GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 4150-A Spring 2021
Lab: Batch Data Analysis using Hadoop, MapReduce, Pig & Hive

References:

[1] A. Bahga, V. Madisetti, ÒCloud Computing Solutions Architect: A Hands-On ApproachÓ, ISBN: 978-0996025591
[2] https://pythonhosted.org/mrjob/
[3] http://hadoop.apache.org/
[4] http://storage.googleapis.com/books/ngrams/books/datasetsv2.html
[5] http://pig.apache.org/docs/r0.15.0/basic.html
[6] https://cwiki.apache.org/confluence/display/Hive/LanguageManual

Due Date:

The lab report will be **due on April 5th, 2021 at 11:59 PM.**

In this lab you will learn how setup a Hadoop cluster and run MapReduce, Pig and Hive job.

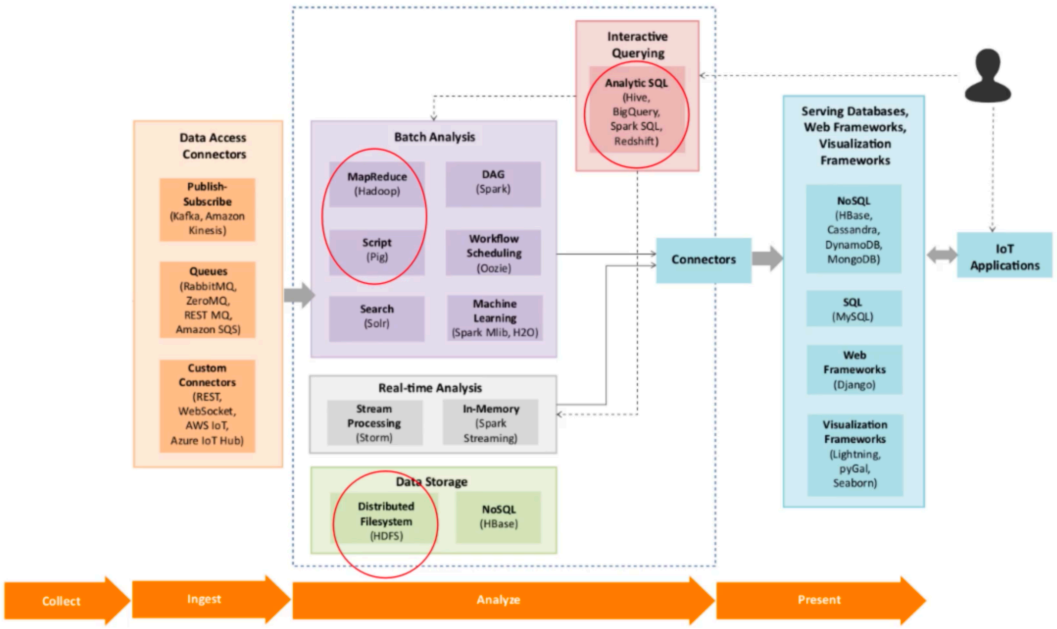
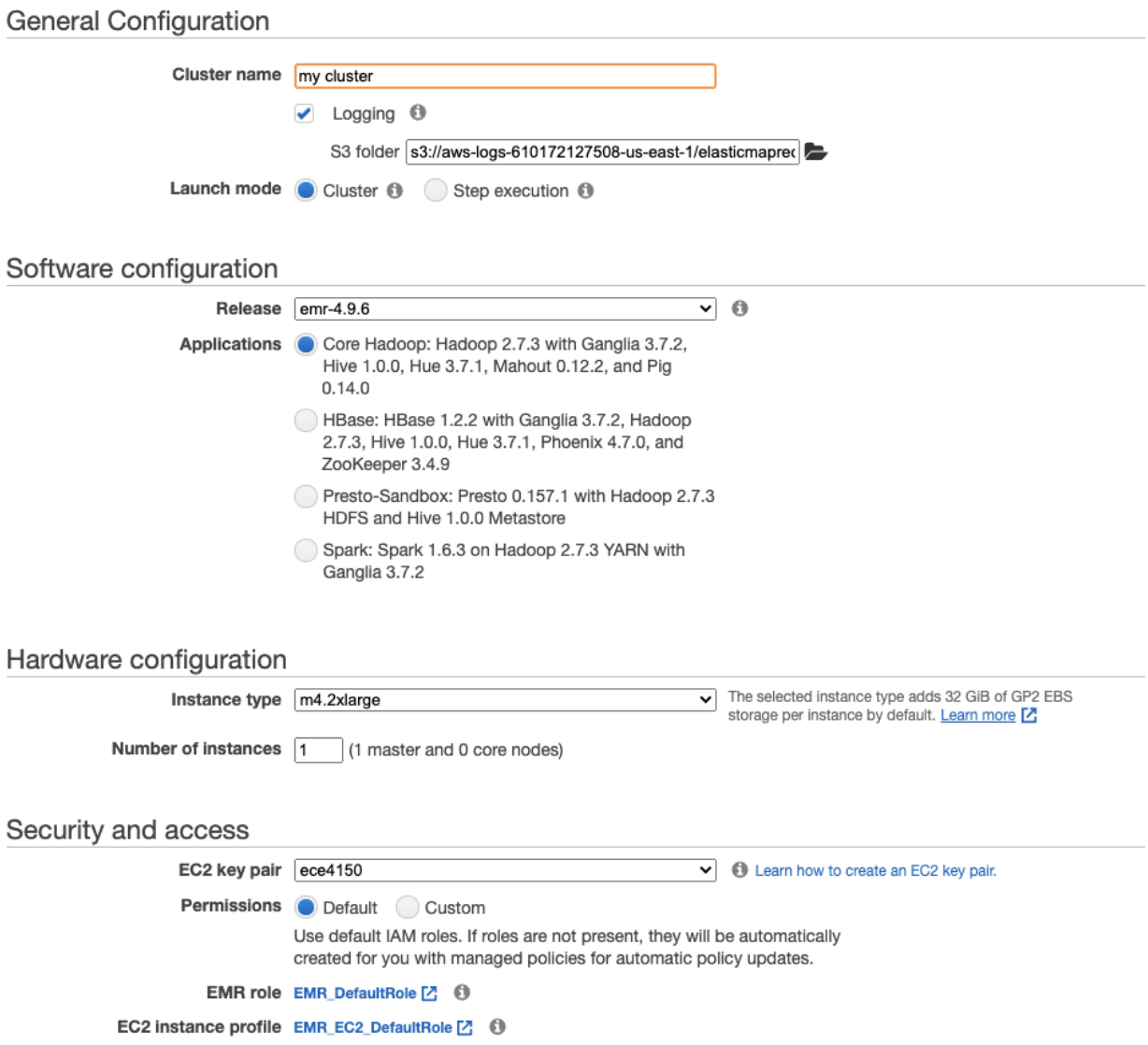


