Nikhil Bhalala
Dsc -333
Assignment-2

Part 1

1) Describe how you would implement a MapReduce job consisting of Map and Reduce description. You can describe it in your own words or as pseudo-code. Keep in mind that map task reads the input file and produces (key, value) pairs. Reduce task takes a list of (key, value) pairs for each key and combines all values for each key.
Remember that Map operates on individual blocks and Reduce on individual keys with a set of values. Thus, for Mapper you need to state what your code does given a block of data and for Reduce you need to state what your reducer does <u>for each key</u>. You can assume that all of the columns accessed by the query exist in the original table.

a) SELECT Last, AVG(Grade)
FROM Student
GROUP BY Last;

⇨ Map: {Last: Student}
Mapper will read it into Key "Last" and Value Student
Reduce: same last in to reducer, {Last: Value: AVG(Grade)}

b) SELECT City, State, COUNT(DISTINCT Name)
FROM Student
GROUP BY City, State;

⇨ Map: {city_state: Student}
Mapper will read it into Key "city_state" value Student
Reduce: same last in to reducer, {city_state: Value: count(DISTINCT Name)}

c) SELECT a.First, a.Last, e.EID, a.AID, e.Age
FROM Employee as emp, Agent as a
WHERE emp.Last = a.Last AND emp.First = a.First;

⇨ Map Employee: {e.First_e.Last: e.EID, e.Age}
Map Agent: {a.First_a.Last: AID }
Reducer read it into Key "e.First_e.Last" and "a.First_a.Last" to combine both key
Reduce: same Year, Month in to reducer, { a.First_a.Last : e.EID, a.AID, e.Age}

2) Suppose you are tasked with analysis of the company's web server logs. The log dump contains a large amount of information with up to 9 different attributes (columns). You regularly run a Hadoop job to perform analysis pertaining to 3 specific attributes – TimeOfAccess, OriginOfAccess and FileName out of 9 total in the file.

a) How would you attempt to speed up the regular execution of the query? (2-a is intentionally an open-ended question, there are several acceptable answers)

⇨ Make the reducer smaller so that it can perform more reducer operations simultaneously.

b) If a Mapper task fails while processing a block of data – which node(s) would be preferred to restart it?

⇨ The job tracker pinpoints the problem and restarts the map process on the same block of data from the beginning where it got halted and stopped if the mapper task fails while processing the block of data. Any accessible node with an anticipated block may be given the mapper task.The job from the mapper will be assigned to the available node with the anticipated block and will wait for the result of the node which finishes the process first, though if the mapper task is taking a very long time to complete the process and the process on other blocks is successfully completed.

c) If the job is executed with 4 Reducers

i) How many files does the output generate?

⇨ As 4 reducers are needed to complete the job, four files will be produced by each reducer. As each reducer creates a file, Hadoop distributed file systems will store all files.

ii) Suggest one possible hash function that may be used to assign keys to reducers.

⇨ x mod 2 Equals H(x) one of the most used hash functions for creating keys to submit to reducers in mapper functions.

3) Consider a Hadoop job that processes an input data file of size equal to 165 disk blocks (165 different blocks, you can assume that HDFS replication factor is set to 1). The mapper in this job requires 1 minute to read and process a single block of data. For the purposes of this assignment, you can assume that the reduce part of this job takes zero time. You can also refer to the supplemental example on how to make this estimate.

a) Approximately how long will it take to process the file if you only had one Hadoop worker node? You can assume that that only one mapper is created on every node.

⇨ 165*1 =165
1*16500/60 =275

b) 10 Hadoop worker nodes?

⇨ 1min+16500/10*1=60+1650 =1710

c) 30 Hadoop worker nodes?

⇨ 1min+16500/30*1=60+550 =610

d) 100 Hadoop worker nodes?

⇨ 1min+16500/100*1=60+165 =225

e) Now suppose you were told that the replication factor has been changed to 3. That is, each block is stored in triplicate, but file size is still 165 blocks. Which of the answers (if any) in a)-d) above will have to change?

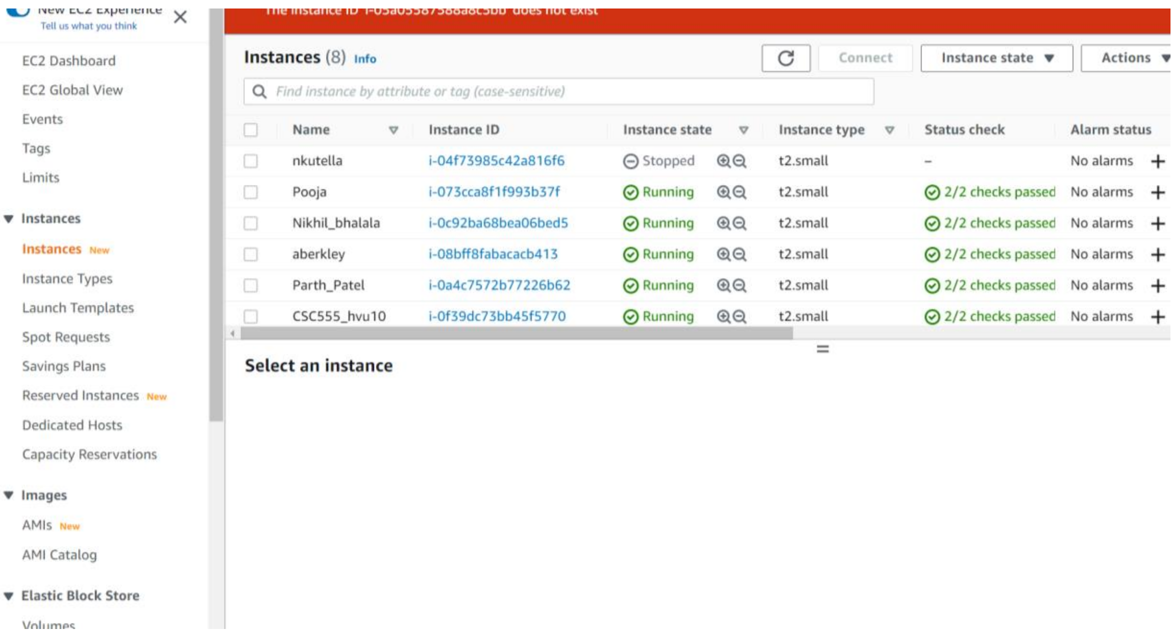
Nikhil Bhalala
Dsc -333
Assignment-2

You can ignore the network transfer costs and other potential overheads as well as the possibility of node failure. State any assumptions you make.

