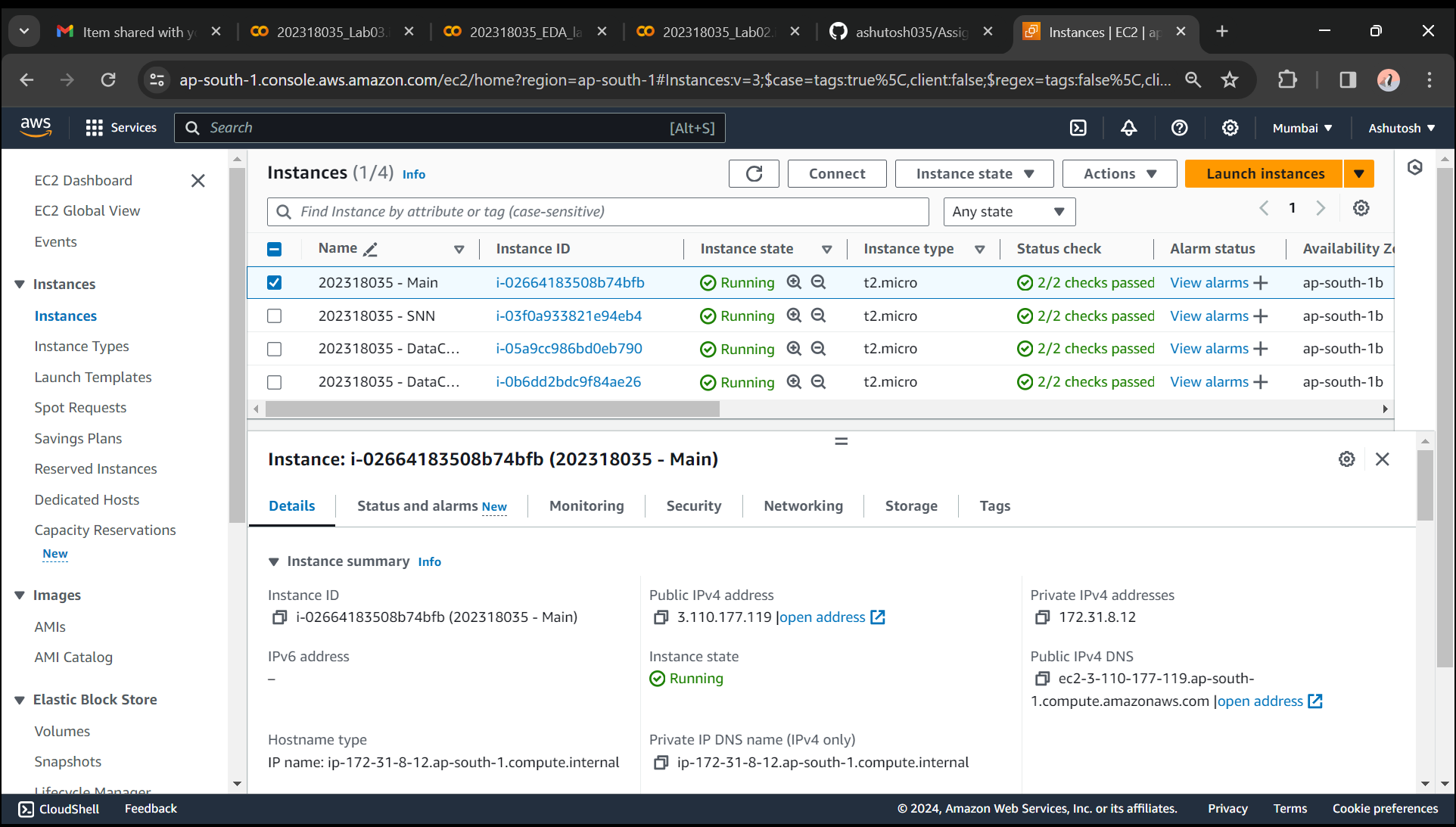
**IT 609 - Big Data Processing Lab 3**

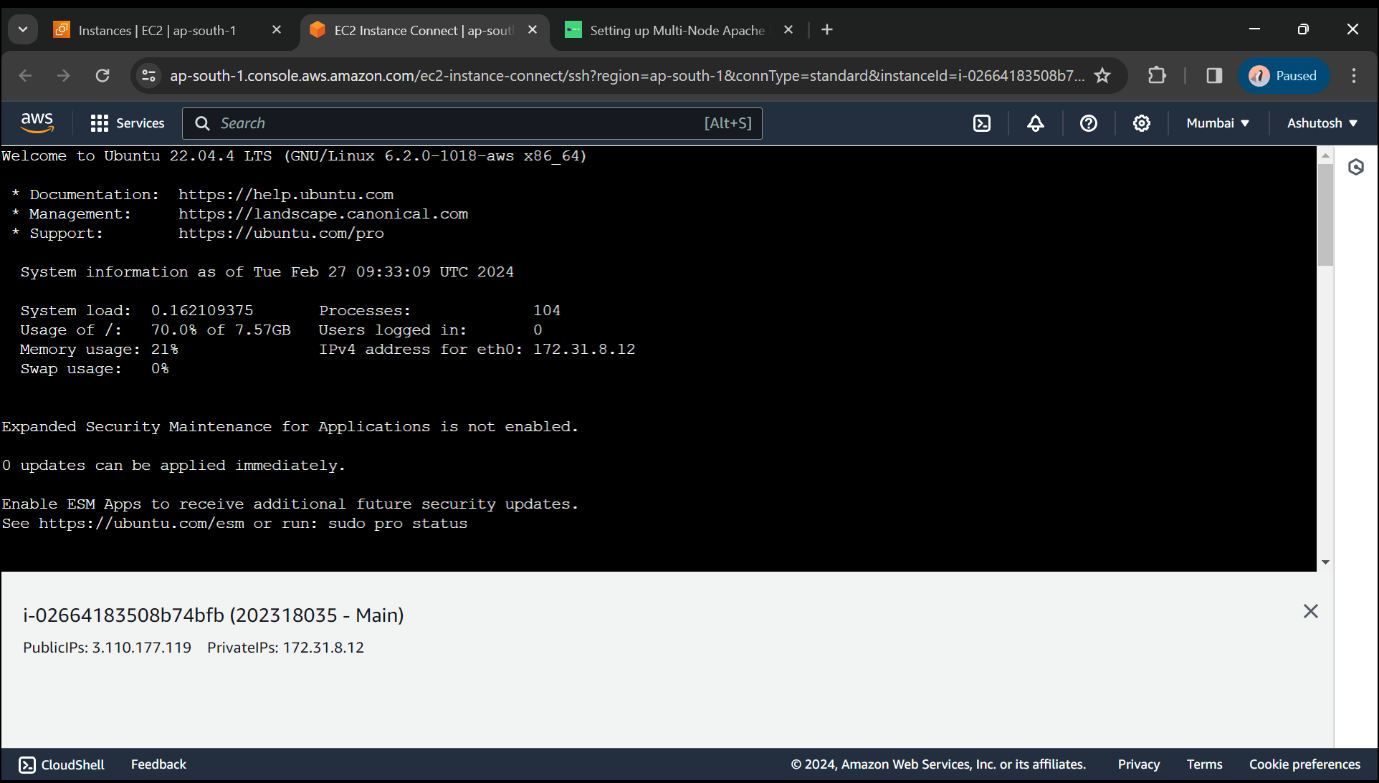
**Ashutosh Anand**

**202318035**

Implement the Map-Reduce algorithm in single Node cluster and Multi Node cluster using Hadoop-streaming Utility.