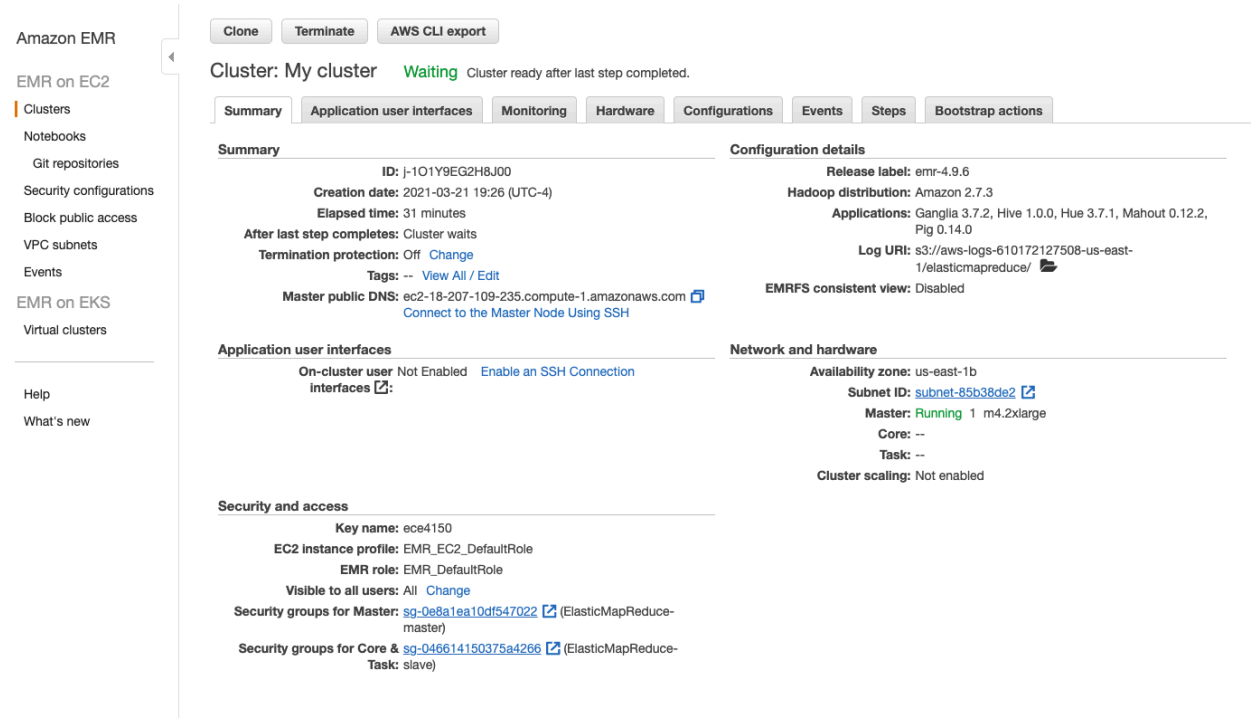
Fig.1 Architecture diagram of data processing in Hadoop