⇨ Since we are ignoring the network transfer charges, other potential overheads, as well as the risk of node failure, it won't have any effect on the processing time of 165 disk blocks. Replication enters the picture in case of node failure.

Part 2: Linux Intro

0. Login to your Amazon EC2 Instance



The instance ID 'i-03a05387588a8c5bb' does not exist

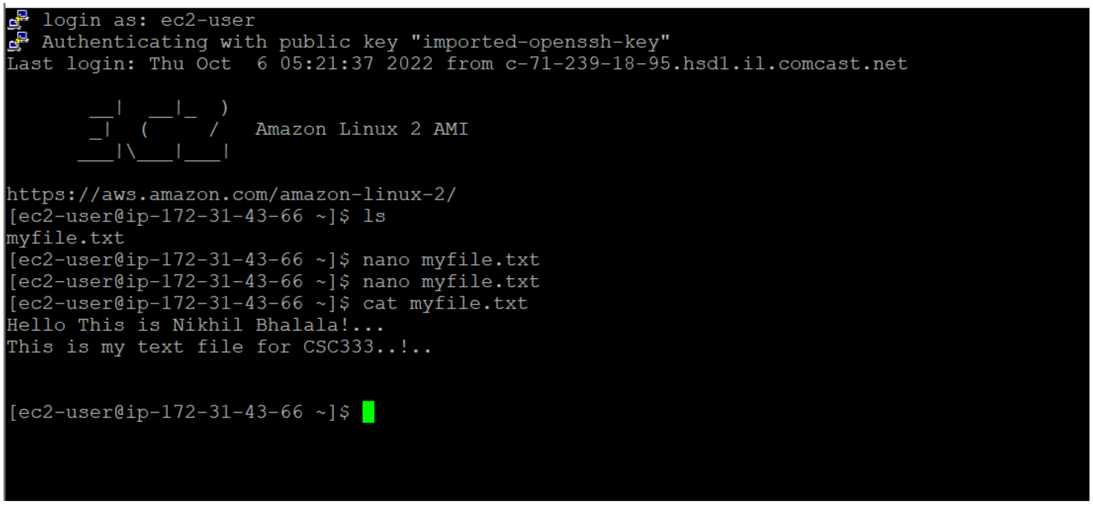
Instances (8) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	nkutella	i-04f73985c42a816f6	Stopped	t2.small	–	No alarms
□	Pooja	i-073cca8f1f993b37f	Running	t2.small	2/2 checks passed	No alarms
□	Nikhil_bhalala	i-0c92ba68bea06bed5	Running	t2.small	2/2 checks passed	No alarms
□	aberkley	i-08bff8fabacacb413	Running	t2.small	2/2 checks passed	No alarms
□	Parth_Patel	i-0a4c7572b77226b62	Running	t2.small	2/2 checks passed	No alarms
□	CSC555_hvu10	i-0f39dc73bb45f5770	Running	t2.small	2/2 checks passed	No alarms

Select an instance

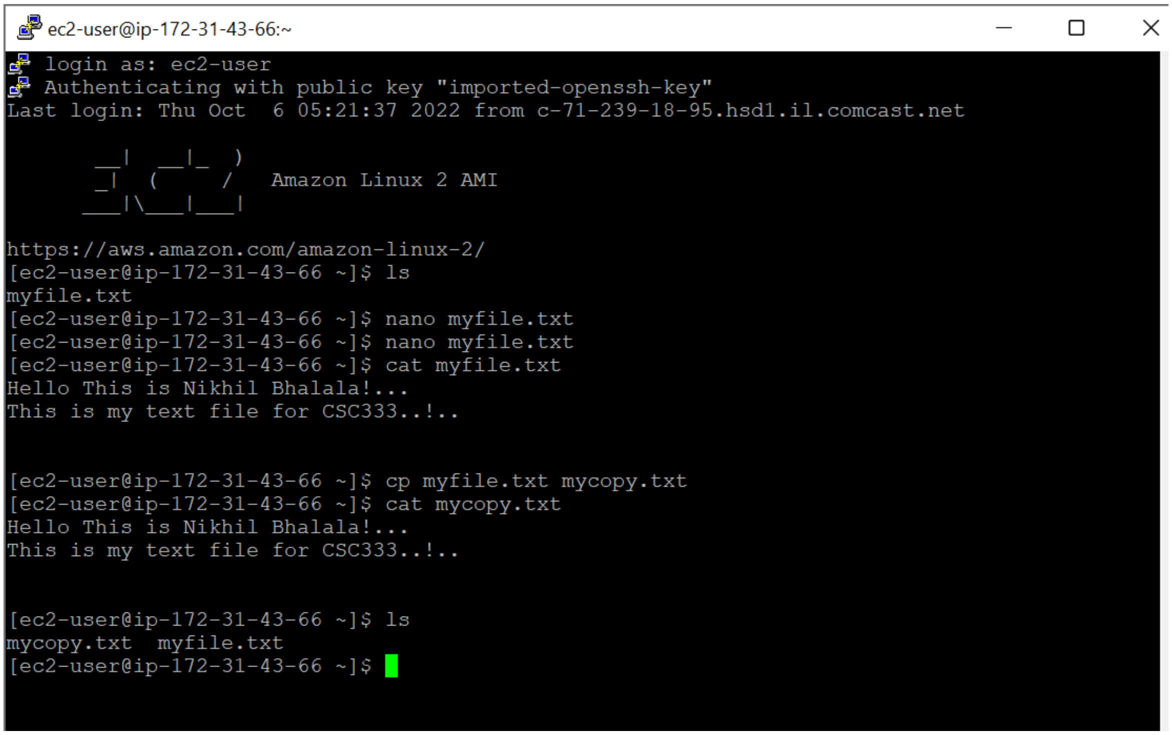
1. Create a text file.

Nikhil Bhalala
Dsc -333
Assignment-2



```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu Oct  6 05:21:37 2022 from c-71-239-18-95.hsd1.il.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-43-66 ~]$ ls
myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ cat myfile.txt
Hello This is Nikhil Bhalala!...
This is my text file for CSC333..!..


[ec2-user@ip-172-31-43-66 ~]$
```

2. Copy your file

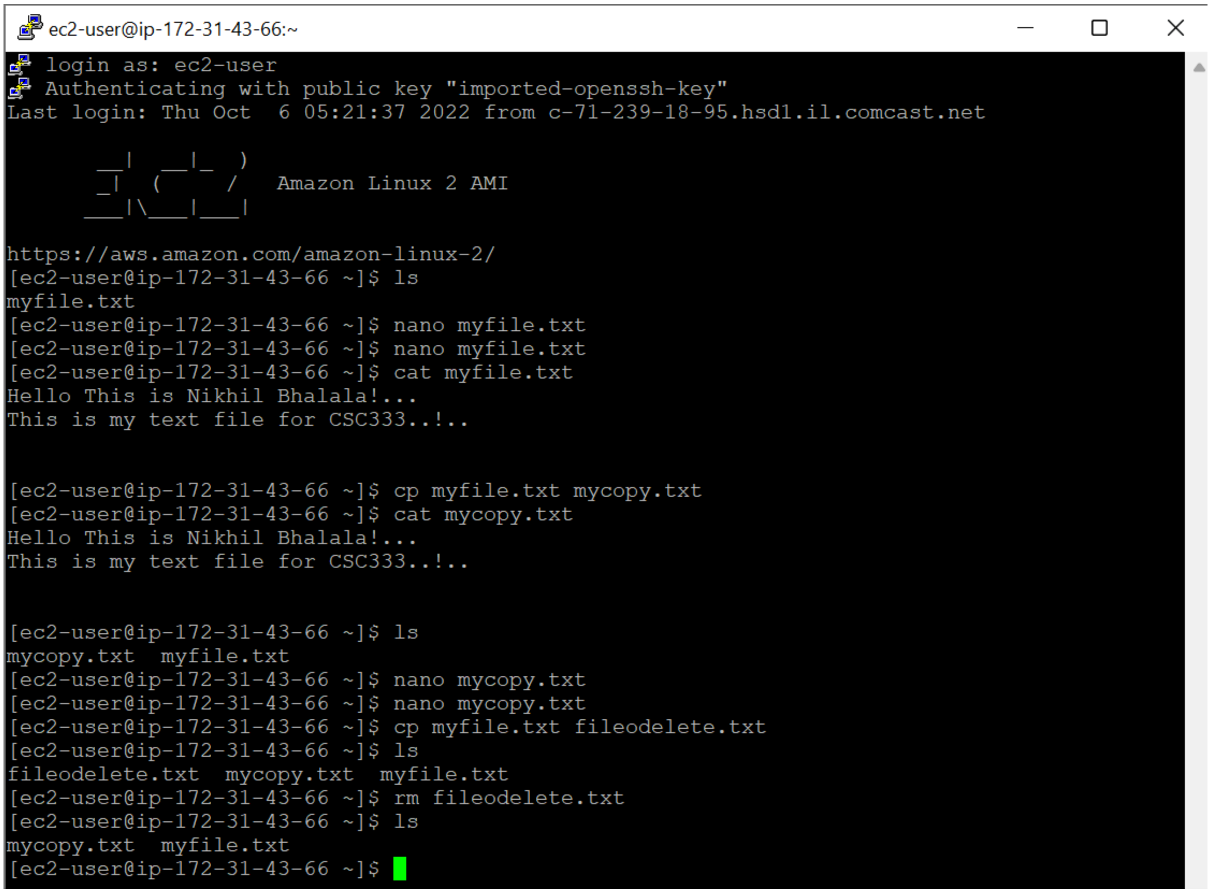


ec2-user@ip-172-31-43-66:~

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Thu Oct  6 05:21:37 2022 from c-71-239-18-95.hsd1.il.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-43-66 ~]$ ls
myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ cat myfile.txt
Hello This is Nikhil Bhalala!...
This is my text file for CSC333..!..


[ec2-user@ip-172-31-43-66 ~]$ cp myfile.txt mycopy.txt
[ec2-user@ip-172-31-43-66 ~]$ cat mycopy.txt
Hello This is Nikhil Bhalala!...
This is my text file for CSC333..!..


[ec2-user@ip-172-31-43-66 ~]$ ls
mycopy.txt  myfile.txt
[ec2-user@ip-172-31-43-66 ~]$
```

Nikhil Bhalala
Dsc -333
Assignment-2

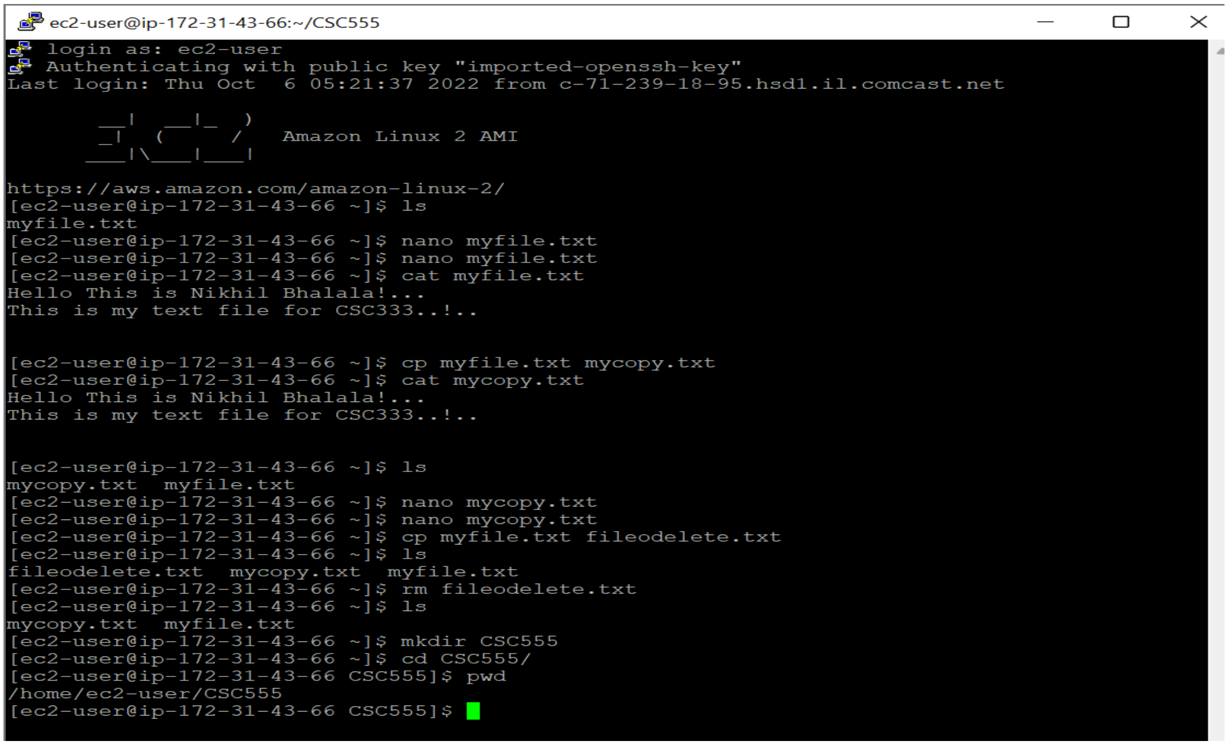
3. Delete a file



```
 login as: ec2-user
 Authenticating with public key "imported-openssh-key"
Last login: Thu Oct  6 05:21:37 2022 from c-71-239-18-95.hsd1.il.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-43-66 ~]$ ls
myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ cat myfile.txt
Hello This is Nikhil Bhalala!...
This is my text file for CSC333..!..