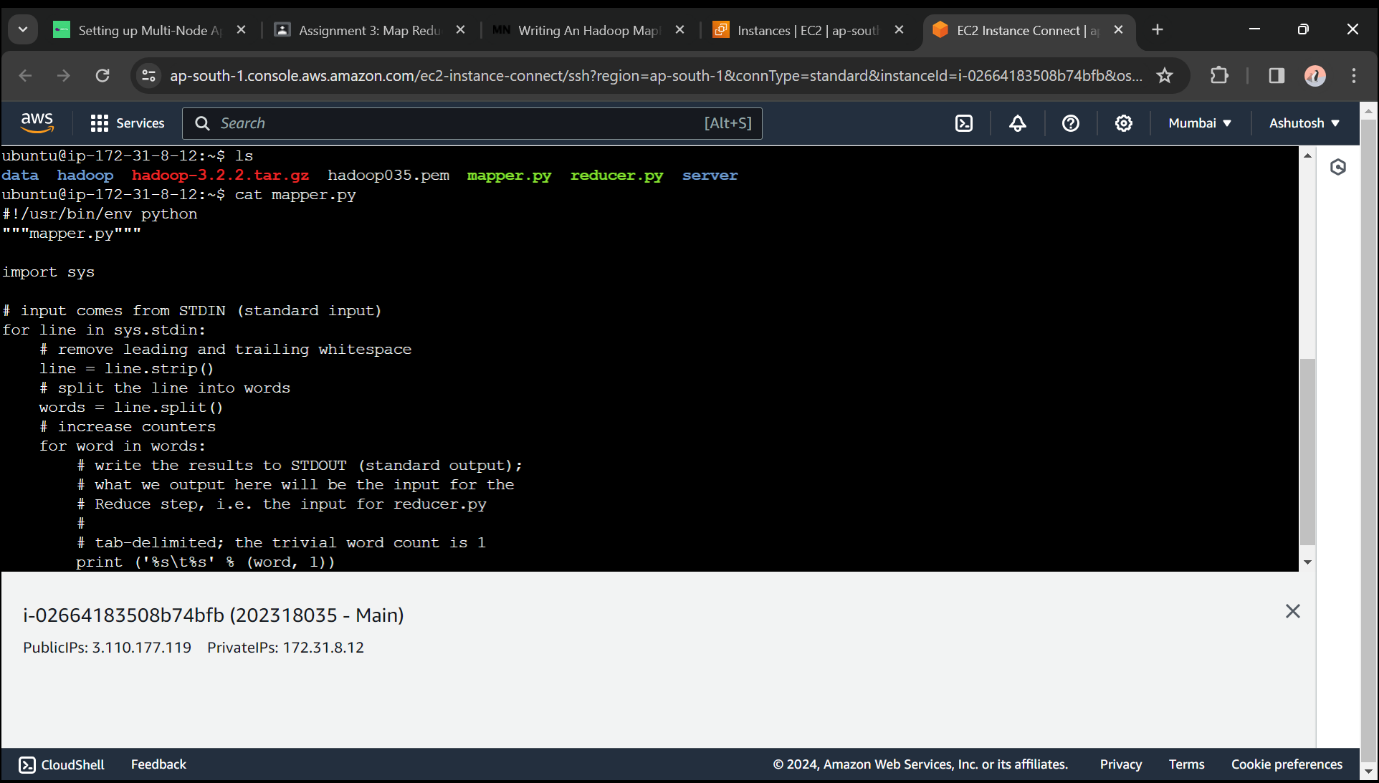
The Instances Created on the EC2, containing the Name, Secondary Name Node, Data Cluster1 & 2.



