Harsh Sharma

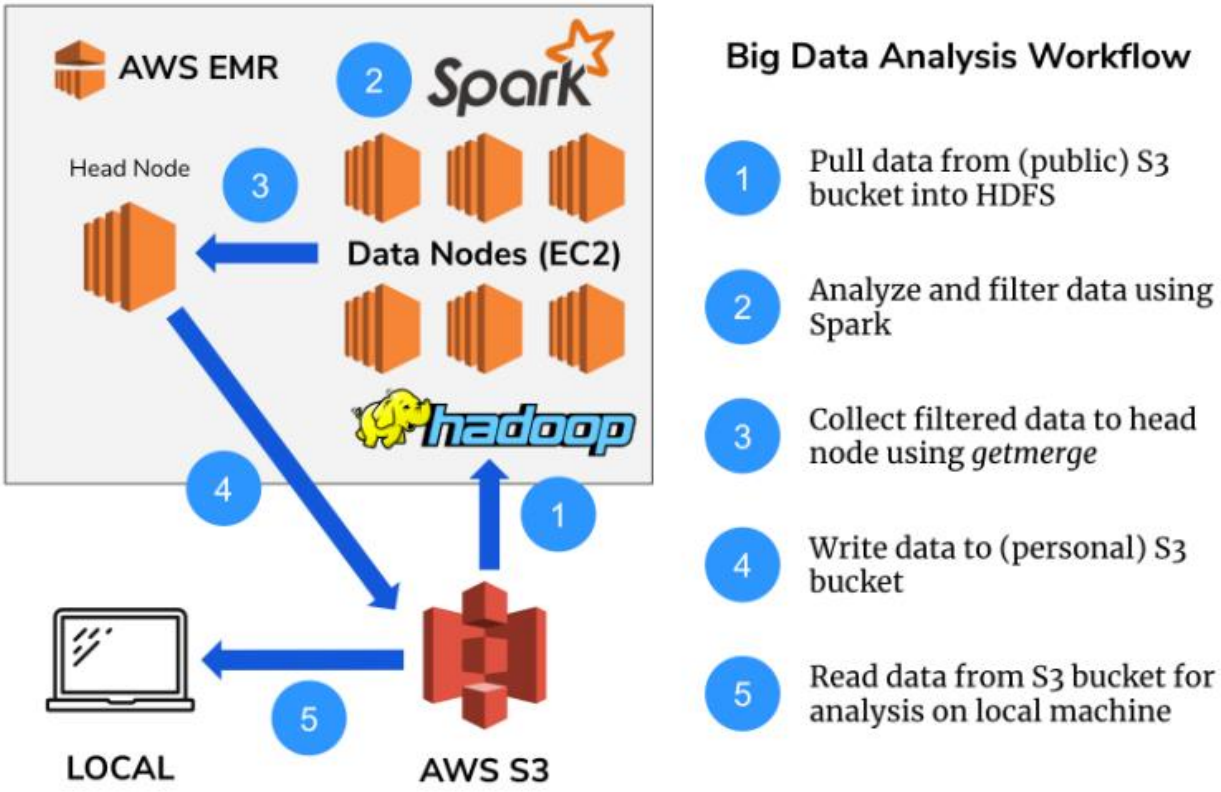
Big Data assignment (Unit 4) Report

5 December 2022

Submitted To: Borna Ghotbi, Fallyn Thompson, Arati Santhanakrishnan

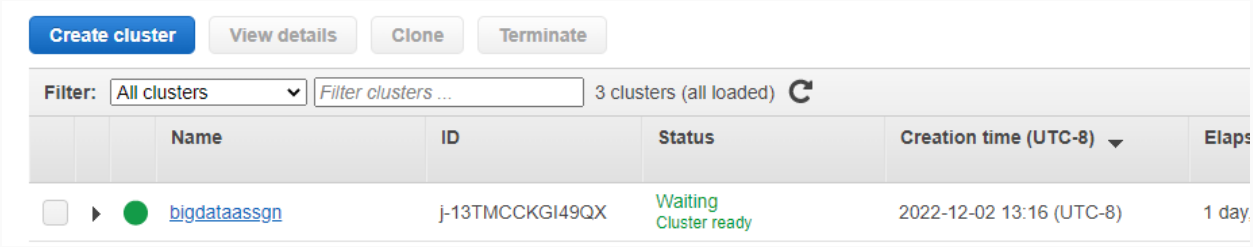
---

Prepare a professional report to summarize the findings and be sure to include an appendix with screenshots of the steps completed for Questions 1 and 2.