[ec2-user@ip-172-31-43-66 ~]$ cp myfile.txt mycopy.txt
[ec2-user@ip-172-31-43-66 ~]$ cat mycopy.txt
Hello This is Nikhil Bhalala!...
This is my text file for CSC333..!..


[ec2-user@ip-172-31-43-66 ~]$ ls
mycopy.txt  myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano mycopy.txt
[ec2-user@ip-172-31-43-66 ~]$ nano mycopy.txt
[ec2-user@ip-172-31-43-66 ~]$ cp myfile.txt fileodelete.txt
[ec2-user@ip-172-31-43-66 ~]$ ls
fileodelete.txt  mycopy.txt  myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ rm fileodelete.txt
[ec2-user@ip-172-31-43-66 ~]$ ls
mycopy.txt  myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ 
```

Nikhil Bhalala
Dsc -333
Assignment-2

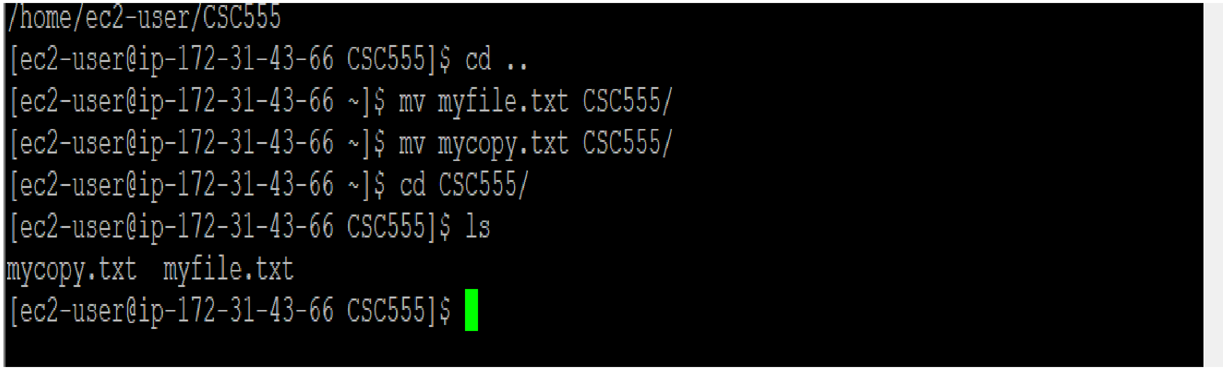
4. Create a directory to put your files.



```
 login as: ec2-user
 Authenticating with public key "imported-openssh-key"
Last login: Thu Oct  6 05:21:37 2022 from c-71-239-18-95.hsd1.il.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-43-66 ~]$ ls
myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ cat myfile.txt
Hello This is Nikhil Bhalala!...
This is my text file for CSC333..!..


[ec2-user@ip-172-31-43-66 ~]$ cp myfile.txt mycopy.txt
[ec2-user@ip-172-31-43-66 ~]$ cat mycopy.txt
Hello This is Nikhil Bhalala!...
This is my text file for CSC333..!..


[ec2-user@ip-172-31-43-66 ~]$ ls
mycopy.txt  myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ nano mycopy.txt
[ec2-user@ip-172-31-43-66 ~]$ nano mycopy.txt
[ec2-user@ip-172-31-43-66 ~]$ cp myfile.txt fileodelete.txt
[ec2-user@ip-172-31-43-66 ~]$ ls
fileodelete.txt  mycopy.txt  myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ rm fileodelete.txt
[ec2-user@ip-172-31-43-66 ~]$ ls
mycopy.txt  myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ mkdir CSC555
[ec2-user@ip-172-31-43-66 ~]$ cd CSC555/
[ec2-user@ip-172-31-43-66 CSC555]$ pwd
/home/ec2-user/CSC555
[ec2-user@ip-172-31-43-66 CSC555]$ 
```

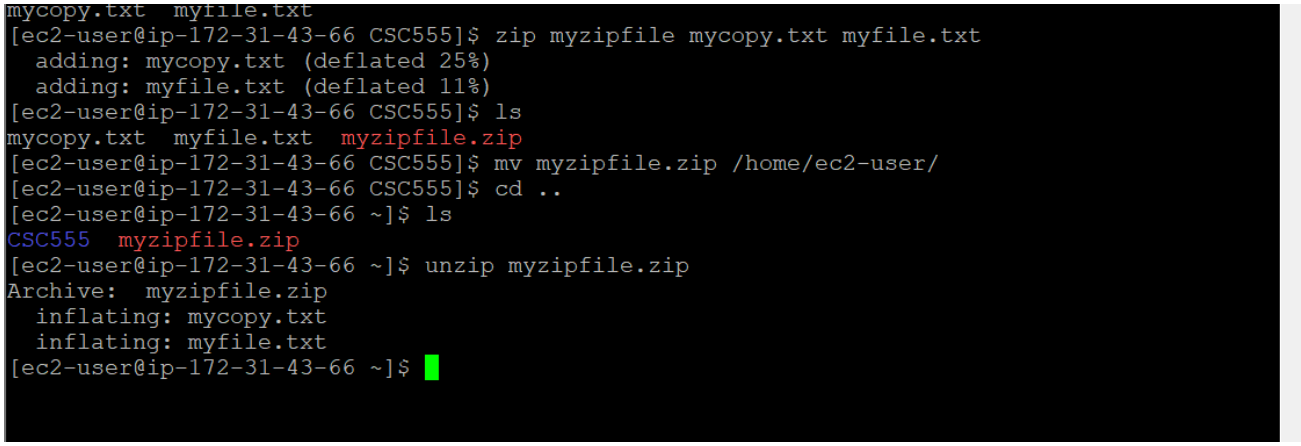
5. Move your files to your new directory.

Nikhil Bhalala
Dsc -333
Assignment-2



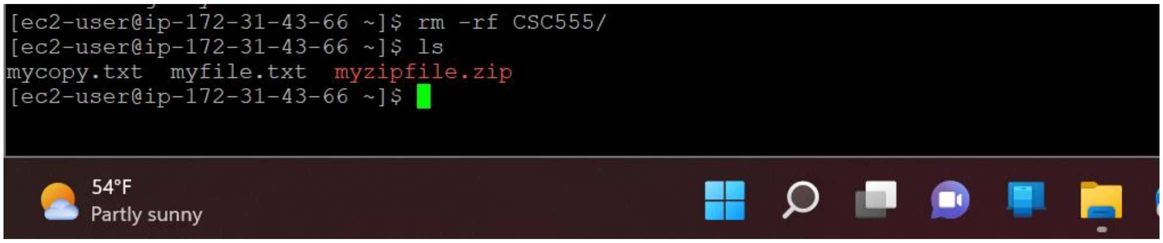
```
/home/ec2-user/CSC555
[ec2-user@ip-172-31-43-66 CSC555]$ cd ..
[ec2-user@ip-172-31-43-66 ~]$ mv myfile.txt CSC555/
[ec2-user@ip-172-31-43-66 ~]$ mv mycopy.txt CSC555/
[ec2-user@ip-172-31-43-66 ~]$ cd CSC555/
[ec2-user@ip-172-31-43-66 CSC555]$ ls
mycopy.txt  myfile.txt
[ec2-user@ip-172-31-43-66 CSC555]$
```

6. Zip and Unzip your files.