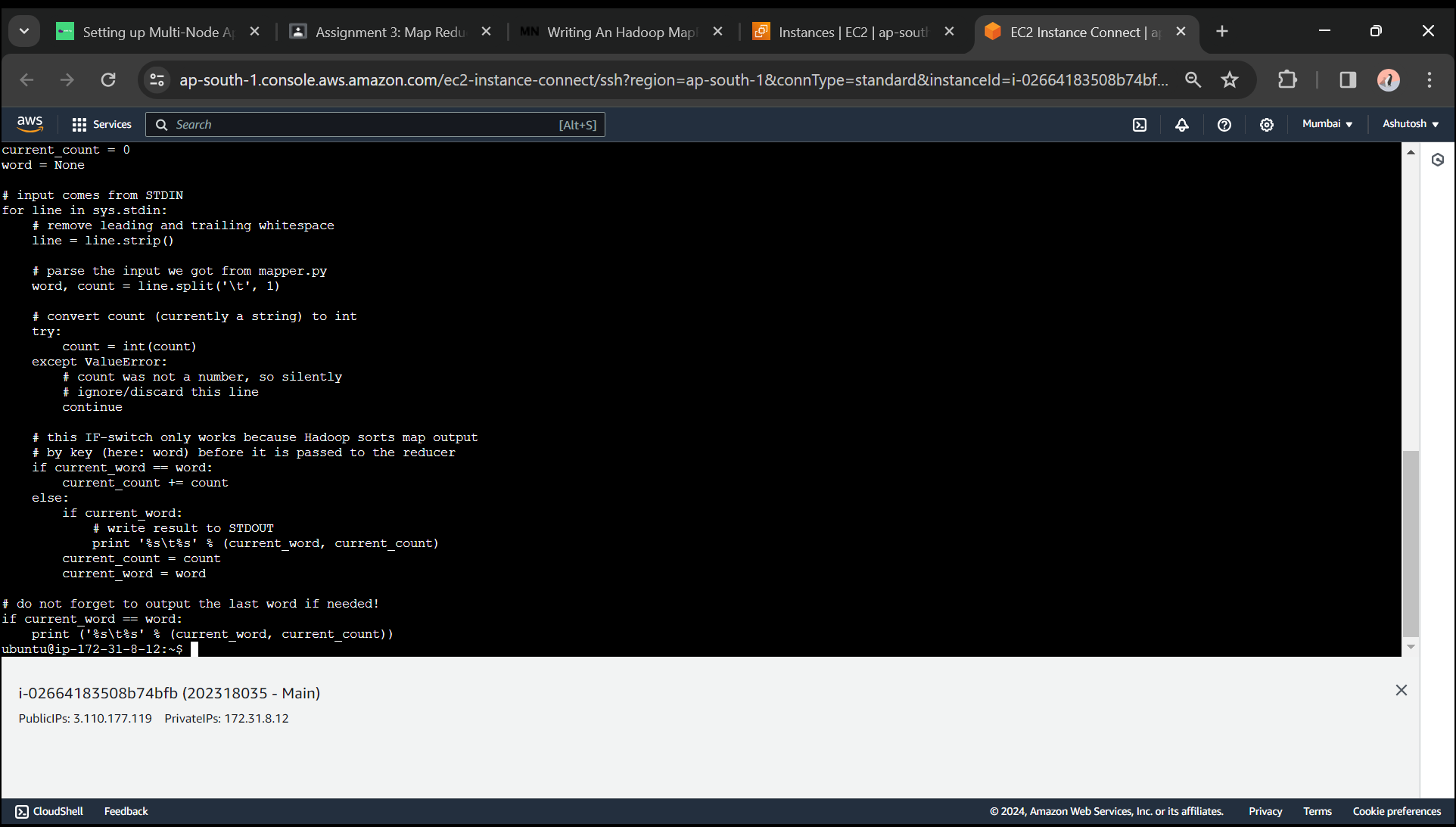
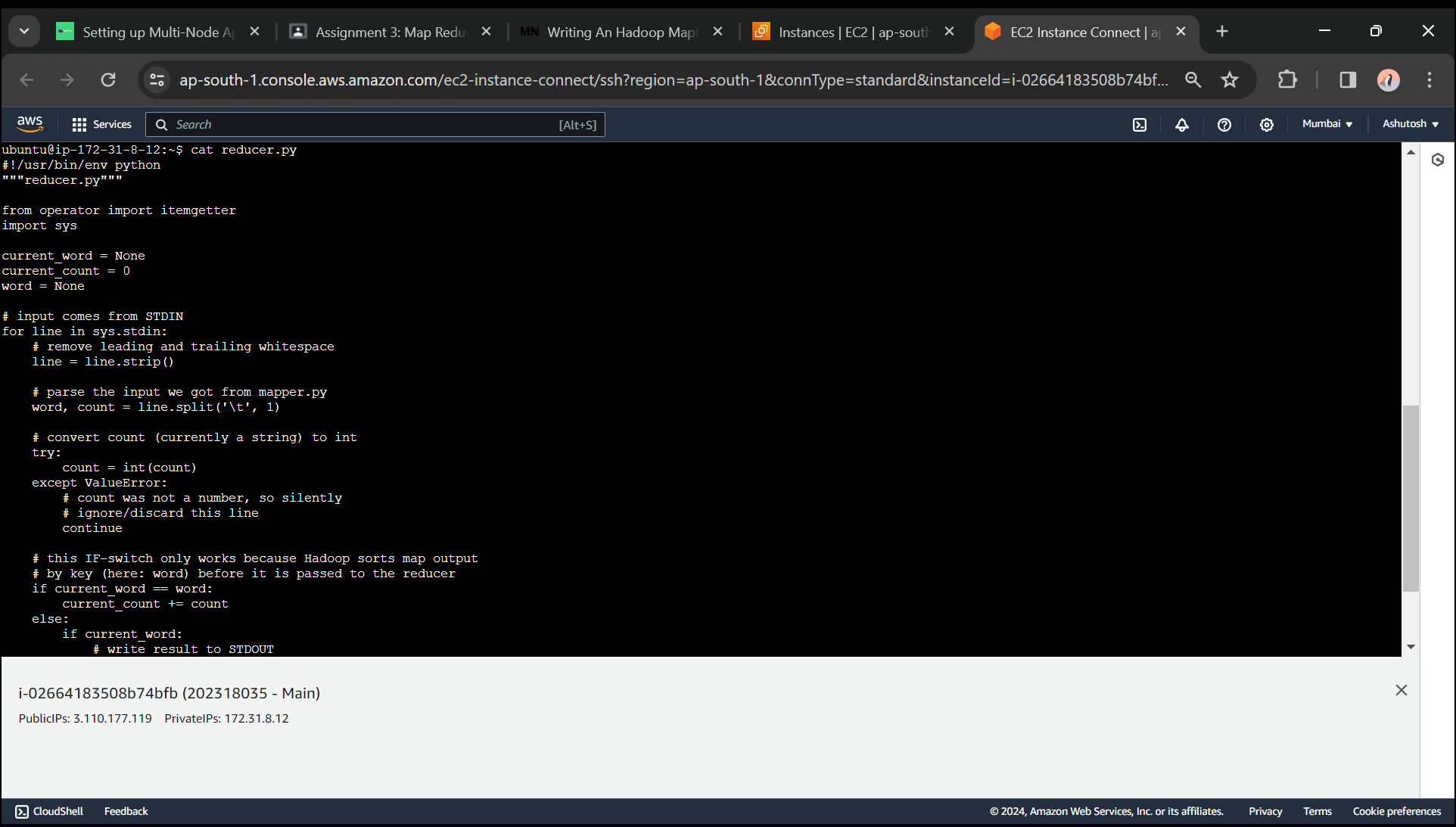
Step: Accessing the Main Node

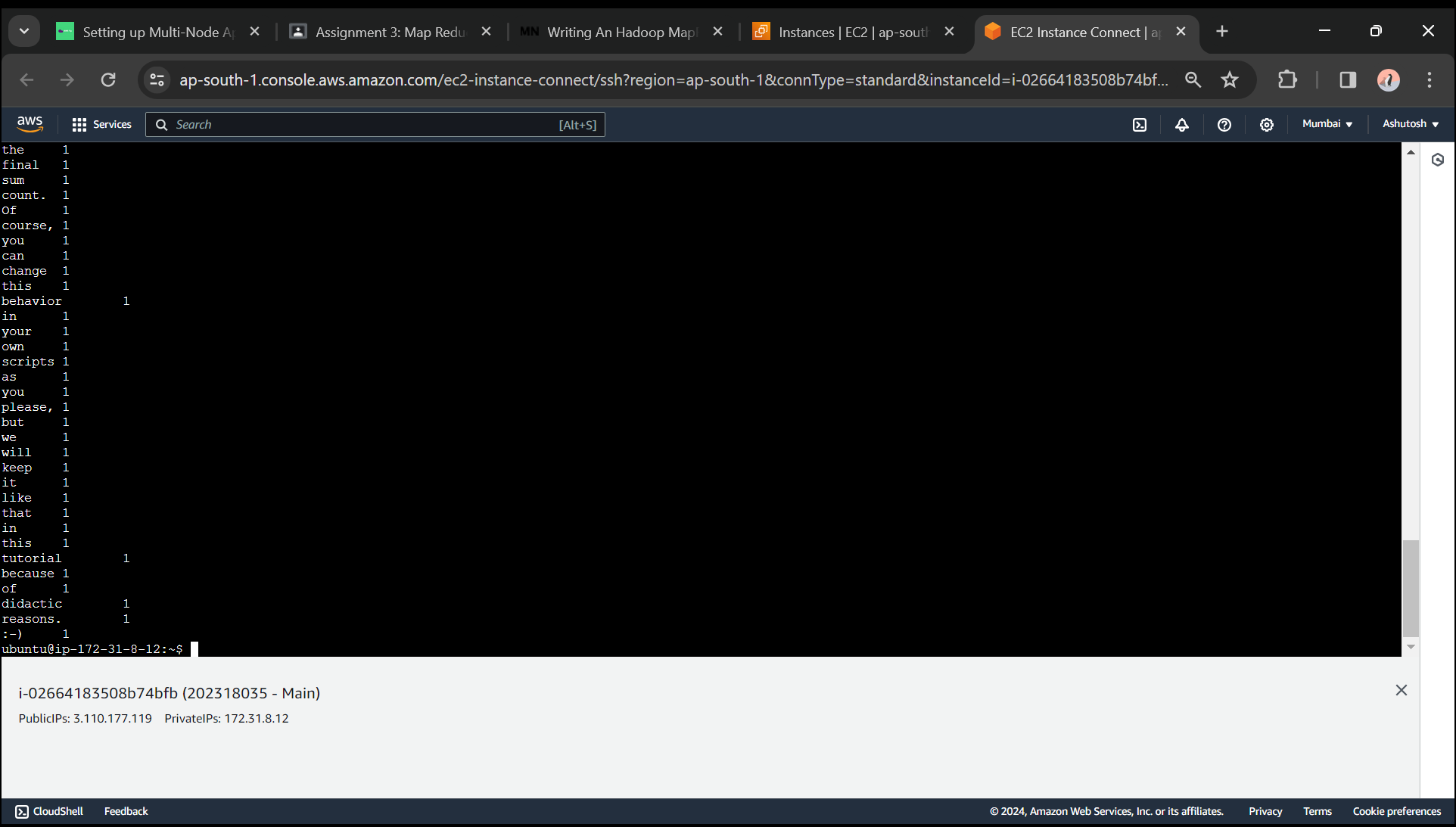
