To start Hadoop Commands:

1. su - hadoop
2. sudo service ssh start
3. ssh localhost
4. start-all.sh
5. hdfs dfs -mkdir word_count_ex #created directory in hadoop cluster

Link: https://www.geeksforgeeks.org/hadoop-streaming-using-python-word-count-problem/

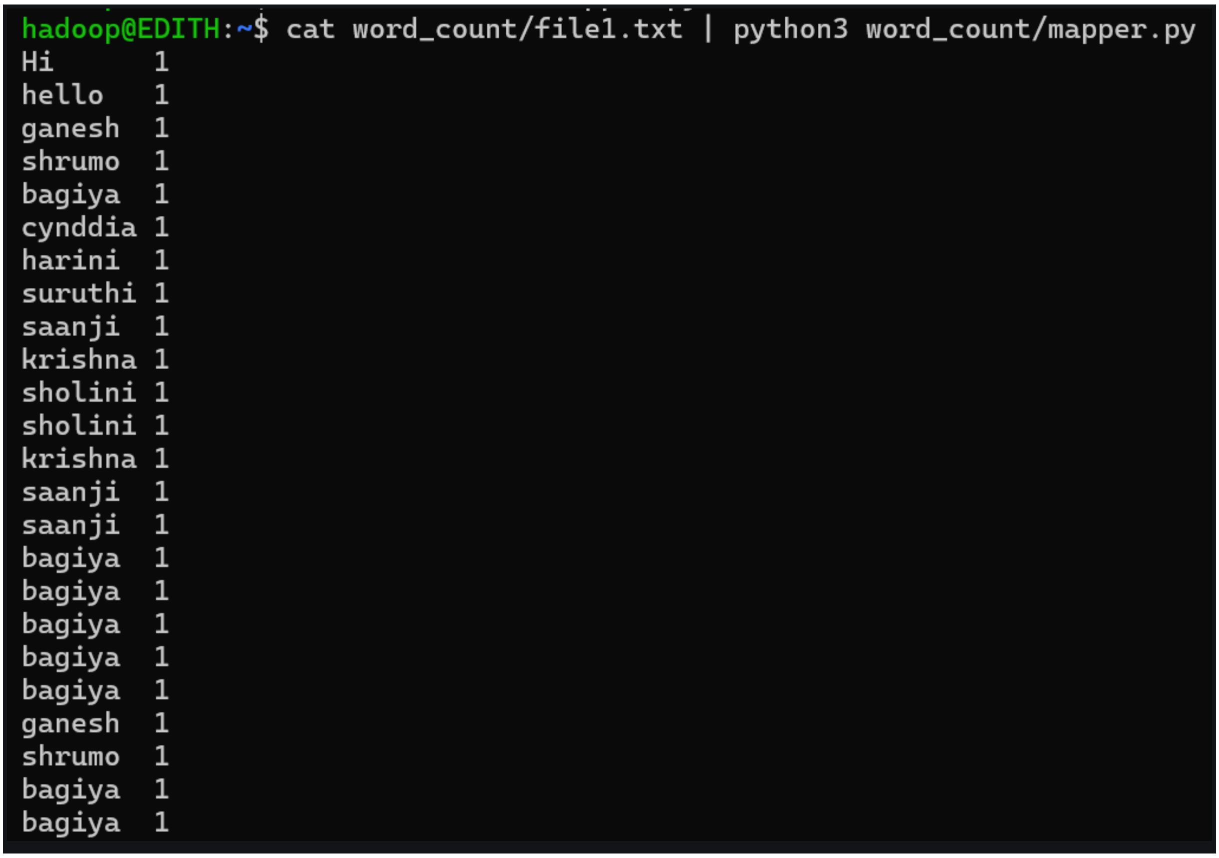
Step 1: Create a file with the name *file1.txt* and add some data to it.

This is for input. Created file1.txt inside word_count directory locally.

Step 2: Create a **mapper.py** file that implements the mapper logic. It will read the data from STDIN and will split the lines into words, and will generate an output of each word with its individual count.

nano word_count/mapper.py

Let's test our **mapper.py** locally that it is working fine or not.



```
hadoop@EDITH:~$ cat word_count/file1.txt | python3 word_count/mapper.py
Hi      1
hello   1
ganesh  1
shrumo  1
bagiya  1
cynddia 1
harini  1
suruthi 1
saanji  1
krishna 1
sholini 1
sholini 1
krishna 1
saanji  1
saanji  1
bagiya  1
bagiya  1
bagiya  1
bagiya  1
bagiya  1
ganesh  1
shrumo  1
bagiya  1
bagiya  1
```

Step 3: Create a ***reducer.py*** file that implements the reducer logic. It will read the output of mapper.py from STDIN(standard input) and will aggregate the occurrence of each word and will write the final output to STDOUT.

We can see that our reducer is also working fine in our local system.

Step 4: Now let’s start all our Hadoop daemons with the below command.