1. Set up a Hadoop Cluster with EMR

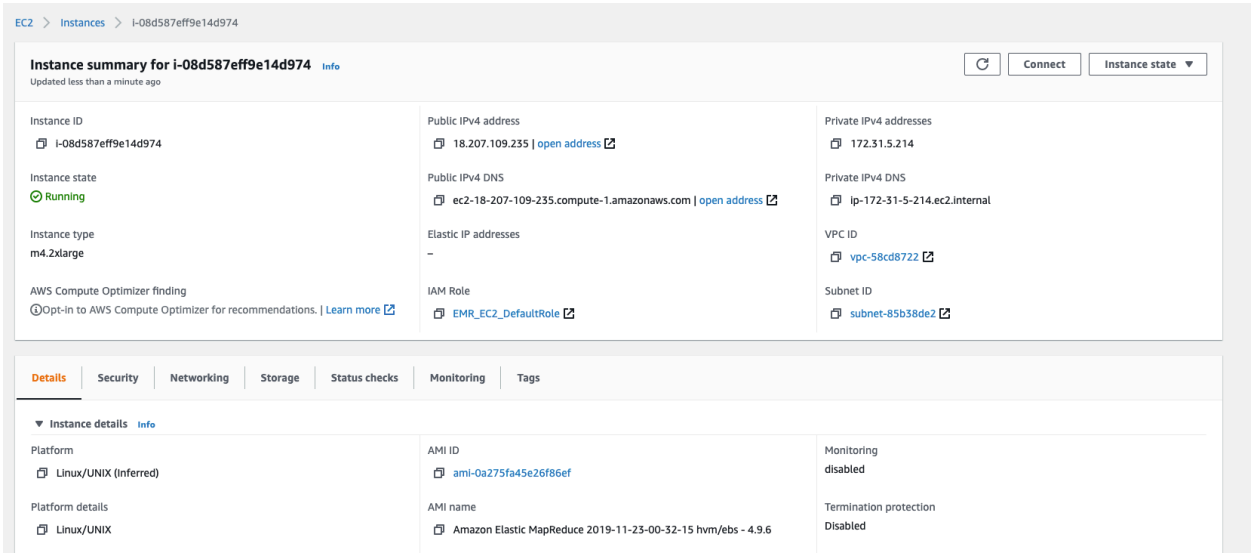
Navigate to Amazon EMR console and create a new cluster with the following configurations:



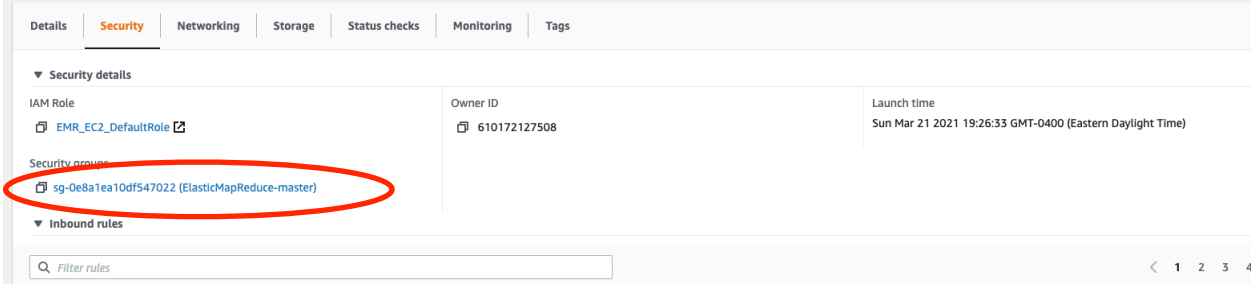
Wait for the cluster to be created and enter the state of “Waiting”, which usually takes 5 minutes to finish.



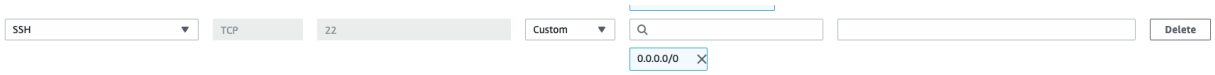
Navigate to **EC2 instance** and open the one that's running, which holds the cluster that you just created:



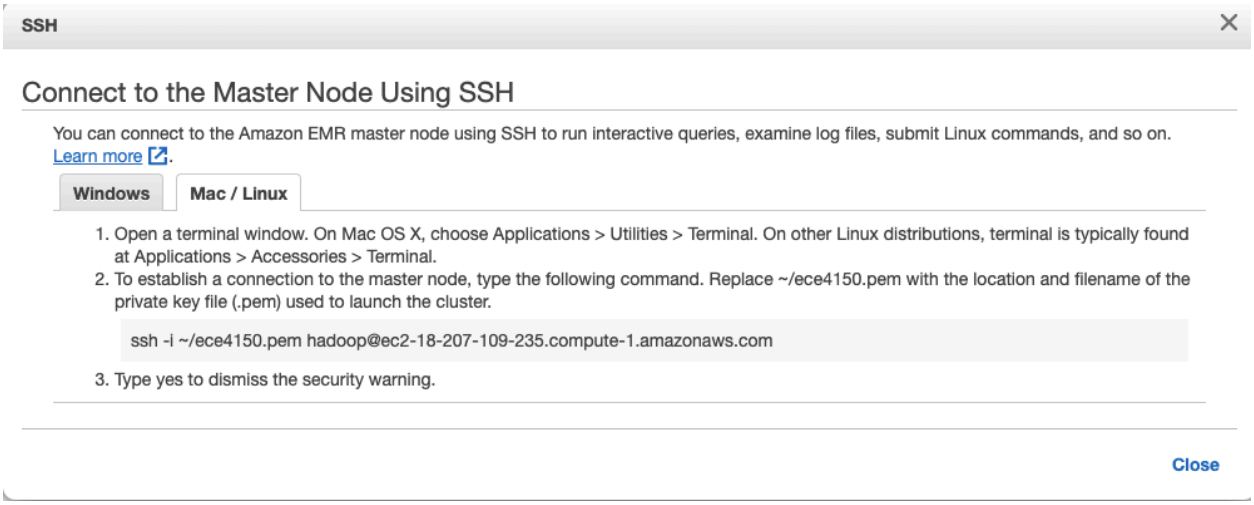
Go to **Security-Security groups** and open the security group for the master cluster, in this case **ElasticMapReduce-master**:



Edit the inbound rules and add a rule for SSH and save it:



Now go back to your **EMR cluster**, click on "**connect to the master node using SSH**" and follow the instruction to connect to the master node:

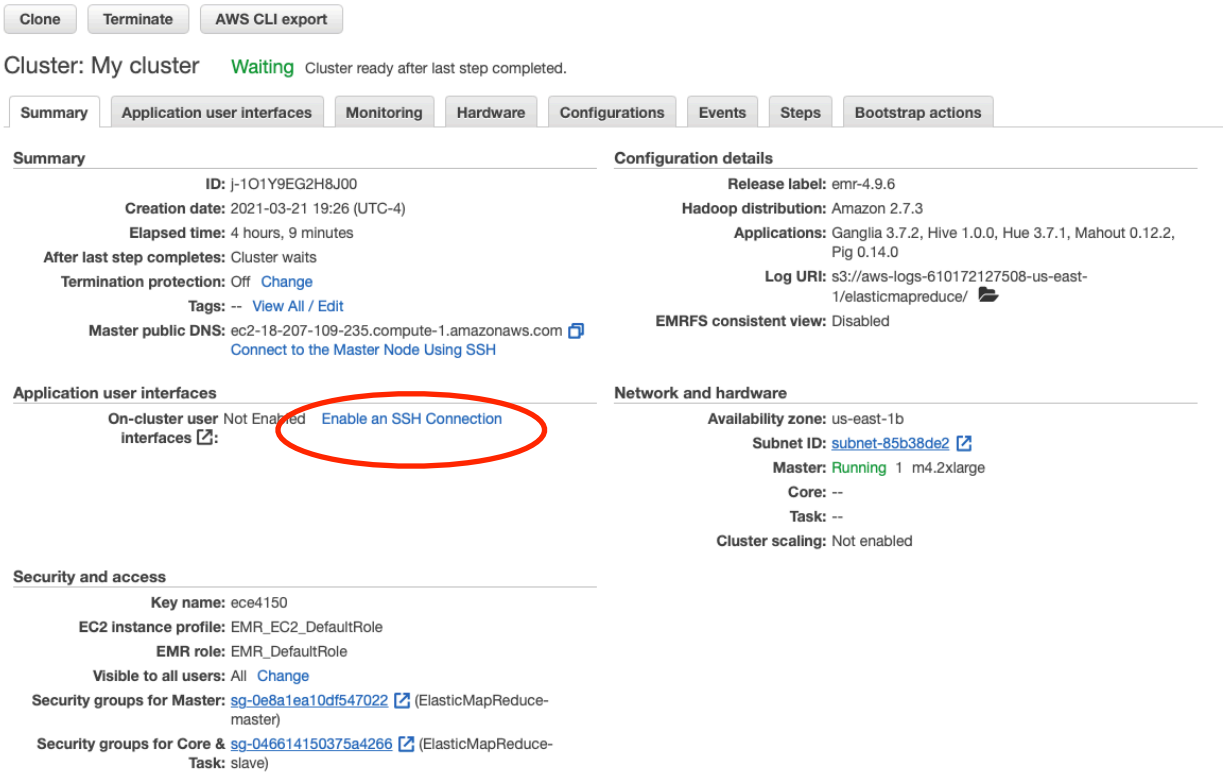


For example, a successful connection would appear like:

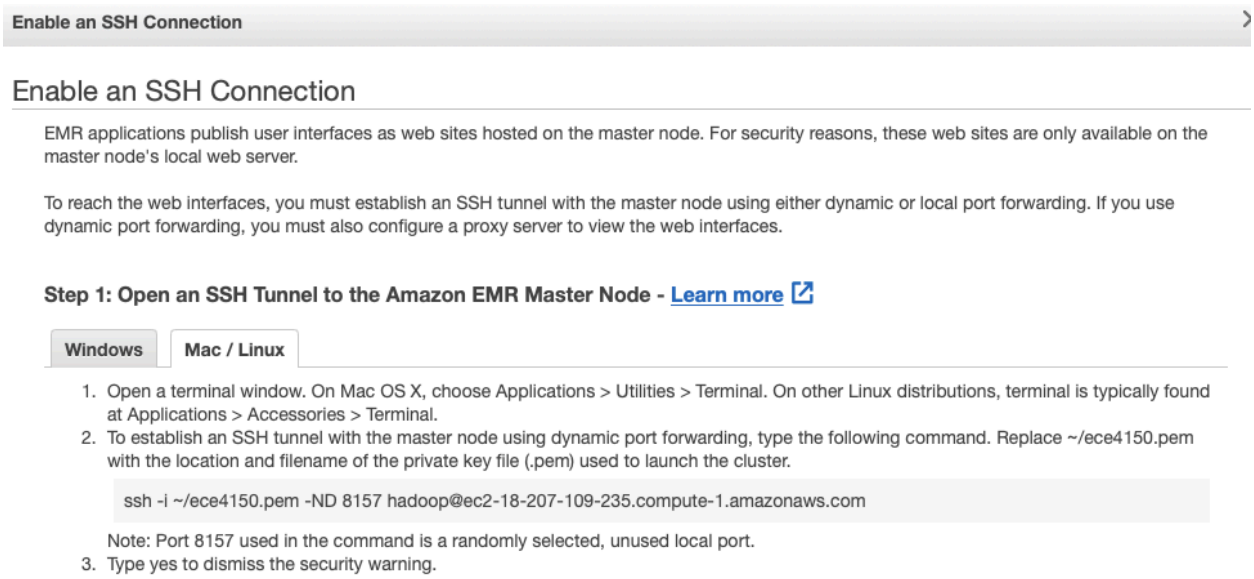
```
(base) [19:59] (/>ω<)/ ~/desktop $ chmod 400 ece4150.pem
(base) [19:59] (/>ω<)/ ~/desktop $ ssh -i ece4150.pem hadoop@ec2-18-207-109-235.compute-1.amazonaws.com
Last login: Sun Mar 21 23:54:02 2021

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/
Amazon Linux version 2018.03 is available.

EEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R:::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R::::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R::::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R::::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R::::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R::::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R::::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-5-214 ~]$
```

2. Upload Datasets to HDFS

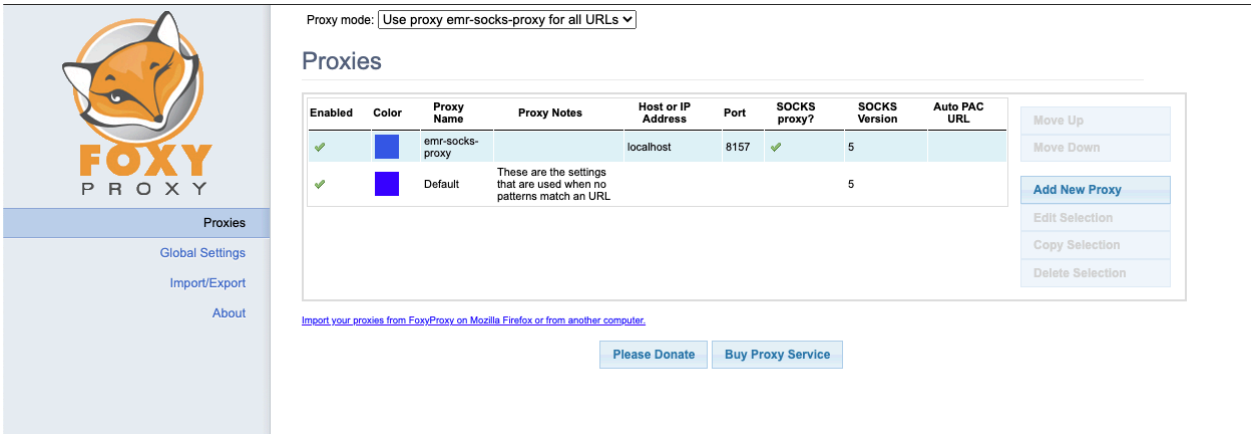
First, we need to enable SSH tunnel in the browser. Navigate to EMR cluster and click on "**Enable an SSH Connection**" in Application user interface:



First, follow the instructions to enable an SSH tunnel to the EMR Master Node:

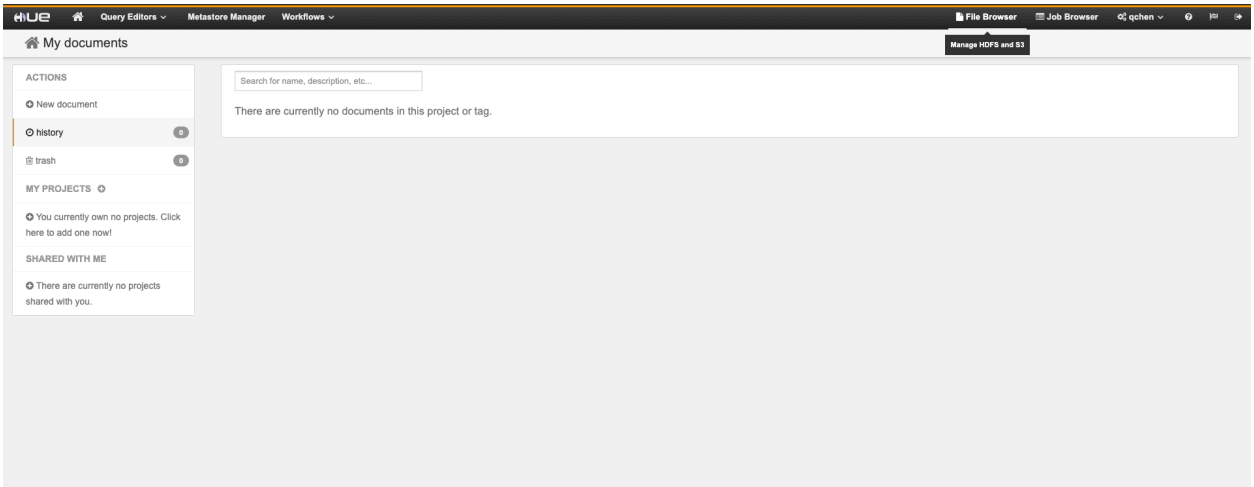


Then, install FoxyProxy in Chrome. It's worth noticing that the url provided in the instruction doesn't work, so please manually install it as an extension on you browser. Take Chrome as an example: go to **chrome store**, search for "**foxyproxy basic**", install and add to chrome, **restart** chrome after installing. Create a file named foxyproxy-settings.xml as suggested and import it to your FoxyProxy. At the top of your FoxyProxy page, choose "use proxy emr-socks-proxy for all URLs" (not the same as AWS instruction!!)

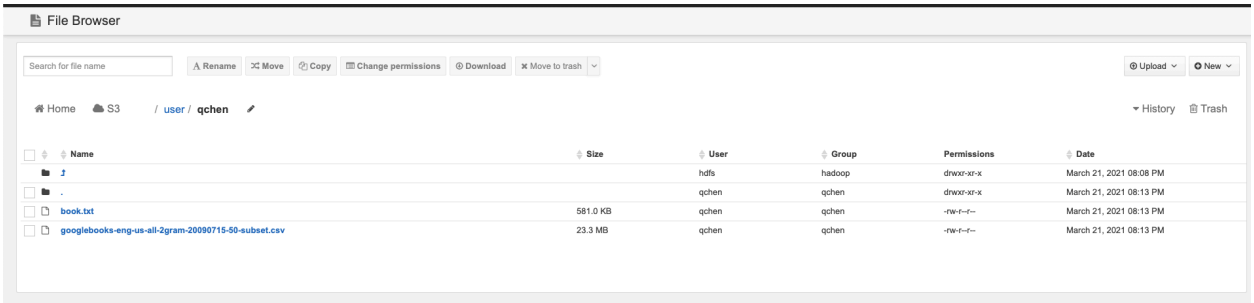


Go back to the EMR cluster and navigate to **Application user interfaces**, copy the url of Hue and open it in the browser in which you just installed the FoxyProxy, create a new account in Hue and **save the username**.

Follow the default setting in quick start and choose **Hue Home** in step 4 to enter the home page of Hue:



Upload the provided dataset to File Browser in the following structure (don't upload the zip file or folder, use only separate files)



Next, create the mrjob configuration file on the Hadoop master node with vim using the provided mrjob.conf, save and exit:

```
runners:
    hadoop:
        hadoop_home: /usr/lib/hadoop
        hadoop_streaming_jar: /usr/lib/hadoop-mapreduce/hadoop-streaming.jar
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"mrjob.conf" 5L, 136C                                         5,0-1         All
```

Create wordcount-mr.py on Hadoop master node with vim using the provided wordcount-mr.py, save and exit:

```
"""
python wordcount-mr.py -r hadoop hdfs:///user/clouduser/book.txt --output-dir=hd
fs:///user/clouduser/wordcountoutput --conf-path=mrjob.conf
"""

from mrjob.job import MRJob

class MRmyjob(MRJob):
        def mapper(self, _, line):
                wordlist = line.split()
                for word in wordlist:
                        yield word,1

        def reducer(self, key, list_of_values):
                yield key,sum(list_of_values)

if __name__ == '__main__':
    MRmyjob.run()
~
~
~
~
~
"wordcount-mr.py" 17L, 414C                                   17,5          All
```