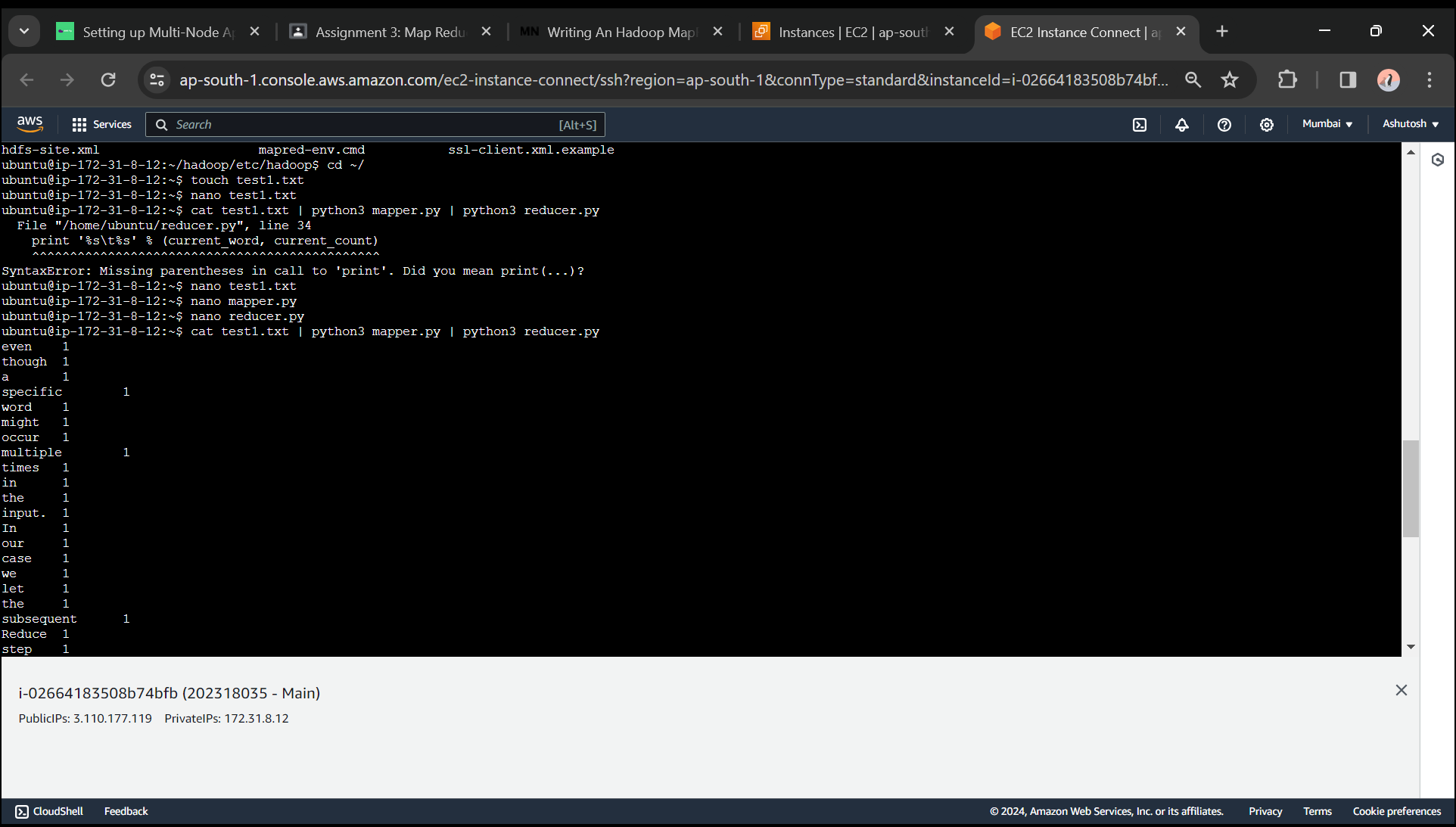
Step: Map Step: mapper.py

It will read data, split it into words and output a list of lines mapping words to their (intermediate) counts to. The Map script will not compute an (intermediate) sum of a word’s occurrences though. Instead, it will output <word> 1 tuples immediately – even though a specific word might occur multiple times in the input.