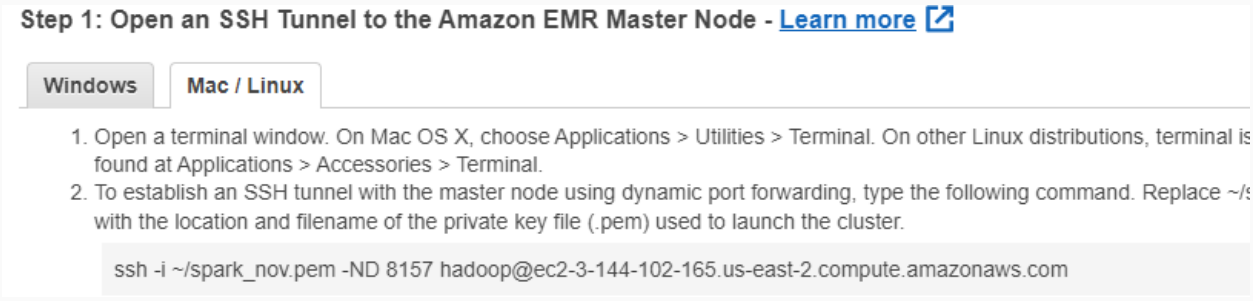
The following screenshots/ snippets are the replication of the above flow process.

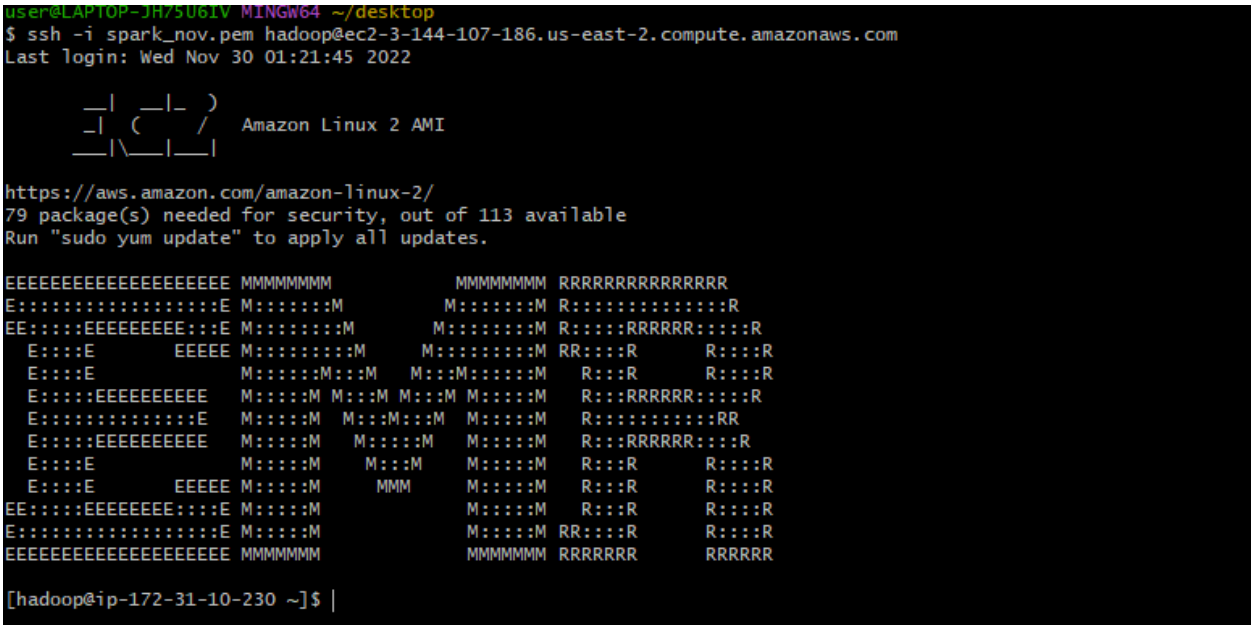
Question 1 Creating a cluster



If you want to create a cluster, you can click on the create cluster. For this assignment I created **bigdataassgn** as my cluster.

