

HW 02

```
jinyang — azure@sandbox:~ — ssh azureSandbox — 80x24
Last login: Tue Sep 12 14:08:35 on ttys001
[jinyangdeMacBook-Pro:~ jinyang$ ssh azureSandbox]
The authenticity of host '52.183.36.160 (52.183.36.160)' can't be established.
ECDSA key fingerprint is SHA256:mcZfmRdFK67LSKTEem1P3GAY5SRqDWUAGIbRcqUfq7k.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.183.36.160' (ECDSA) to the list of known hosts.
[Password:]
[Password:]
Last failed login: Tue Sep 12 21:24:11 UTC 2017 from 207.237.205.155 on ssh:notty
There was 1 failed login attempt since the last successful login.
La

jinyang — maria_dev@sandbox:~ — ssh -p 2222 maria_dev@localhost — 102x25
[azure@sandbox ~]$
[azure@sandbox ~]$
n pipe
[jinyangdeMacBook-Pro:~ jinyang$ ssh -p 2222 maria_dev@localhost]
[Password:]
[Password:]
Last failed login: Tue Sep 12 21:24:11 UTC 2017 from 104.194.99.180 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Tue Sep 12 21:24:11 UTC 2017 from 207.237.205.155 on ssh:notty
[azure@sandbox ~]$
Merged Map outputs=2
Physical memory (bytes) snapshot=539078656
Reduce input groups=2
Reduce input records=4
Reduce output records=2
Reduce shuffle bytes=67
Shuffled Maps =2
Spilled Records=8
Total committed heap usage (bytes)=278921216
Virtual memory (bytes) snapshot=5824172032
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
Permission WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
100% 449 6.4KB/s 00:00
Streaming final output from hdfs:///user/maria_dev/tmp/mrjob/WordCount2.maria_dev.20170912.233229.9809
96/output:..maria_dev@localhost:/home/maria_dev
"a_to_n"Count2.46
"others"Count2.46
100% 470 7.4KB/s 00:00
Removing HDFS temp directory hdfs:///user/maria_dev/tmp/mrjob/WordCount2.maria_dev.20170912.233229.980
96...
Removing temp directory /tmp/WordCount2.maria_dev.20170912.233229.98096...
100% 472 7.3KB/s 00:00
[maria_dev@sandbox ~]$
```

1)

