

Tracy Michaels Assignment 1

1) Setting up Hadoop.

- Install java JDK and JRE
- Download Hadoop .tar file
- Uncompress file
- Update Hadoop-env.sh file to include 'export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/'
- Create tmp folder in hadoop
- Update core-site.xml in etc directory to include necessary properties
- Do the same for hdfs-site.xml
- Update .bashrc for Hadoop to append following lines to file
 - o Export HADOOP_HOME=/home/rob/Hadoop
 - o Export PATH=\$HADOOP_HOME/bin:\$PATH

(I do not have screen shots of this part as I had set it up before this assignment was assigned)

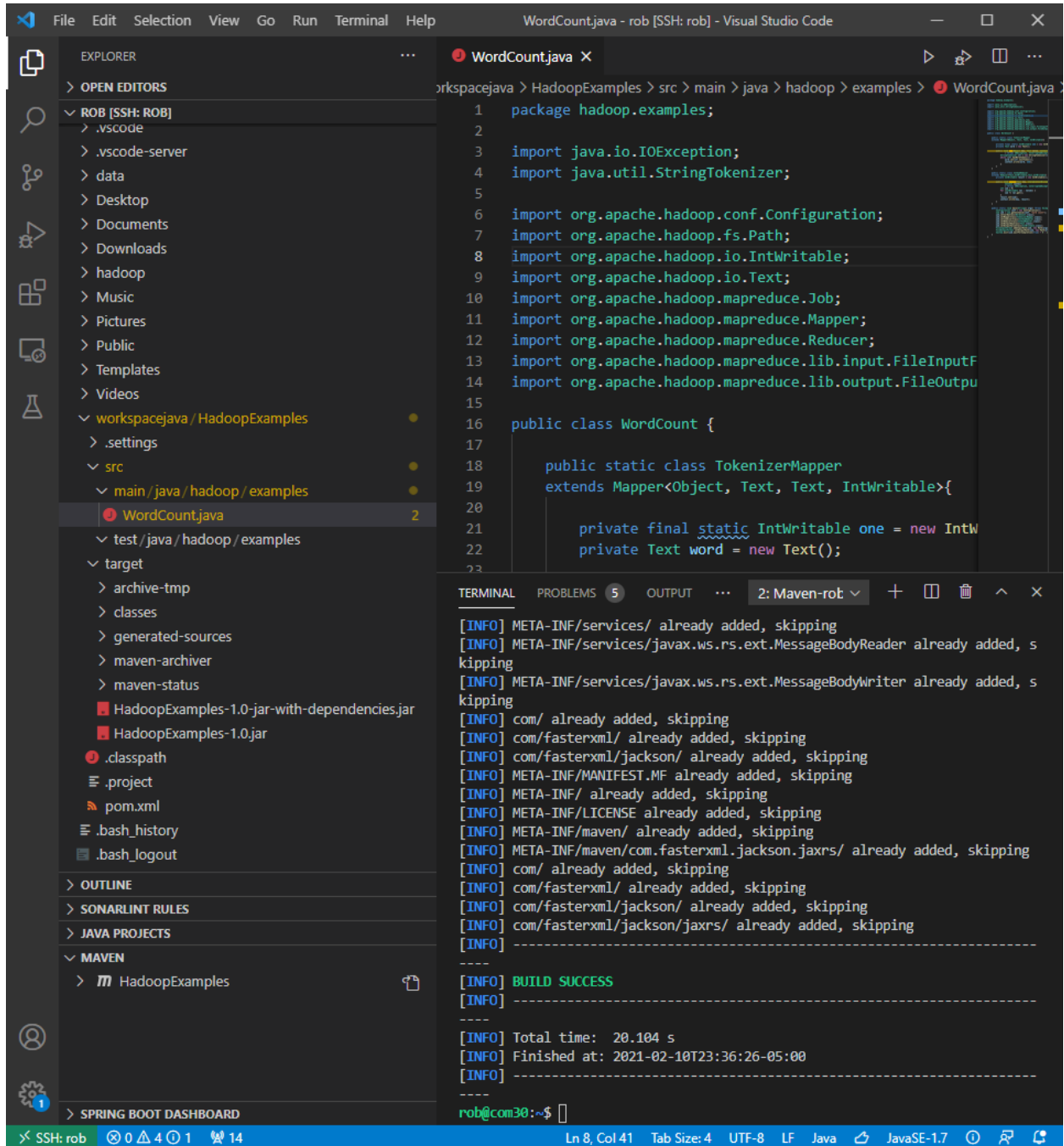
- 2) Setup HDFS and upload datasets 'test.txt' and 'peterpan.txt' into HDFS
- Install openssh-client and server
 - Create an rsa encrypted public key and store in 'authorized_keys' file
 - This allows ssh to local host without need for a password
 - Use command 'hdfs namenode -format' to prepare hdfs
 - Start hdfs with command 'sbin/start-dfs.sh' from inside the Hadoop folder
 - Use command 'hdfs dfs -mkdir /path' to create necessary folders
 - Use command 'hdfs dfs -put [local source] [hdfs destination]' to copy files from local directory to hdfs