Question 2 Connect to the head node of the cluster using SSH.





From the snippet, I have connected to server on my local windows machine.

Question 3 Copy the data folder from the S3 bucket *directly* into a directory on the Hadoop File System (HDFS) named `/user/hadoop/eng_1M_1gram`

1. Checking if I have anything in the hdfs.

```
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -ls
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -lsr
lsr: DEPRECATED: Please use 'ls -R' instead.
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -ls -R
[hadoop@ip-172-31-15-205 ~]$ client_loop: send disconnect: Connection reset by peer
hh(base)
```

From above, it can be observed that there is nothing in the hdfs.

2. Create a directory a do a sanity check

```
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -mkdir /user/hadoop/eng_1M_1gram
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -ls
Found 1 items
drwxr-xr-x   - hadoop hadoop          0 2022-12-02 23:56 eng_1M_1gram
```

So now I have eng_1M_1gram directory in hdfs.

3. Copying the .csv file from the S3 brainstation bucket DIRECTLY to the HDFS using ***distcp.***

```
[hadoop@ip-172-31-15-205 ~]$ hadoop distcp s3://brainstation-dsft/eng_1M_1gram.csv /user/hadoop/eng_1M_1gram/eng_1M_1gram.csv
2022-12-03 00:00:34,547 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwri
false, useRdiff=false, fromSnapshot=null, toSnapshot=null, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBandwidth=0.0, copyStrategy='uni
SIZE], atomicWorkPath=null, logPath=null, sourceFileListing=null, sourcePaths=[s3://brainstation-dsft/eng_1M_1gram.csv], targetPath=/user/hadoop/eng_1M_1gram/en
l', blocksPerChunk=0, copyBufferSize=8192, verboseLog=false, directWrite=false}, sourcePaths=[s3://brainstation-dsft/eng_1M_1gram.csv], targetPathExists=false,
2022-12-03 00:00:34,821 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-15-205.us-east-2.compute.internal/172.31.15.205:8032
2022-12-03 00:00:34,998 INFO client.AHSProxy: Connecting to Application History server at ip-172-31-15-205.us-east-2.compute.internal/172.31.15.205:10200
2022-12-03 00:00:38,517 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
2022-12-03 00:00:38,517 INFO tools.SimpleCopyListing: Build file listing completed
```

4. Sanity Check to see if the data is really copied or not.

```
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -ls
Found 1 items
drwxr-xr-x   - hadoop hadoop          0 2022-12-03 00:01 eng_1M_1gram
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -lsr
lsr: DEPRECATED: Please use 'ls -R' instead.
drwxr-xr-x   - hadoop hadoop          0 2022-12-03 00:01 eng_1M_1gram
-rw-r--r--   1 hadoop hadoop 5292105197 2022-12-03 00:01 eng_1M_1gram/eng_1M_1gram.csv
[hadoop@ip-172-31-15-205 ~]$ pwd
/home/hadoop
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -ls /user/hadoop/eng_1M_1gram/
Found 1 items
-rw-r--r--   1 hadoop hadoop 5292105197 2022-12-03 00:01 /user/hadoop/eng_1M_1gram/eng_1M_1gram.csv
```

You can see that eng_1M_1gram.csv file is sitting in eng_1M_1gram directory in HDFS.

5. Make a new dataframe and writing it into the directory.

```
3]: df2.write.csv("/user/hadoop/eng_1M_1gram/datatoken.csv", header=True)
```

▸ Spark Job Progress

After doing this I wanted it to appear in the directory in HDFS.

```
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -ls
Found 2 items
drwxr-xr-x   - hadoop hadoop          0 2022-12-03 00:01 eng_1M_1gram
drwxr-xr-x   - livy   hadoop          0 2022-12-03 18:16 eng_1M_1gram2
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -lsr
lsr: DEPRECATED: Please use 'ls -R' instead.
drwxr-xr-x   - hadoop hadoop          0 2022-12-03 00:01 eng_1M_1gram
-rw-r--r--   1 hadoop hadoop 5292105197 2022-12-03 00:01 eng_1M_1gram/eng_1M_1gram.csv
drwxr-xr-x   - livy   hadoop          0 2022-12-03 18:16 eng_1M_1gram2
drwxr-xr-x   - livy   hadoop          0 2022-12-03 18:17 eng_1M_1gram2/datatoken.csv
-rw-r--r--   1 livy   hadoop          0 2022-12-03 18:17 eng_1M_1gram2/datatoken.csv/_SUCCESS
-rw-r--r--   1 livy   hadoop         33 2022-12-03 18:16 eng_1M_1gram2/datatoken.csv/part-00000-88ec2ae9-7637-400c-91f3-9f67125cfc84-c000.csv
-rw-r--r--   1 livy   hadoop       7305 2022-12-03 18:16 eng_1M_1gram2/datatoken.csv/part-00023-88ec2ae9-7637-400c-91f3-9f67125cfc84-c000.csv
```

The datatoken.csv file was saved as a directory itself as it can be observed that memory occupied by it is 0. However, 2 separate csv are created as can be seen in last 2 lines.

6. Next Step: Merging these 2 csvs' on the local drive of the head node.

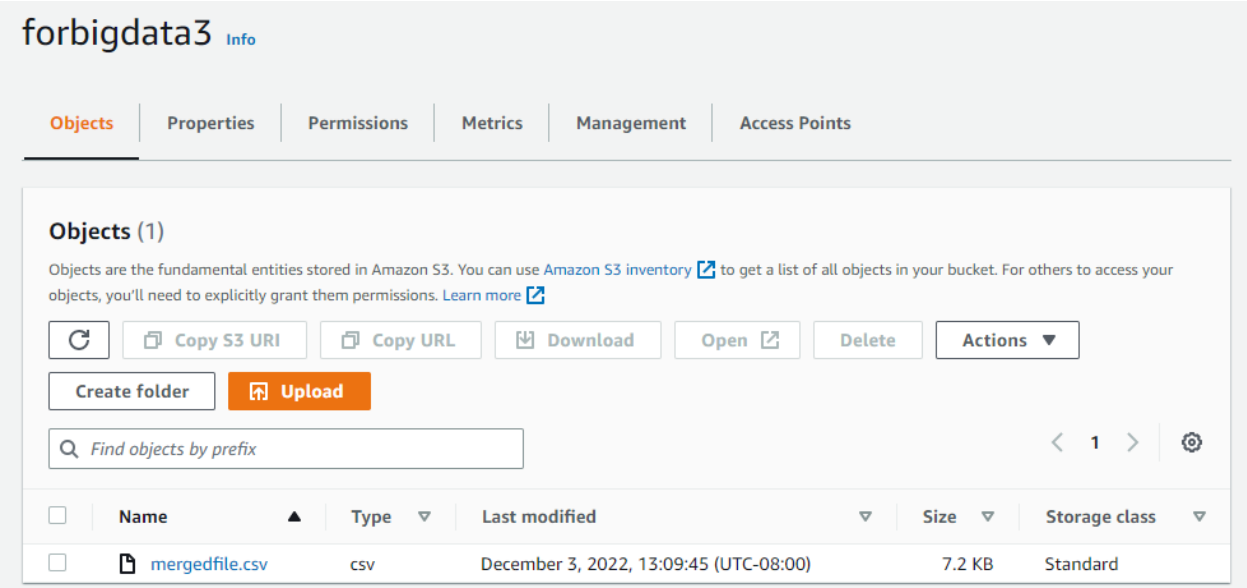
```
[hadoop@ip-172-31-15-205 ~]$ hadoop fs -getmerge /user/hadoop/eng_1M_1gram2/datatoken.csv/part-00000-88ec2ae9-7637-400c-91f3-9f67125cfc84-c000.csv /user/hadoop/eng_1M_1gram2/datatoken.csv/part
-00023-88ec2ae9-7637-400c-91f3-9f67125cfc84-c000.csv mergedfile.csv
[hadoop@ip-172-31-15-205 ~]$ cat mergedfile.csv
token,year,frequency,pages,books
token,year,frequency,pages,books
data,1584,16,14,1
data,1614,3,2,1
data,1627,1,1,1
data,1631,22,18,1
data,1637,1,1,1
data,1638,2,2,1
data,1640,1,1,1
data,1642,1,1,1
data,1644,4,4,1
data,1647,1,1,1
data,1651,1,1,1
```

Also after doing the sanity check you can see (using cat mergedfile.csv) that the 2 csvs' are now combined/merged.

However, the name of the columns also appear 2 times as can be seen from above.