```
mycopy.txt  myfile.txt
[ec2-user@ip-172-31-43-66 CSC555]$ zip myzipfile mycopy.txt myfile.txt
  adding: mycopy.txt (deflated 25%)
  adding: myfile.txt (deflated 11%)
[ec2-user@ip-172-31-43-66 CSC555]$ ls
mycopy.txt  myfile.txt  myzipfile.zip
[ec2-user@ip-172-31-43-66 CSC555]$ mv myzipfile.zip /home/ec2-user/
[ec2-user@ip-172-31-43-66 CSC555]$ cd ..
[ec2-user@ip-172-31-43-66 ~]$ ls
CSC555  myzipfile.zip
[ec2-user@ip-172-31-43-66 ~]$ unzip myzipfile.zip
Archive:  myzipfile.zip
  inflating: mycopy.txt
  inflating: myfile.txt
[ec2-user@ip-172-31-43-66 ~]$
```

7. Remove your CSC555 directory.



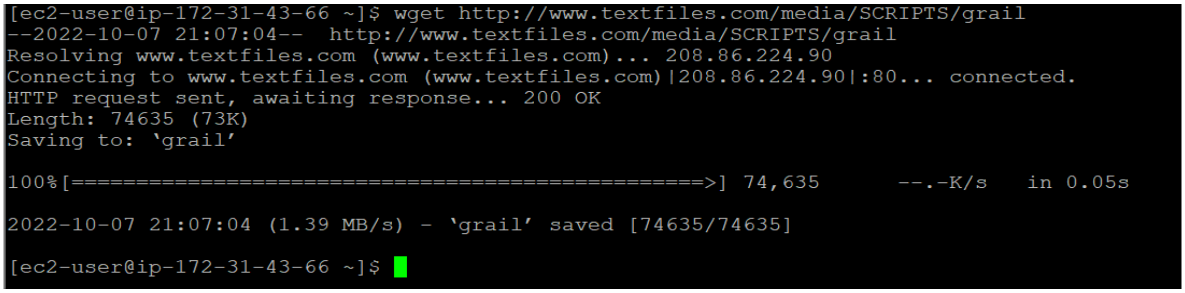
```
[ec2-user@ip-172-31-43-66 ~]$ rm -rf CSC555/
[ec2-user@ip-172-31-43-66 ~]$ ls
mycopy.txt  myfile.txt  myzipfile.zip
[ec2-user@ip-172-31-43-66 ~]$
```

Nikhil Bhalala

Dsc -333

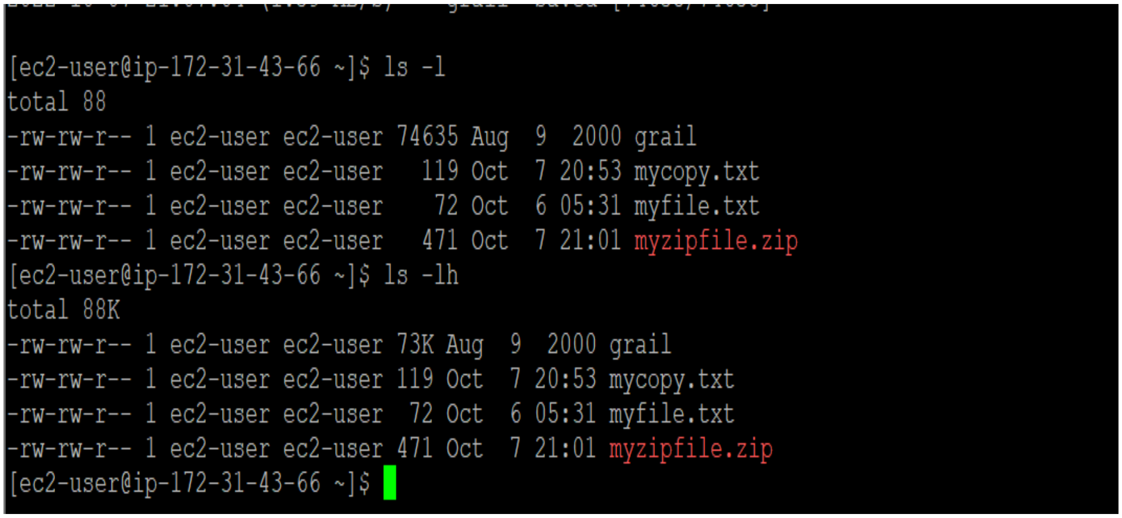
Assignment-2

8. Download a file from the web.



