http://www.nagazuka.nl/2013/06/running-hadoop-examples-on-cloudera.html

1. Open a Terminal (Right-click on Desktop or click Terminal icon in the top toolbar)

2. Navigate to the Hadoop library directory:
```
cd /usr/lib/hadoop-mapreduce/
```

3. Execute the Hadoop jar command to run the WordCount example:
```
hadoop jar hadoop-mapreduce-examples.jar wordcount
```

4. The wordcount example complains that it needs input and output parameters.
```
 Usage: wordcount <in> <out>
```

5. Create one or more text files with a few words in it for testing, or use a log file:
```
echo "count these words for me hadoop" > /home/cloudera/file1
echo "hadoop counts words for me" > /home/cloudera/file2
```

6. Create a directory on the HDFS file system:
```
hdfs dfs -mkdir /user/cloudera/input
```

7. Copy the files from local filesystem to the HDFS filesystem:
```
hdfs dfs -put /home/cloudera/file1 /user/cloudera/input
hdfs dfs -put /home/cloudera/file2 /user/cloudera/input
```

8. Run the Hadoop WordCount example with the input and output specified:
```
hadoop jar hadoop-mapreduce-examples.jar wordcount /user/cloudera/input
/user/cloudera/output
```

9. Hadoop prints out a whole lot of logging information, after completion view the output directory:
```
hdfs dfs -ls /user/cloudera/output
```

10. Check the output file to see the results:
```
hdfs dfs -cat /user/cloudera/output/part-r-00000
```

11. To delete all files from a folder use:

```
hdfs dfs –rm -r /user/cloudera/output
```

For more fun and games you can run other example applications. Just run the Hadoop jar command without specifying which example to run, you'll see the complete list of examples available:

```
hadoop jar hadoop-mapreduce-examples.jar

Valid program names are:
  aggregatewordcount: An Aggregate based map/reduce program that counts the words
in the input files.
  aggregatewordhist: An Aggregate based map/reduce program that computes the
histogram of the words in the input files.
  dbcount: An example job that count the pageview counts from a database.
```

grep: A map/reduce program that counts the matches of a regex in the input.
join: A job that effects a join over sorted, equally partitioned datasets
multifilewc: A job that counts words from several files.
pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
pi: A map/reduce program that estimates Pi using monte-carlo method.
randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
randomwriter: A map/reduce program that writes 10GB of random data per node.
secondarysort: An example defining a secondary sort to the reduce.
sleep: A job that sleeps at each map and reduce task.
sort: A map/reduce program that sorts the data written by the random writer.
sudoku: A sudoku solver.
teragen: Generate data for the terasort
terasort: Run the terasort
teravalidate: Checking results of terasort
wordcount: A map/reduce program that counts the words in the input files.