Input to the hadoop cluster:
hdfs dfs -copyFromLocal /home/hadoop/word_count/file1.txt /word_count_ex

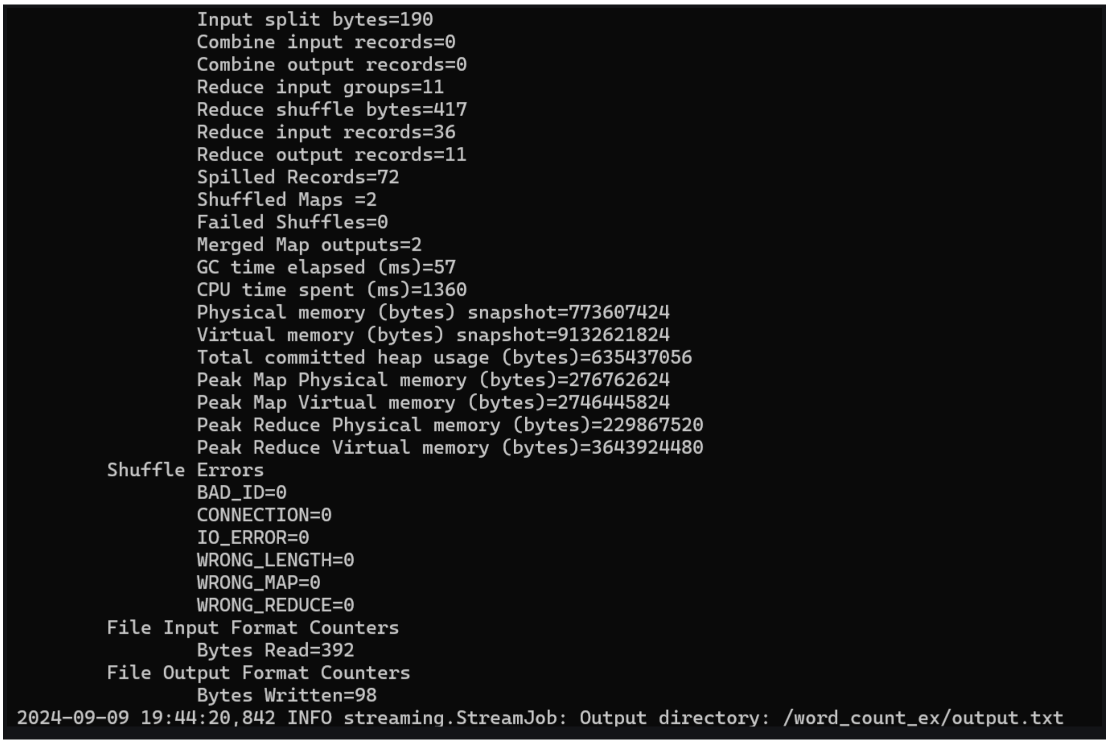
Let’s give executable permission to our **mapper.py** and **reducer.py** with the help of below command.

```
chmod 777 mapper.py reducer.py
```

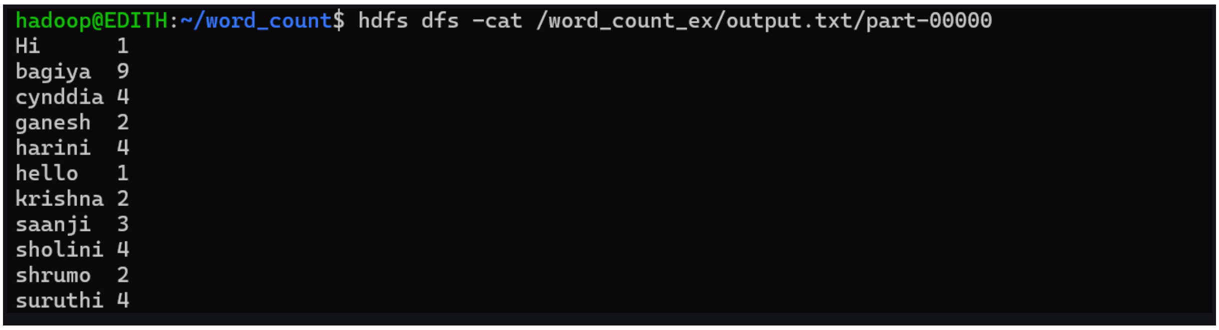
Step 5: Now download the latest **hadoop-streaming jar** file from this Link**.** Then place, this Hadoop,-streaming jar file to a place from you can easily access it. In my case, I am placing it to ***/Documents*** folder where **mapper.py** and **reducer.py** file is present.

```
 hadoop jar /home/shruthimohan/hadoop-streaming-3.3.6.jar -input
/cia1/input/ncdc_dataset.csv -output /user/hadoop/cia1/output -mapper
/home/hadoop/cia1/mapper.py -reducer /home/hadoop/cia1/reducer.py
```

Worked!



```
                Input split bytes=190
                Combine input records=0
                Combine output records=0
                Reduce input groups=11
                Reduce shuffle bytes=417
                Reduce input records=36
                Reduce output records=11
                Spilled Records=72
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=57
                CPU time spent (ms)=1360
                Physical memory (bytes) snapshot=773607424
                Virtual memory (bytes) snapshot=9132621824
                Total committed heap usage (bytes)=635437056
                Peak Map Physical memory (bytes)=276762624
                Peak Map Virtual memory (bytes)=2746445824
                Peak Reduce Physical memory (bytes)=229867520
                Peak Reduce Virtual memory (bytes)=3643924480
        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=392
        File Output Format Counters
                Bytes Written=98
 2024-09-09 19:44:20,842 INFO streaming.StreamJob: Output directory: /word_count_ex/output.txt
```



```
hadoop@EDITH:~/word_count$ hdfs dfs -cat /word_count_ex/output.txt/part-00000
Hi      1
bagiya  9
cynddia 4
ganesh  2
harini  4
hello   1
krishna 2
saanji  3
sholini 4
shrumo  2
suruthi 4
```

Part-00000: first reducer’s output