```
[ec2-user@ip-172-31-43-66 ~]$ wget http://www.textfiles.com/media/SCRIPTS/grail
--2022-10-07 21:07:04--  http://www.textfiles.com/media/SCRIPTS/grail
Resolving www.textfiles.com (www.textfiles.com)... 208.86.224.90
Connecting to www.textfiles.com (www.textfiles.com)|208.86.224.90|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 74635 (73K)
Saving to: ‘grail’

100%[===============================================>] 74,635      --.-K/s   in 0.05s

2022-10-07 21:07:04 (1.39 MB/s) - ‘grail’ saved [74635/74635]

[ec2-user@ip-172-31-43-66 ~]$
```

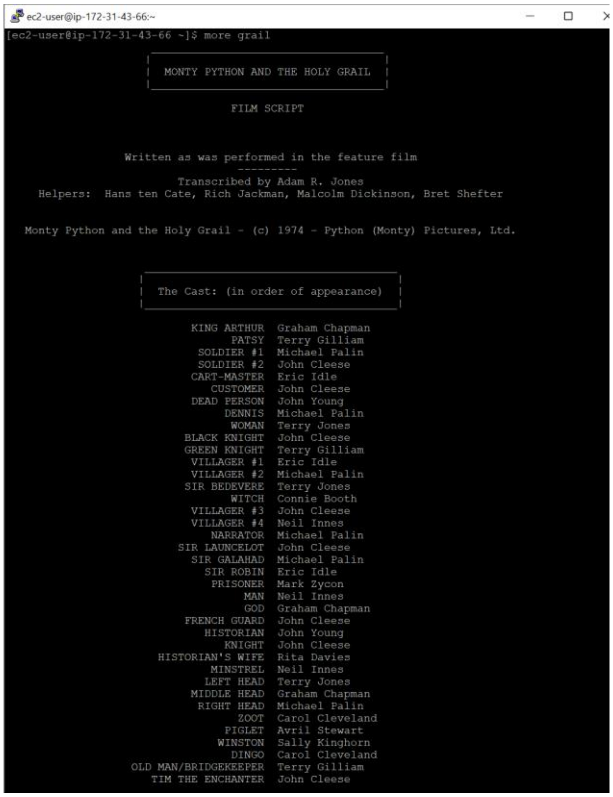
9. ls formats



```
2022-10-07 21:07:04 (1.39 MB/s) - ‘grail’ saved [74635/74635]

[ec2-user@ip-172-31-43-66 ~]$ ls -l
total 88
-rw-rw-r-- 1 ec2-user ec2-user 74635 Aug  9  2000 grail
-rw-rw-r-- 1 ec2-user ec2-user   119 Oct  7 20:53 mycopy.txt
-rw-rw-r-- 1 ec2-user ec2-user    72 Oct  6 05:31 myfile.txt
-rw-rw-r-- 1 ec2-user ec2-user   471 Oct  7 21:01 myzipfile.zip
[ec2-user@ip-172-31-43-66 ~]$ ls -lh
total 88K
-rw-rw-r-- 1 ec2-user ec2-user 73K Aug  9  2000 grail
-rw-rw-r-- 1 ec2-user ec2-user 119 Oct  7 20:53 mycopy.txt
-rw-rw-r-- 1 ec2-user ec2-user  72 Oct  6 05:31 myfile.txt
-rw-rw-r-- 1 ec2-user ec2-user 471 Oct  7 21:01 myzipfile.zip
[ec2-user@ip-172-31-43-66 ~]$
```

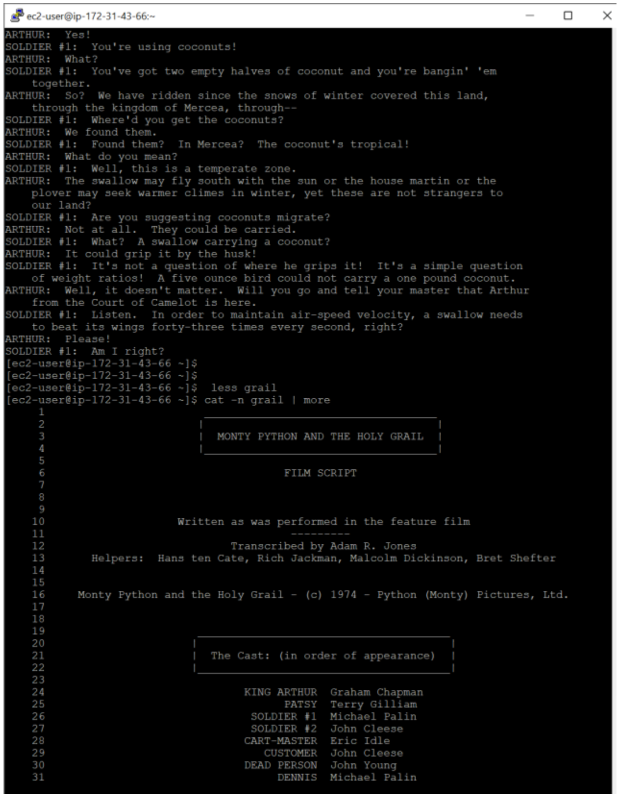
Nikhil Bhalala
Dsc -333
Assignment-2

10. More on viewing files.

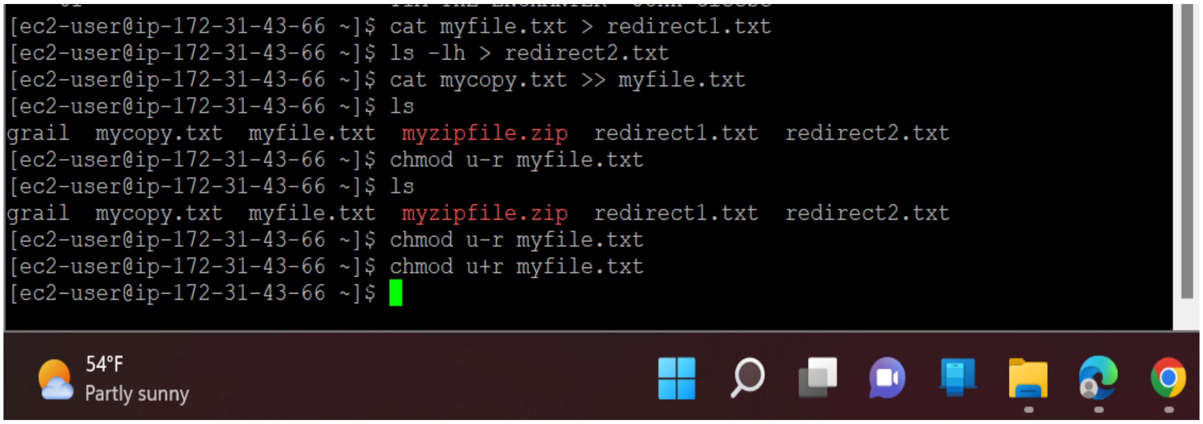


```
[ec2-user@ip-172-31-43-66 ~]$ more grail
                      ___________________________________
                     |                                   |
                     |  MONTY PYTHON AND THE HOLY GRAIL  |
                     |___________________________________|

                                  FILM SCRIPT



                  Written as was performed in the feature film
                                   ---------
                          Transcribed by Adam R. Jones
     Helpers:  Hans ten Cate, Rich Jackman, Malcolm Dickinson, Bret Shefter


   Monty Python and the Holy Grail - (c) 1974 - Python (Monty) Pictures, Ltd.