3. Setup mrJob on Master Node

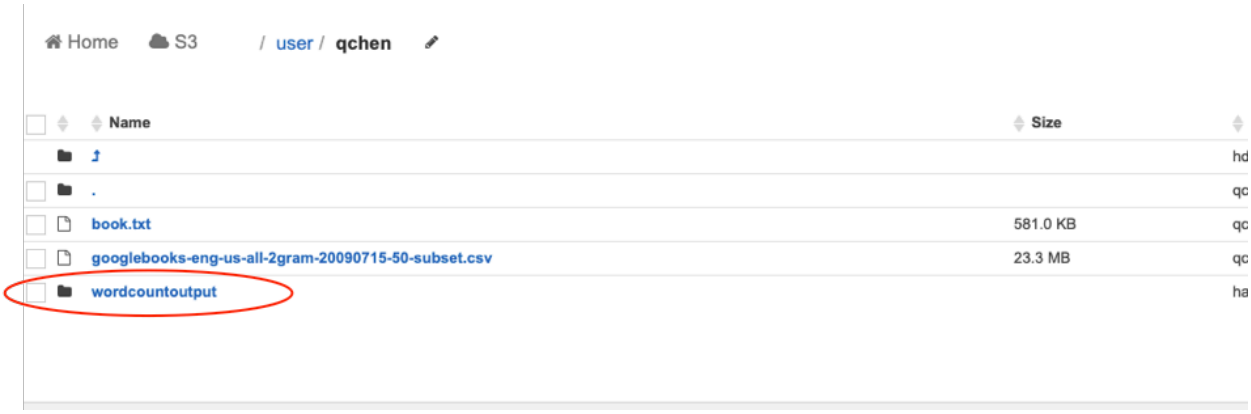
In the Hadoop instance, run the following commands to set up mrJob:

sudo yum install python-pip

sudo pip install mrjob

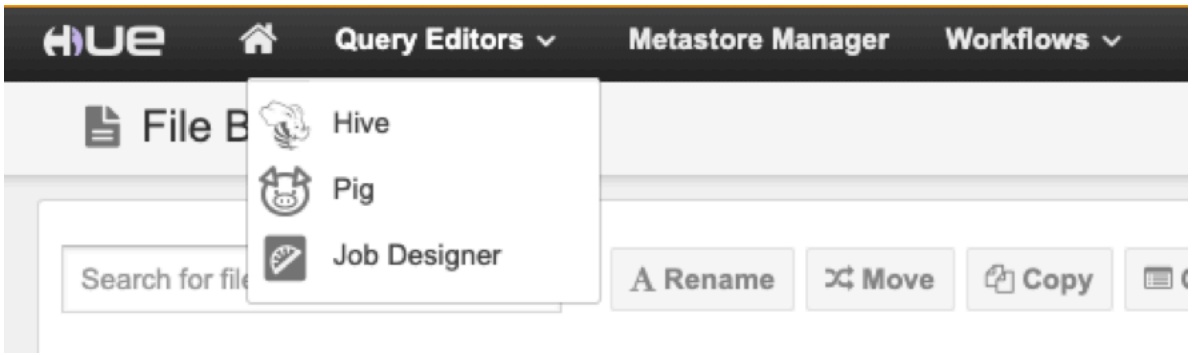
4. Run MapReduce Program

In the Hadoop instance, paste the command line from wordcount-mr.py and change the clouduser to your username. After the program finishes execution, you can view the results in the file browser:



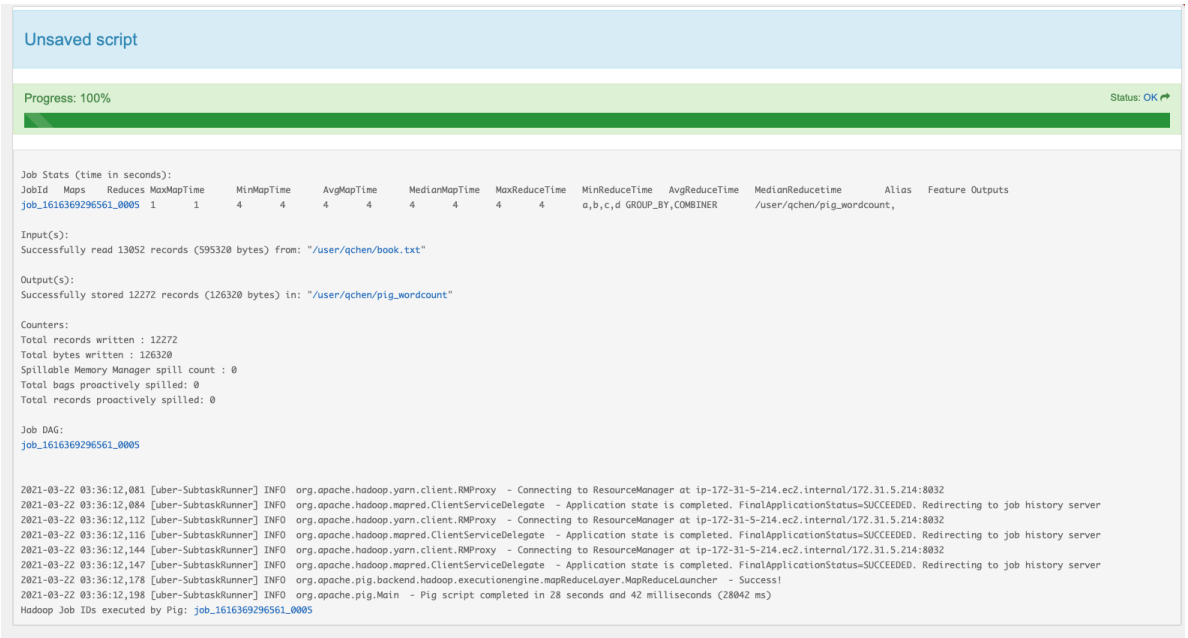
5. Run Pig Program

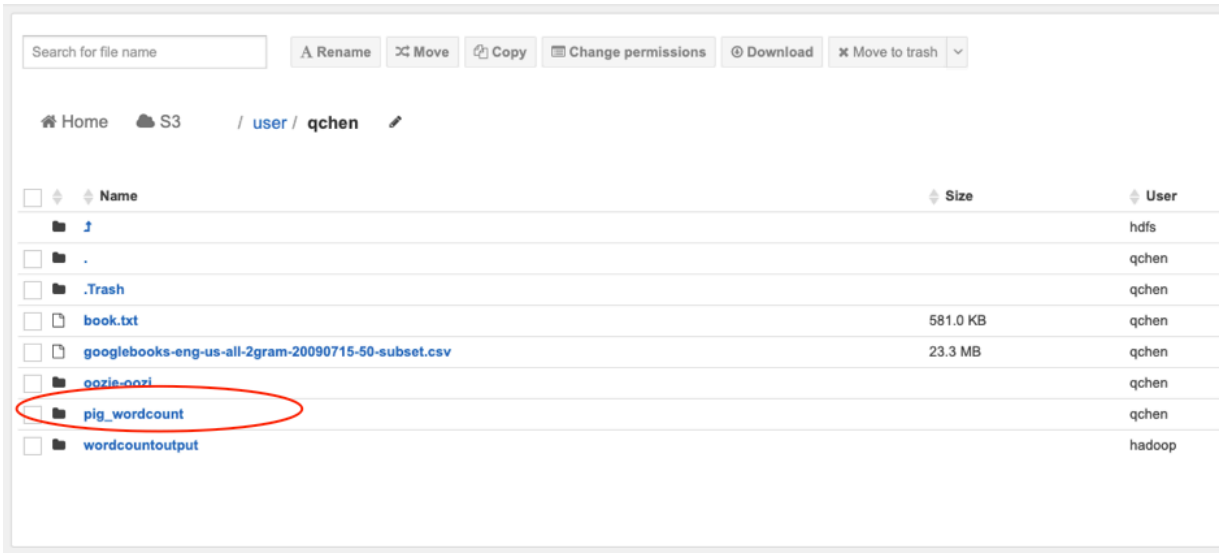
Find the provided code in wordcount-pig.txt, navigate to pig editor in Hue interface:



Paste the code in script, change “clouduser” to your own username and run:

After the job is finished, you can view the output in file browser:





6. Challenges (75%)

1. Implement a MapReduce program that emits the bigrams which were coined after year 1992 (or which started appearing after the year 1992).

   Output of the program should include: (bigram, year)

   Example output: (mobile phone, 1996) means that the bigram 'mobile phone' first appeared in the dataset in the year 1996.

2. Implement a MapReduce program that emits the average number of times each bigram appears in a book (over all the years). [Average for a particular n-gram is the total count of n-gram (over all the years) divided by the total number of books in which the n-gram appeared (over all the years)]

   Output of the program should include: (bigram, average)

   Example output: (how are, 6) means that the bigram 'how are' appears on average 6 times in a book (over all the years).

3. Implement a Pig program that computes the most common bigram in the year 2003 in thedataset (as determined by the count field).

   Output of the program should include: (bigram, count)

   Example output: (how many, 5001) means that the bigram 'how many' was the most popular bigram in the year 2002 and it appeared a total of 5001 times in all the books in that year.

4. Implement a Pig program that computes the most common bigram in each year in the dataset (as determined by the count field).

   Output of the program should include: (year, bigram, count)

   Example output: (2003, mobile phone, 3012) means that in the year 2003 the most popular bigram was 'mobile phone' and it appeared 3012 times in all the books in that year. Emit such tuples for each year in the dataset.

5. Create a Hive meta-store table from the N-Gram dataset (CSV) file from the Hue web interface.

   Implement a Hive query (in the SQL-like Hive Query Language) to find the most popular bigram (over all the years).

Deliverables

1. The complete code with the modifications needed to complete each exercise, including the new lambda function.
2. Output files (.txt or .csv) that contain results for each exercise program.