Col 1
INS