                     ______________________________________
                    |                                      |
                    |  The Cast: (in order of appearance)  |
                    |______________________________________|

                           KING ARTHUR  Graham Chapman
                                 PATSY  Terry Gilliam
                            SOLDIER #1  Michael Palin
                            SOLDIER #2  John Cleese
                           CART-MASTER  Eric Idle
                              CUSTOMER  John Cleese
                           DEAD PERSON  John Young
                                DENNIS  Michael Palin
                                 WOMAN  Terry Jones
                          BLACK KNIGHT  John Cleese
                          GREEN KNIGHT  Terry Gilliam
                           VILLAGER #1  Eric Idle
                           VILLAGER #2  Michael Palin
                          SIR BEDEVERE  Terry Jones
                                 WITCH  Connie Booth
                           VILLAGER #3  John Cleese
                           VILLAGER #4  Neil Innes
                              NARRATOR  Michael Palin
                         SIR LAUNCELOT  John Cleese
                           SIR GALAHAD  Michael Palin
                             SIR ROBIN  Eric Idle
                              PRISONER  Mark Zycon
                                   MAN  Neil Innes
                                   GOD  Graham Chapman
                          FRENCH GUARD  John Cleese
                             HISTORIAN  John Young
                                KNIGHT  John Cleese
                      HISTORIAN'S WIFE  Rita Davies
                              MINSTREL  Neil Innes
                             LEFT HEAD  Terry Jones
                           MIDDLE HEAD  Graham Chapman
                            RIGHT HEAD  Michael Palin
                                  ZOOT  Carol Cleveland
                                PIGLET  Avril Stewart
                               WINSTON  Sally Kinghorn
                                 DINGO  Carol Cleveland
                  OLD MAN/BRIDGEKEEPER  Terry Gilliam
                     TIM THE ENCHANTER  John Cleese
```

Nikhil Bhalala
Dsc -333
Assignment-2



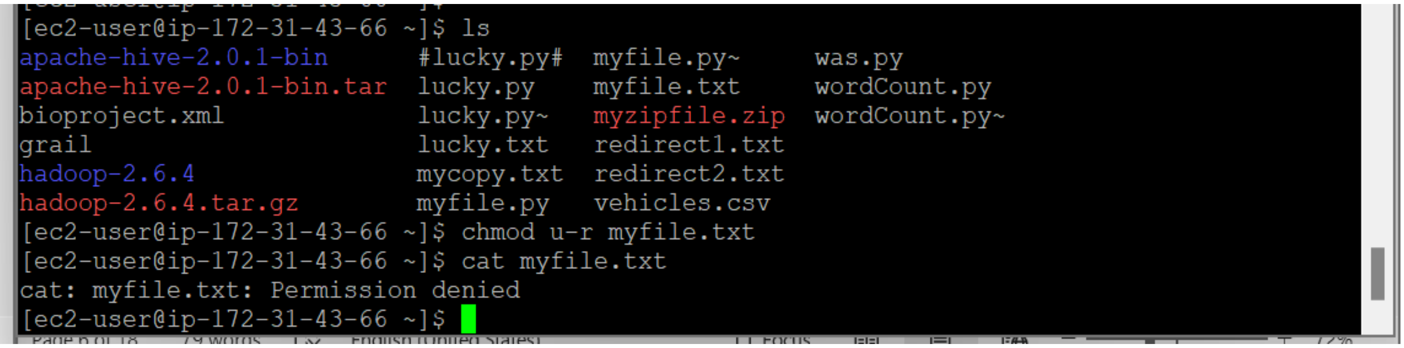
```
ARTHUR:  Yes!
SOLDIER #1:  You're using coconuts!
ARTHUR:  What?
SOLDIER #1:  You've got two empty halves of coconut and you're bangin' 'em
    together.
ARTHUR:  So?  We have ridden since the snows of winter covered this land,
    through the kingdom of Mercea, through--
SOLDIER #1:  Where'd you get the coconuts?
ARTHUR:  We found them.
SOLDIER #1:  Found them?  In Mercea?  The coconut's tropical!
ARTHUR:  What do you mean?
SOLDIER #1:  Well, this is a temperate zone.
ARTHUR:  The swallow may fly south with the sun or the house martin or the
    plover may seek warmer climes in winter, yet these are not strangers to
    our land?
SOLDIER #1:  Are you suggesting coconuts migrate?
ARTHUR:  Not at all.  They could be carried.
SOLDIER #1:  What?  A swallow carrying a coconut?
ARTHUR:  It could grip it by the husk!
SOLDIER #1:  It's not a question of where he grips it!  It's a simple question
    of weight ratios!  A five ounce bird could not carry a one pound coconut.
ARTHUR:  Well, it doesn't matter.  Will you go and tell your master that Arthur
    from the Court of Camelot is here.
SOLDIER #1:  Listen.  In order to maintain air-speed velocity, a swallow needs
    to beat its wings forty-three times every second, right?
ARTHUR:  Please!
SOLDIER #1:  Am I right?
[ec2-user@ip-172-31-43-66 ~]$
[ec2-user@ip-172-31-43-66 ~]$
[ec2-user@ip-172-31-43-66 ~]$  less grail
[ec2-user@ip-172-31-43-66 ~]$ cat -n grail | more
     1                           ___________________________________
     2                          |                                   |
     3                          |  MONTY PYTHON AND THE HOLY GRAIL  |
     4                          |___________________________________|
     5
     6                                      FILM SCRIPT
     7
     8
     9
    10                      Written as was performed in the feature film
    11                                       ----------
    12                               Transcribed by Adam R. Jones
    13        Helpers:  Hans ten Cate, Rich Jackman, Malcolm Dickinson, Bret Shefter
    14
    15
    16      Monty Python and the Holy Grail - (c) 1974 - Python (Monty) Pictures, Ltd.
    17
    18
    19
    20                         _______________________________________
    21                        |                                       |
    22                        |  The Cast: (in order of appearance)   |
    23                        |_______________________________________|
    24                              KING ARTHUR  Graham Chapman
    25                                    PATSY  Terry Gilliam
    26                               SOLDIER #1  Michael Palin
    27                               SOLDIER #2  John Cleese
    28                              CART-MASTER  Eric Idle
    29                                 CUSTOMER  John Cleese
    30                              DEAD PERSON  John Young
    31                                   DENNIS  Michael Palin
```



