**Assignment 3: Hadoop Streaming and MRjob**

1. Use **both** Hadoop streaming jar at the **University Hadoop system** and streaming step at **AWS** EMR to execute a word count mapreduce job using “NEWS.txt”. Before running the mapreduce job on Hadoop and EMR, please test both mapper and reducer python files (mapper.py and reducer.py) locally using the random String "peak a boo boo peek peek peak boo a" and the sample file “NEWS.txt”. Write down **all your steps** and paste the corresponding **screenshots**.

Change the execution permission of the python files:

chmod +x mapper.py

chmod +x reducer.py

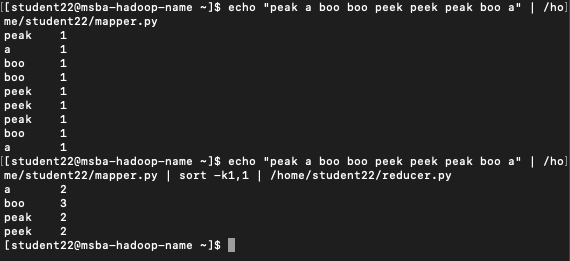


Test the two python files locally before running them using Hadoop:

echo "peak a boo boo peek peek peak boo a" | /home/student22/mapper.py

echo "peak a boo boo peek peek peak boo a" | /home/student22/mapper.py | sort -k1,1 | /home/student22/reducer.py

cat NEWS.txt | /home/student22/mapper.py



Text

Description automatically generated

Copy the sample data to HDFS and Execute the mapper and reducer using hadoop streaming:

hdfs dfs -copyFromLocal NEWS.txt /home/22student22/input3



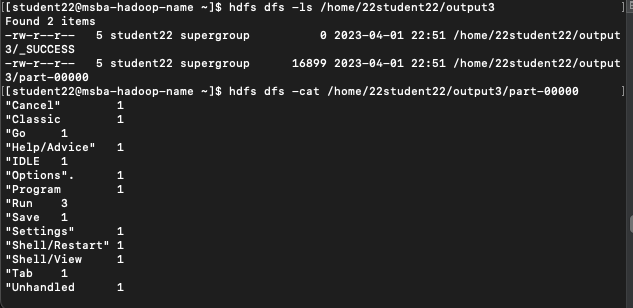
hadoop jar hadoop-streaming-2.7.3.jar -file /home/student22/mapper.py -mapper /home/student22/mapper.py -file /home/student22/reducer.py -reducer /home/student22/reducer.py -input /home/22student22/input3 -output /home/22student22/output3

Text

Description automatically generated

Find the print out the final output:

hdfs dfs -cat /home/22student22/output3/part-00000



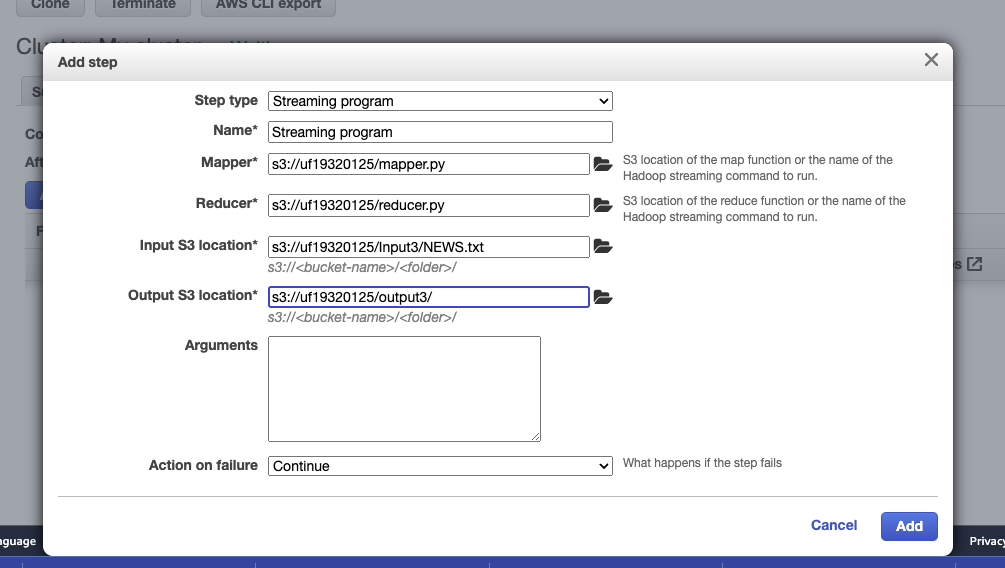
**AWS**

S3 objects : creating input folder, upload mapper.py file and reducer.py file

Graphical user interface, text, application, email

Description automatically generated

Execute the map and reduce python scripts at AWS EMR



Output directory

Graphical user interface, text, application, email

Description automatically generated

Output file content

Text

Description automatically generated

2. Create and execute a MaxTemp python file at the University Hadoop system both **locally** and **on the Hadoop cluster** using mrjob library with “sample.txt”. Write the outputs to output files. Write down **all your steps** and paste the corresponding **screenshots**.

Creating and executing the MaxTemp python file at the University Hadoop system locally using mrjob library with “sample.txt”:

Creating file: Text

Description automatically generated

Executing:

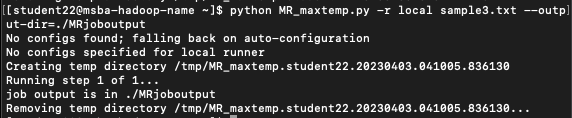
python MR\_maxtemp.py -r local sample3.txt

Text

Description automatically generated

Writing output to an output file:

python MR\_maxtemp.py -r local sample3.txt --output-dir=./MRjoboutput



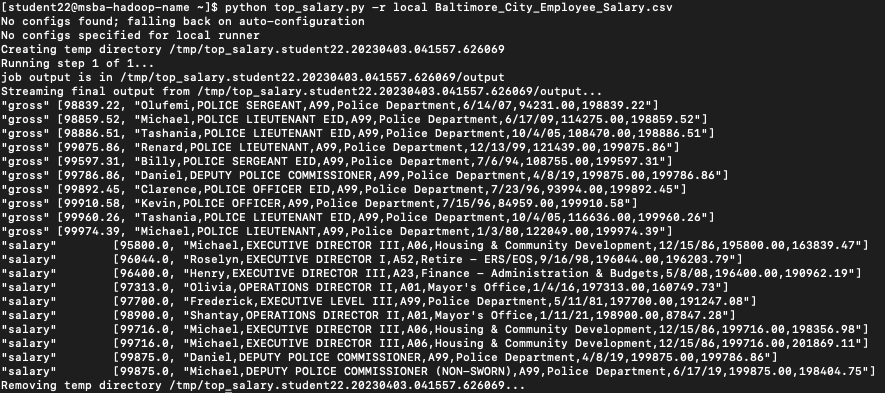
Graphical user interface, application, Teams

Description automatically generated

3. Execute the TopSalary python file at the University Hadoop system both **locally** and **on the Hadoop cluster** using mrjob library with “Baltimore\_City\_Employee\_Salary.csv”. Write the outputs to output files. Write down **all your steps** and paste the corresponding **screenshots**.

Executing the TopSalary python file at the University Hadoop system locally using mrjob library with “Baltimore\_City\_Employee\_Salary.csv”:

python top\_salary.py -r local Baltimore\_City\_Employee\_Salary.csv



Writing output to an outpu file:

python top\_salary.py -r local Baltimore\_City\_Employee\_Salary.csv --output-dir=./salaries\_output

Text

Description automatically generated

