1. Command for compiling Java file:

javac -cp src/:hadoop-common-2.6.1.jar:hadoop-mapreduce-client-core-2.6.1.jar:commons-cli-2.0.jar -d . MapFileWriteDemo.java

Command for running Java in Hadoop

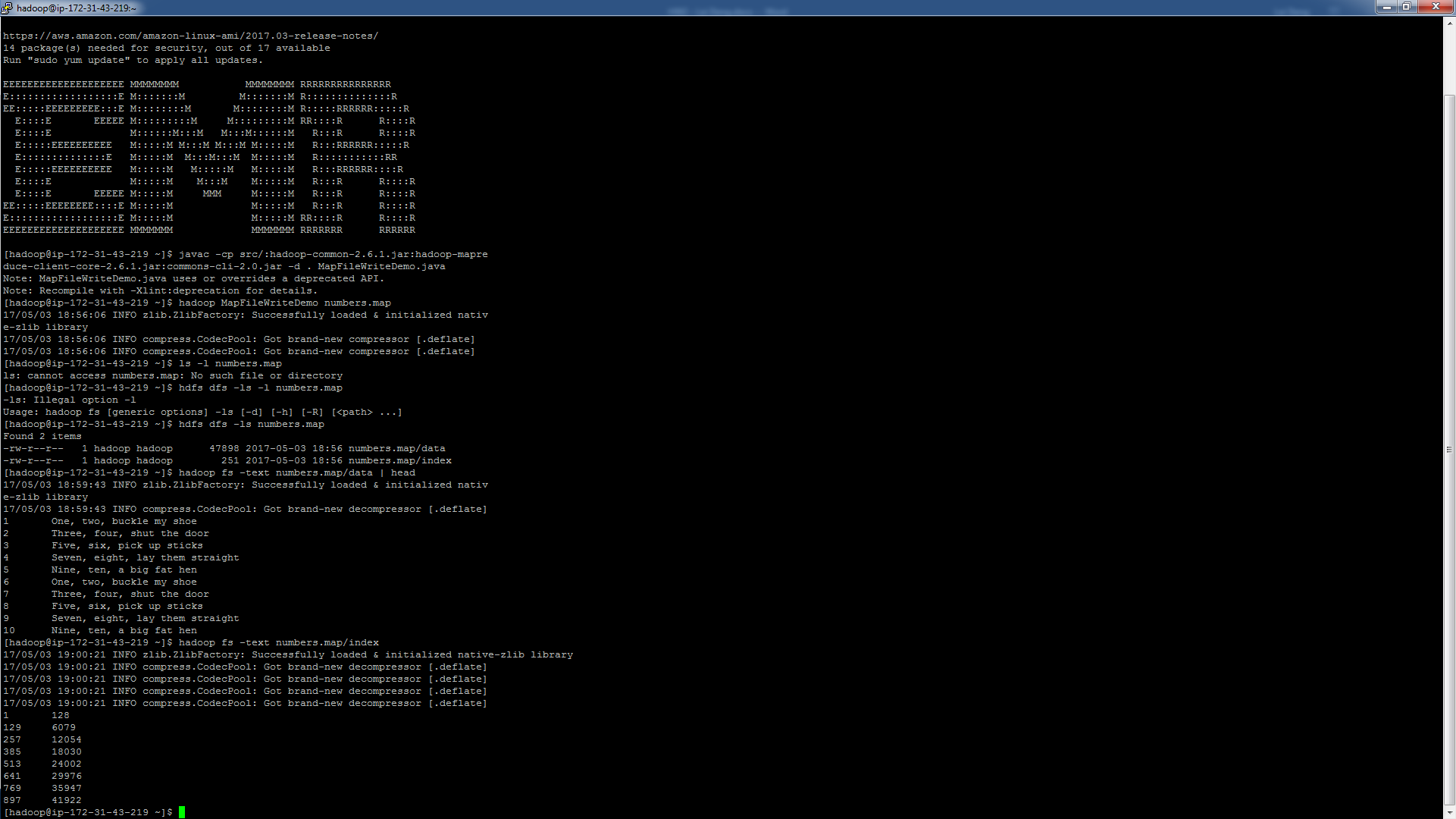
hadoop MapFileWriteDemo numbers.map

hdfs dfs -ls numbers.map

hadoop fs -text numbers.map/data | head

hadoop fs -text numbers.map/index

Screenshot of final results in EMR:



2.

2.1 Command for compiling Java file

javac -cp src/:hadoop-common-2.6.1.jar:hadoop-mapreduce-client-core-2.6.1.jar:commons-cli-2.0.jar -d . SequenceFileWriteDemo.java

2.2 Command for running Java in Hadoop

2.2.1 writing a sequencefile:

hadoop SequenceFileWriteDemo numbers.seq

2.2.2 sort the sequence file numbers.seq into a new directory called number.map2:

hadoop jar hadoop-mapreduce-examples-2.2.0.jar sort -r 1 \

-inFormat org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat \

-outFormat org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat \

-outKey org.apache.hadoop.io.IntWritable \

-outValue org.apache.hadoop.io.Text \

numbers.seq numbers.map2

2.2.3 check the numbers.map2 file exist:

hdfs dfs -ls numbers.map2

2.2.4 rename the MapReduce output to be the data file:

hadoop fs -mv numbers.map2/part-r-00000 numbers.map2/data

2.2.5 compile java file:

javac -cp src/:hadoop-common-2.6.1.jar:hadoop-mapreduce-client-core-2.6.1.jar:commons-cli-2.0.jar -d . MapFileFixer.java

2.2.6 create the index file:

hadoop MapFileFixer numbers.map2

Screenshot of final results in EMR:

