1) Start your EMR cluster
Go to https://console.aws.amazon.com/
Go to EMR by typing EMR to AWS Service search bar
Click to Create Cluster button
Give a name to your cluster e.g "Spark Cluster with 4 nodes"
Under "Software configuration" select Spark
Under "Hardware configuration" enter 5 for number of clusters (THIS STEP IS OPTIONAL!)
Under "Security and Access" select a key pair if you already created one, otherwise click on the link "Learn how to create an EC2 key pair." next to dropdown menu. Please read the instructions carefully.

Click to Create Cluster button when you finished with key pair selection.

On the next screen, you have to wait until the state(green text next to "Cluster Name") of cluster changed from "Starting" to "Waiting".

When the cluster is ready you can connect it by using "Master public DNS" information. This info might be missing just wait for it to appear.
If you don't know how to use SSH please click the "SSH" link next to "Master public DNS" value and follow instructions carefully.

2) Preparing Dataset
You have 2 options to use the dataset.
Either you can use the S3 link given below or you can download a script on your master node and run it to upload the dataset to your spark cluster's hadoop storage. Second method takes a little bit time to prepare but you will have a copy of your dataset on your cluster which make it faster. S3 might perform slower as it might be shared by multiple students. **Note: We strongly suggest you to use HDFS option!!!**

S3 link: **s3://dal-cs-csci6515-enron/maildir/**
(Please see the sample code at the end of this document about how to connect S3)

2.1) Uploading dataset to HDFS
after you logged in to master server type following commands

```
wget http://web.cs.dal.ca/~gercek/spark/script/upload_all_documents.sh
chmod +x ./upload_all_documents.sh
nohup ./upload_all_documents.sh > /mnt/upload.log &
```

More info about nohup:
https://www.cyberciti.biz/tips/nohup-execute-commands-after-you-exit-from-a-shell-prompt.html

Note: you can check the progress by typing this command

```
tail -f /mnt/upload.log
```

Press Ctrl + C to stop it.

The script(upload_all_documents.sh) will complete following tasks;

1. download the enron_dataset and extract it to /mnt/enron_tmp/maildir
2. create maildir folder on HDFS
3. for each user;

1. create /maildir/{username} folder on HDFS
2. upload "all_documents" folder to HDFS /maildir/{username}/all_documents

You might see following warning messages during the script execution but they are safe to ignore;

warning(1)
```
put: `./maildir/hyatt-k/all_documents': No such file or directory
```

Some users in the dataset do not have "all_documents" folder.

warning(2)
```
WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1249)
        at java.lang.Thread.join(Thread.java:1323)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:609)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:370)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:546)
```

It is a known bug and can be ignored. https://issues.apache.org/jira/browse/HDFS-10429

After everything is finished ("END OF SCRIPT!!!" should be in /mnt/upload.log file) you can test your spark cluster with pyspark shell.

3) Check HDFS

to list the folders

```
hadoop fs -ls /
```

to list files of enron dataset

```
hadoop fs -ls /maildir
```

Further commands can be found:

https://dzone.com/articles/top-10-hadoop-shell-commands
http://hortonworks.com/hadoop-tutorial/using-commandline-manage-files-hdfs/

List of hadoop fs <args>

https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-common/FileSystemShell.html

4) Test Spark Cluster

4.1) PySpark with IPython (optional)

if you want to install IPython type;

```
sudo pip install ipython
```

to start PySpark

```
PYSPARK_DRIVER_PYTHON=/usr/local/bin/ipython pyspark
```

4.2) PySpark Without IPython

to start PySpark

```
pyspark
```

Testing S3 Bucket: paste following command to pyspark shell. This will return first line of that file. (S3 will be slower compare to HDFS)

```
testRdd = sc.textFile("s3://dal-cs-csci6515-enron/maildir/allen-p/all_documents/1.")
testRdd.take(1)
```

Testing HDFS: paste following command to pyspark shell. This will return first line of that file.

```
testRdd = sc.textFile("hdfs:///maildir/allen-p/all_documents/1.")
testRdd.take(1)
```

5) Submit Spark App

To submit your app to your cluster, you need to upload your code to Master node. There are 2 methods you can use for that.

1) You can copy and paste content of your files while an SSH connection is open. For example, assume that you copied the content of mysparkapp.py.
Type following command after you logged in to Master Node;

```
nano mysparkapp.py
```

this command will open the file with a text editor program called nano. Paste the content of your file and hit CTRL + O and hit Enter. To exit from editor use CTRL + X. For more information: https://www.nano-editor.org/dist/v2.2/nano.html#Editor-Basics

2) You can use “scp”. This program uses ssh to transfer files between computers. To transfer mysparkapp.py file to your Master node use following command;

```
scp -i myprivatekey.pem mysparkapp.py hadoop@masternode.public.dns:~/mysparkapp.py
```

You should replace myprivatekey.pem and hadoop@masternode.public.dns values with your cluster’s private key and Master Public DNS values. Please see “Start your EMR Cluster” section of this document.

After uploading your spark application just call spark-submit

6) Basic Code Structure For the Assignment

You can download a sample code that will read and parse the whole dataset. Type following commands after logged in to Master cluster type following command to run the sample code.

```
wget http://web.cs.dal.ca/~gercek/spark/script/assignment_sample.py
spark-submit assignment_sample.py
```

this code will print 10 sample from the dataset. If you want you can save the output by uncommenting last line.