WordCount2.py:

```
from mrjob.job import MRJob
import re

WORD_RE = re.compile(r"[\w']+")

class MRWordCount(MRJob):

    def mapper(self, _, line):
        for word in WORD_RE.findall(line):
```

```

        if
word.lower().startswith( ("a","b","c","d","e","f","g","h","i","j","k",
"l","m","n") ):
            yield "a_to_n", 1
        else:
            yield "others", 1

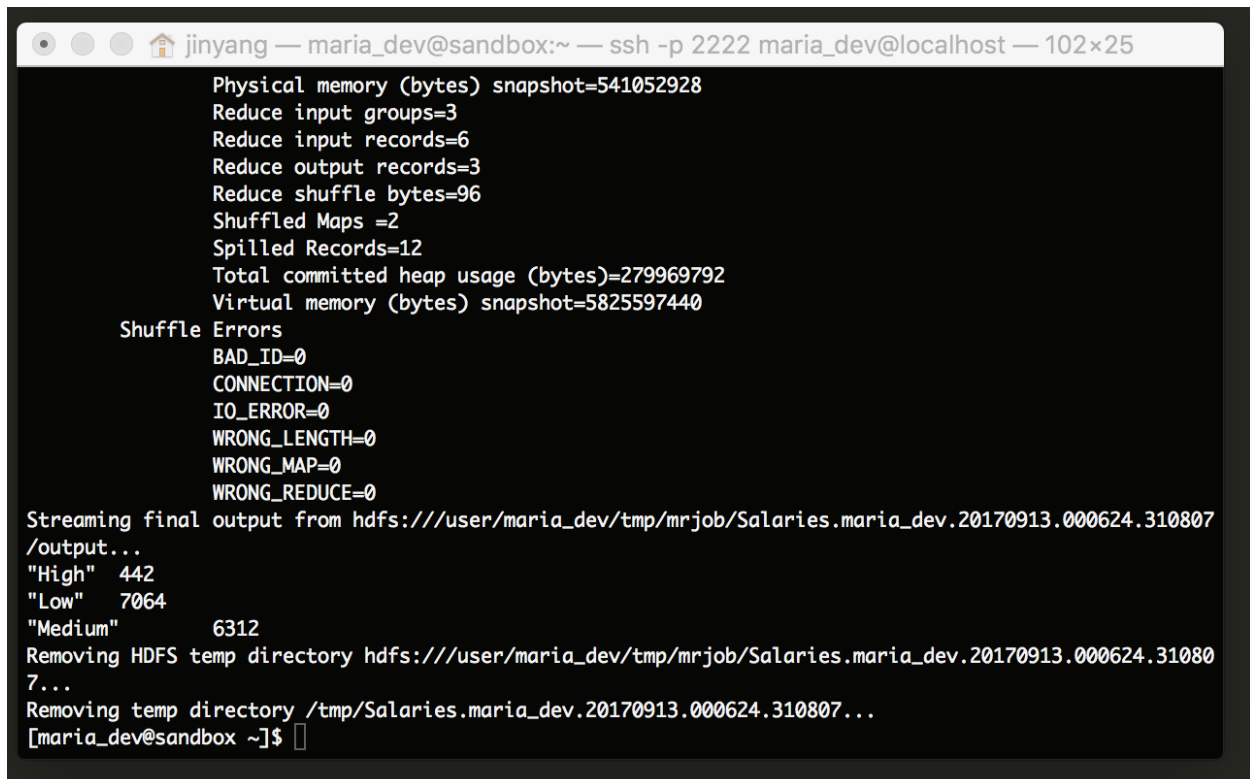
    def combiner(self, word, counts):
        yield word, sum(counts)

    def reducer(self, word, counts):
        yield word, sum(counts)

if __name__ == '__main__':
    MRWordCount.run()

```

2)



The screenshot shows a terminal window with a title bar indicating the user is 'jinyang' on a 'maria_dev@sandbox' machine, connected via SSH to 'maria_dev@localhost'. The terminal displays the output of a Hadoop job, including resource usage statistics and shuffle errors. The output shows that the job completed successfully with no errors. The final output is streamed from HDFS to the local file system, showing counts for 'High' (442), 'Low' (7064), and 'Medium' (6312) categories. The terminal also shows the removal of temporary HDFS and local directories used during the job execution.

```

Physical memory (bytes) snapshot=541052928
Reduce input groups=3
Reduce input records=6
Reduce output records=3
Reduce shuffle bytes=96
Shuffled Maps =2
Spilled Records=12
Total committed heap usage (bytes)=279969792
Virtual memory (bytes) snapshot=5825597440
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
Streaming final output from hdfs:///user/maria_dev/tmp/mrjob/Salaries.maria_dev.20170913.000624.310807
/output...
"High" 442
"Low" 7064
"Medium" 6312
Removing HDFS temp directory hdfs:///user/maria_dev/tmp/mrjob/Salaries.maria_dev.20170913.000624.310807...
Removing temp directory /tmp/Salaries.maria_dev.20170913.000624.310807...
[maria_dev@sandbox ~]$

```

```
from mrjob.job import MRJob

class MRSalaries(MRJob):

    def mapper(self, _, line):
        (name,jobTitle,agencyID,agency,hireDate,annualSalary,grossPay) = line.split('\t')
        if float(annualSalary) >= 100000.00:
            yield "High", 1
        elif 0.00 <= float(annualSalary) <= 49999.00:
            yield "Low", 1
        else:
            yield "Medium", 1

    def combiner(self, jobTitle, counts):
        yield jobTitle, sum(counts)

    def reducer(self, jobTitle, counts):
        yield jobTitle, sum(counts)

if __name__ == '__main__':
    MRSalaries.run()
```