1. 실행 환경

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfsadmin -report
Configured Capacity: 58649149440 (54.62 GB)
Present Capacity: 35344531456 (32.92 GB)
DFS Remaining: 35340173312 (32.91 GB)
DFS Used: 4358144 (4.16 MB)
DFS Used%: 0.01%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (4):

Name: 192.168.56.101:50010 (hadoop01)
Hostname: hadoop01
Decommission Status : Normal
Configured Capacity: 14662287360 (13.66 GB)
DFS Used: 1089536 (1.04 MB)
Non DFS Used: 6317424640 (5.88 GB)
DFS Remaining: 8343773184 (7.77 GB)
DFS Used%: 0.01%
DFS Remaining%: 56.91%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Mon May 22 23:38:03 KST 2017

Name: 192.168.56.102:50010 (hadoop02)
Hostname: hadoop02
Decommission Status : Normal
Configured Capacity: 14662287360 (13.66 GB)
DFS Used: 1089536 (1.04 MB)
Non DFS Used: 5663068160 (5.27 GB)
DFS Remaining: 8998129664 (8.38 GB)
DFS Used%: 0.01%
DFS Remaining%: 61.37%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Mon May 22 23:38:02 KST 2017
```

```
Name: 192.168.56.103:50010 (hadoop03)
Hostname: hadoop03
Decommission Status : Normal
Configured Capacity: 14662287360 (13.66 GB)
DFS Used: 1089536 (1.04 MB)
Non DFS Used: 5662048256 (5.27 GB)
DFS Remaining: 8999149568 (8.38 GB)
DFS Used%: 0.01%
DFS Remaining%: 61.38%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Mon May 22 23:38:02 KST 2017

Name: 192.168.56.104:50010 (hadoop04)
Hostname: hadoop04
Decommission Status : Normal
Configured Capacity: 14662287360 (13.66 GB)
DFS Used: 1089536 (1.04 MB)
Non DFS Used: 5662076928 (5.27 GB)
DFS Remaining: 8999120896 (8.38 GB)
DFS Used%: 0.01%
DFS Remaining%: 61.38%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Mon May 22 23:38:02 KST 2017

hadoop@hadoop02:~/hadoop-2.7.3$ bin/yarn node -list
17/05/22 23:43:19 INFO client.RMProxy: Connecting to ResourceManager at hadoop02/192.168.56.102:8032
Total Nodes:4
         Node-Id             Node-State Node-Http-Address    Number-of-Running-Containers
  hadoop02:42287                RUNNING     hadoop02:8042                               0
  hadoop04:43969                RUNNING     hadoop04:8042                               0
  hadoop01:36421                RUNNING     hadoop01:8042                               0
  hadoop03:41169                RUNNING     hadoop03:8042                               0
```

2. WordCount V1

2-1. WordCount.java 작성

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

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {
   public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable>{
      private final static IntWritable one = new IntWritable(1);
      private Text word = new Text();

      public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
         while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
         }
      }
   }

   public static class IntSumReducer extends Reducer<Text,IntWritable,Text,IntWritable> {
      private IntWritable result = new IntWritable();

      public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException {
         int sum = 0;
         for (IntWritable val : values) {
            sum += val.get();                                    }
         result.set(sum);
         context.write(key, result);
      }
   }

   public static void main(String[] args) throws Exception {
      Configuration conf = new Configuration();
      Job job = Job.getInstance(conf, "word count");
      job.setJarByClass(WordCount.class);
      job.setMapperClass(TokenizerMapper.class);
      job.setCombinerClass(IntSumReducer.class);
      job.setReducerClass(IntSumReducer.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);
   }
}
```

2-2. ~/.barsrc 에 환경변수 설정

```
export JAVA_HOME=/usr/lib/jvm/java-7-oracle
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

2-3. WordCount 컴파일 & jar 생성

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hadoop com.sun.tools.javac.Main WordCount.java
hadoop@hadoop01:~/hadoop-2.7.3$ jar cf wc.jar WordCount*.class
hadoop@hadoop01:~/hadoop-2.7.3$ ls
bin     file02   lib          logs        sbin    WordCount.class                WordCount$TokenizerMapper.class
etc     hdfs     libexec      NOTICE.txt  share   WordCount$IntSumReducer.class
file01  include  LICENSE.txt  README.txt  wc.jar  WordCount.java
```