The screenshot shows the Visual Studio Code interface with a terminal window open. The Explorer panel on the left shows the file structure of the project, including a 'data' folder containing 'peterpan.txt' and 'test.txt'. The terminal window displays the following commands and output:

```
rob@com30:~/hadoop$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [com30]
rob@com30:~/hadoop$ hdfs dfs -ls
Found 1 items
drwxr-xr-x  - rob supergroup          0 2021-02-02 11:21 data
rob@com30:~/hadoop$ hdfs dfs -ls /user
Found 1 items
drwxr-xr-x  - rob supergroup          0 2021-02-02 11:21 /user/rob
rob@com30:~/hadoop$ hdfs dfs -ls /user/rob
Found 1 items
drwxr-xr-x  - rob supergroup          0 2021-02-02 11:21 /user/rob/data
rob@com30:~/hadoop$ hdfs dfs -ls /user/rob/data
rob@com30:~/hadoop$ hdfs dfs -put /home/rob/data/peterpan.txt /user/rob/data
rob@com30:~/hadoop$ hdfs dfs -put /home/rob/data/test.txt /user/rob/data
rob@com30:~/hadoop$ hdfs dfs -ls /user/rob/data
Found 2 items
-rw-r--r--  1 rob supergroup    291927 2021-02-10 22:09 /user/rob/data/peterpan.txt
-rw-r--r--  1 rob supergroup      66 2021-02-10 22:09 /user/rob/data/test.txt
rob@com30:~/hadoop$
```

- 3) Create a maven project in VS Code, import the WordCount.java file

 - Add dependencies and plug-ins to pom.xml file
 - Create .jar file using maven



4) Open a terminal, and run the 'WordCount.jar' file on "test.txt" and "peterpan.txt"

On test.txt:

`hadoop jar /home/rob/workspacejava/HadoopExamples/target/HadoopExamples-1.0-jar-with-dependencies.jar file:///home/rob/data/test.txt file:///home/rob/test_output`

```
test_output > part-r-00000
1  hadoop  4
2  hbase   1
3  hive    1
4  pig     2
5  spark   3
6
```

```
Map output bytes=109
Map output materialized bytes=64
Input split bytes=93
Combine input records=11
Combine output records=5
Reduce input groups=5
Reduce shuffle bytes=64
Reduce input records=5
Reduce output records=5
Spilled Records=10
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=6
Total committed heap usage (bytes)=360710144

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=66
File Output Format Counters
Bytes Written=50

rob@com30:~$
```

On peterpan.txt:

hadoop jar /home/rob/workspacejava/HadoopExamples/target/HadoopExamples-1.0-jar-with-dependencies.jar file:///home/rob/data/peterpan.txt file:///home/rob/peterpan_output

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Displays the file system structure. The `peterpan_output` directory is expanded, showing files like `_SUCCESS`, `_SUCCESS.crc`, `.part-r-00000.crc`, and `part-r-00000`. The `part-r-00000` file is selected.
- WordCount.java:** The main editor shows the content of the selected file, which is a list of word counts. The content is as follows:

```
1 #10000, 2
2 #16] 1
3 $5,000) 1
4 ($1 1
5 (2) 1
6 (3) 1
7 (4) 1
8 (801) 1
9 (I 1
10 (Morgan's 1
11 (Or 1
12 (Slightly 1
13 (a) 1
14 (and 2
15 (any 1
16 (as 1
17 (available 1
18 (b) 1
19 (but 2
20 (c) 1
21 (c)1991 1
22 (does 1
23 (he 1
24 (if 1
25 (or 3
26 (the 1
27 (though 1
28 (trademark/copyright) 1
29 (which 1
30 (winking 1
31 (www.gutenberg.org), 1
32 (zipped), 1
33 ("the 1
34 ** 4
35 *** 8
36 ***** 2
37 - 7
38 /etext 1
39 00 1
```
- TERMINAL:** The terminal output shows the results of the Hadoop job:

```
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=291927
File Output Format Counters
Bytes Written=96332
rob@com30:~$
```