7. Let's move this into the S3 bucket (forbigdata3) and then try to open it without spark in jupyter notebook with pandas.

```
[hadoop@ip-172-31-15-205 ~]$ aws s3 cp mergedfile.csv s3://forbigdata3
upload: ./mergedfile.csv to s3://forbigdata3/mergedfile.csv
```



Now you can open this mergedfile.csv into jupyter notebook using Pandas.

Question 8: Compare Hadoop and Spark as distributed file systems.

a. What are the advantages/ differences between Hadoop and Spark? List two advantages for each.
b. Explain how the HDFS stores the data.

1. **Advantages: Hadoop**
   a) It is very highly scalable storage platform because it can store and distribute very large datasets across hundreds of servers that work in parallel.
   b) Hadoop is fault tolerant. When data is sent to a individual node, that data is also replicated to other nodes in the cluster, which means that in the event of failure, there is another copy available to use.

Advantages: Spark

a) Spark has all the advantages as Hadoop has. But Spark offers outstanding streaming with the latest methods for writing and maintaining streaming code. By spark streaming, you can use a variety of applications such as stream mining, network optimization, real time scoring of analytical models and more.
b) Spark provides python addition, also called PySpark which let programmers to interface with the Spark framework and learn how to manipulate data at a large scale.

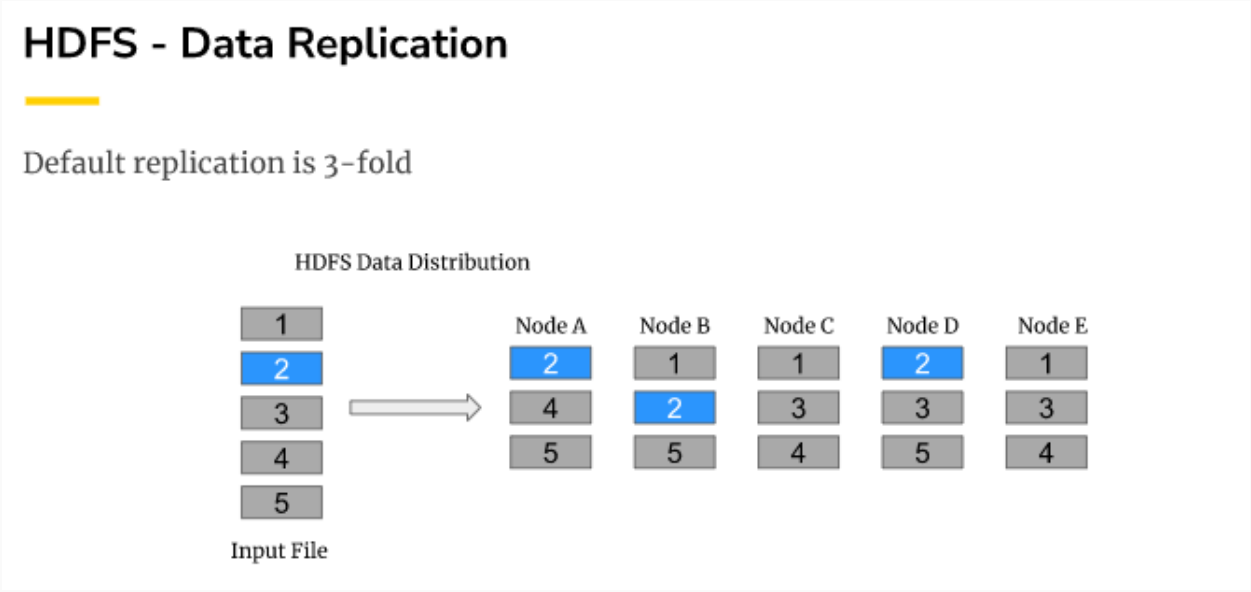
Difference between Spark and Hadoop.

Unlike Hadoop, Spark does not write to disk but keeps data in memory and so is typically much faster for performing data engineering, analysis, and modeling tasks within Big Data environments and with large data sets.
Spark runs 100 times faster in memory and ten times faster on disk than Hadoop MapReduce.

Question 8.b Explain how HDFS store the data.

Answer:

In HDFS, data is split into blocks typically 64MB or 128MB. And these blocks are split into machines at load time. Also blocks are replicated across multiple machines as shown below.



NameNode keeps track of which blocks make up a file and where they are stored.