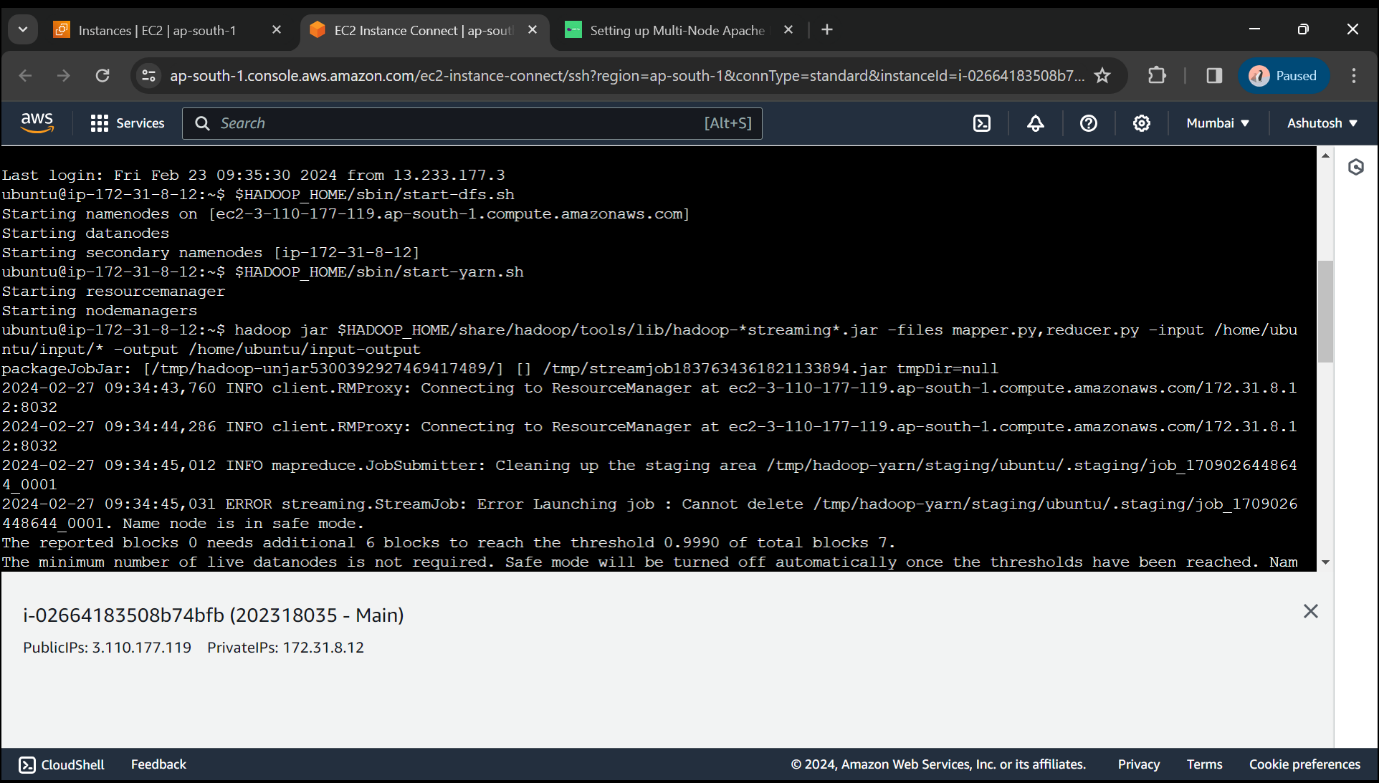
Step: Reduce Step: reducer.py

It will read the results of mapper.py  and sum the occurrences of each word to a final count, and then output its results.

Step: Checking the mapper.py and reducer.py file with a short txt file to check if the code is working properly or not.

Step: Here we are starting the dfs.sh and yarn.sh and leaving the safe mode to run the MapReduce Job.

We leverage the Hadoop Streaming API for helping us passing data between our Map and Reduce code.



The job will read all the files in the HDFS directory /home/ubuntu/input/\* , process it, and store the results in the HDFS directory /home/ubuntu/input-output . In general Hadoop will create one output file per reducer.

As per the Screenshot the MapReduce Job is running.