2-4. input 파일 생성

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -mkdir /user/hadoop/wordcount
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -mkdir /user/hadoop/wordcount/input
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -put file01 /user/hadoop/wordcount/input
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -put file02 /user/hadoop/wordcount/input
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -ls /user/hadoop/wordcount/input
Found 2 items
-rw-r--r--   4 hadoop supergroup         22 2017-05-23 00:06 /user/hadoop/wordcount/input/file01
-rw-r--r--   4 hadoop supergroup         28 2017-05-23 00:06 /user/hadoop/wordcount/input/file02
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount/input/file01
Hello World Bye World
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount/input/file02
Hello Hadoop Goodbye Hadoop
```

2-5. WordCount 실행

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hadoop jar wc.jar WordCount /user/hadoop/wordcount/input /user/hadoop/wordcount/output
```

2-6. 실행 결과

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -ls /user/hadoop/wordcount/output
Found 2 items
-rw-r--r--   4 hadoop supergroup          0 2017-05-23 00:08 /user/hadoop/wordcount/output/_SUCCESS
-rw-r--r--   4 hadoop supergroup         41 2017-05-23 00:08 /user/hadoop/wordcount/output/part-r-00000
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount/output/part-r-00000
Bye     1
Goodbye 1
Hadoop  2
Hello   2
World   2
```

3. WordCount V2

3-1. WordCount2.java 작성 (107 번 줄에서 ||스페이스 한칸 지우기)

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.net.URI;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.Counter;
import org.apache.hadoop.util.GenericOptionsParser;
import org.apache.hadoop.util.StringUtils;

public class WordCount2 {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    static enum CountersEnum { INPUT_WORDS }

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    private boolean caseSensitive;
    private Set<String> patternsToSkip = new HashSet<String>();

    private Configuration conf;
    private BufferedReader fis;

    @Override
    public void setup(Context context) throws IOException,
        InterruptedException {
      conf = context.getConfiguration();
      caseSensitive = conf.getBoolean("wordcount.case.sensitive", true);
      if (conf.getBoolean("wordcount.skip.patterns", true)) {
        URI[] patternsURIs = Job.getInstance(conf).getCacheFiles();
        for (URI patternsURI : patternsURIs) {
          Path patternsPath = new Path(patternsURI.getPath());
          String patternsFileName = patternsPath.getName().toString();
          parseSkipFile(patternsFileName);
        }
      }
    }

    private void parseSkipFile(String fileName) {
      try {
```

```
        fis = new BufferedReader(new FileReader(fileName));
        String pattern = null;
        while ((pattern = fis.readLine()) != null) {
          patternsToSkip.add(pattern);
        }
      } catch (IOException ioe) {
        System.err.println("Caught exception while parsing the cached file '"
            + StringUtils.stringifyException(ioe));
      }
    }

    @Override
    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      String line = (caseSensitive) ?
          value.toString() : value.toString().toLowerCase();
      for (String pattern : patternsToSkip) {
        line = line.replaceAll(pattern, "");
      }
      StringTokenizer itr = new StringTokenizer(line);
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
        Counter counter = context.getCounter(CountersEnum.class.getName(),
            CountersEnum.INPUT_WORDS.toString());
        counter.increment(1);
      }
    }
  }

  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                       ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    GenericOptionsParser optionParser = new GenericOptionsParser(conf, args);
    String[] remainingArgs = optionParser.getRemainingArgs();
    if (!(remainingArgs.length != 2 || remainingArgs.length != 4)) {
      System.err.println("Usage: wordcount <in> <out> [-skip skipPatternFile]");
      System.exit(2);
    }
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount2.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
```

```
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    List<String> otherArgs = new ArrayList<String>();
    for (int i=0; i < remainingArgs.length; ++i) {
      if ("-skip".equals(remainingArgs[i])) {
        job.addCacheFile(new Path(remainingArgs[++i]).toUri());
        job.getConfiguration().setBoolean("wordcount.skip.patterns", true);
      } else {
        otherArgs.add(remainingArgs[i]);
      }
    }
    FileInputFormat.addInputPath(job, new Path(otherArgs.get(0)));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs.get(1)));

    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

3-2. WordCount2 컴파일 & jar 생성

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hadoop com.sun.tools.javac.Main WordCount2.java
hadoop@hadoop01:~/hadoop-2.7.3$ jar cf wc.jar WordCount2*.class
hadoop@hadoop01:~/hadoop-2.7.3$ ls
bin           patterns.txt
etc           README.txt
file01        sbin
file02        share
hdfs          wc.jar
include       wordcount
lib           WordCount2.class
libexec       WordCount2$IntSumReducer.class
LICENSE.txt   WordCount2.java
logs          WordCount2$TokenizerMapper.class
NOTICE.txt    WordCount2$TokenizerMapper$CountersEnum.class
```

3-3. input 파일 생성

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -mkdir /user/hadoop/wordcount2
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -mkdir /user/hadoop/wordcount2/input
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -put file01 /user/hadoop/wordcount2/input
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -put file02 /user/hadoop/wordcount2/input
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount2/input/file01
Hello World, Bye World!
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount2/input/file02
Hello Hadoop, Goodbye to hadoop.
```

3-4. patterns.txt 생성 (없으면 NullPointerException 발생)

처음에는 빈 파일로 생성!

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -put patterns.txt /user/hadoop/wordcount2
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount2/patterns.txt
hadoop@hadoop01:~/hadoop-2.7.3$
```

3-5. WordCount2 실행

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hadoop jar wc.jar WordCount2 -Dwordcount.case.sensitive=true /user/hadoop/wordcount2/input /user/hadoop/wordcount2/output -skip /user/hadoop/wordcount2/patterns.txt
```

3-6. 실행 결과

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -ls /user/hadoop/wordcount2/output
Found 2 items
-rw-r--r--   4 hadoop supergroup          0 2017-05-23 14:21 /user/hadoop/wordcount2/output/_SUCCESS
-rw-r--r--   4 hadoop supergroup         67 2017-05-23 14:21 /user/hadoop/wordcount2/output/part-r-00000
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount2/output/part-r-00000
Bye     1
Goodbye 1
Hadoop, 1
Hello   2
World!  1
World,  1
hadoop. 1
to      1
```

3-7. patterns.txt 에 내용 추가

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount2/patterns.txt
\.
\,
\!
to
```

3-8. WordCount2 실행

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hadoop jar wc.jar WordCount2 -Dwordcount.case.sensitive=true /user/hadoop/wordcount2/input /user/hadoop/wo
rdcount2/output -skip /user/hadoop/wordcount2/patterns.txt
```

3-9. 실행 결과

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -ls /user/hadoop/wordcount2/output
Found 2 items
-rw-r--r--   4 hadoop supergroup          0 2017-05-23 14:27 /user/hadoop/wordcount2/output/_SUCCESS
-rw-r--r--   4 hadoop supergroup         50 2017-05-23 14:27 /user/hadoop/wordcount2/output/part-r-00000
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount2/output/part-r-00000
Bye     1
Goodbye 1
Hadoop  1
Hello   2
World   2
hadoop  1
```

3-10. -Dwordcount.case.sensitive=false 로 WordCount2 실행

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hadoop jar wc.jar WordCount2 -Dwordcount.case.sensitive=false /user/hadoop/wordcount2/input /user/hadoop/w
ordcount2/output -skip /user/hadoop/wordcount2/patterns.txt
```

3-11. 실행 결과

```
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -ls /user/hadoop/wordcount2/output
Found 2 items
-rw-r--r--   4 hadoop supergroup          0 2017-05-23 14:30 /user/hadoop/wordcount2/output/_SUCCESS
-rw-r--r--   4 hadoop supergroup         41 2017-05-23 14:30 /user/hadoop/wordcount2/output/part-r-00000
hadoop@hadoop01:~/hadoop-2.7.3$ bin/hdfs dfs -cat /user/hadoop/wordcount2/output/part-r-00000
bye     1
goodbye 1
hadoop  2
hello   2
world   2
```