```
[ec2-user@ip-172-31-43-66 ~]$ cat myfile.txt > redirect1.txt
[ec2-user@ip-172-31-43-66 ~]$ ls -lh > redirect2.txt
[ec2-user@ip-172-31-43-66 ~]$ cat mycopy.txt >> myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ ls
grail  mycopy.txt  myfile.txt  myzipfile.zip  redirect1.txt  redirect2.txt
[ec2-user@ip-172-31-43-66 ~]$ chmod u-r myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ ls
grail  mycopy.txt  myfile.txt  myzipfile.zip  redirect1.txt  redirect2.txt
[ec2-user@ip-172-31-43-66 ~]$ chmod u-r myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ chmod u+r myfile.txt
[ec2-user@ip-172-31-43-66 ~]$
```

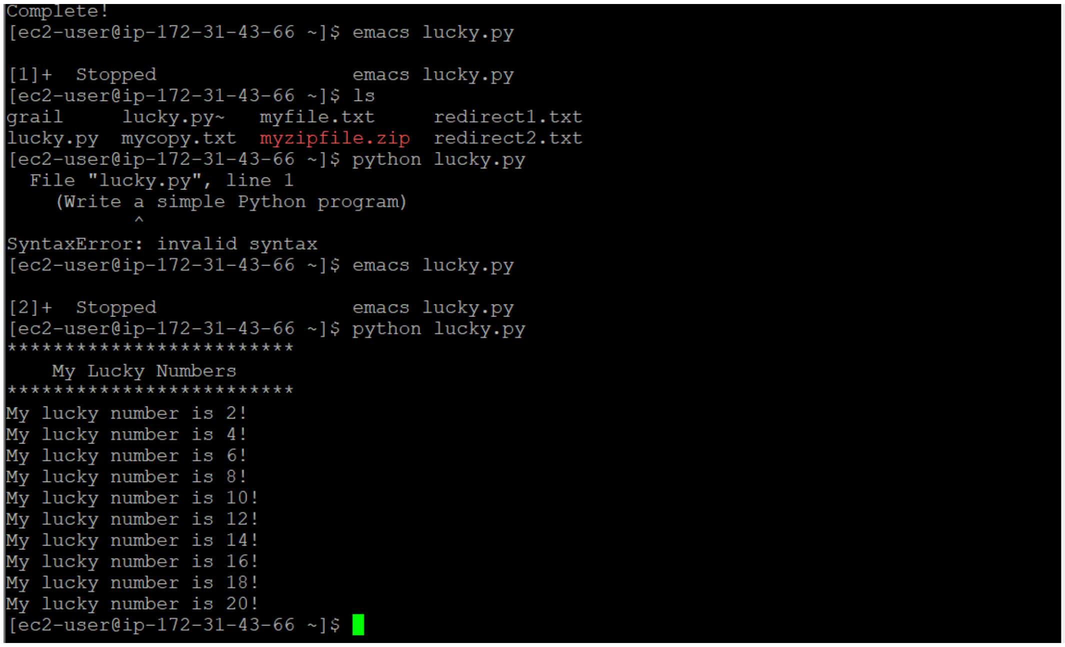
Nikhil Bhalala
Dsc -333
Assignment-2

11. Change access permissions to objects with the *change mode* command.



```
[ec2-user@ip-172-31-43-66 ~]$ ls
apache-hive-2.0.1-bin       #lucky.py#  myfile.py~     was.py
apache-hive-2.0.1-bin.tar   lucky.py    myfile.txt     wordCount.py
bioproject.xml              lucky.py~   myzipfile.zip  wordCount.py~
grail                       lucky.txt   redirect1.txt
hadoop-2.6.4                mycopy.txt  redirect2.txt
hadoop-2.6.4.tar.gz         myfile.py   vehicles.csv
[ec2-user@ip-172-31-43-66 ~]$ chmod u-r myfile.txt
[ec2-user@ip-172-31-43-66 ~]$ cat myfile.txt
cat: myfile.txt: Permission denied
[ec2-user@ip-172-31-43-66 ~]$
```

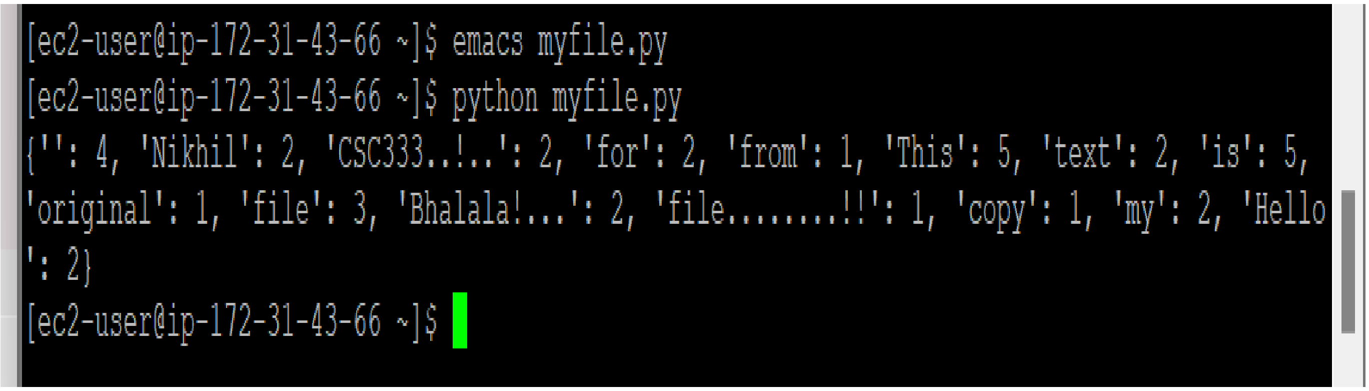
12. Python examples



```
Complete!
[ec2-user@ip-172-31-43-66 ~]$ emacs lucky.py

[1]+  Stopped                 emacs lucky.py
[ec2-user@ip-172-31-43-66 ~]$ ls
grail     lucky.py~  myfile.txt     redirect1.txt
lucky.py  mycopy.txt  myzipfile.zip  redirect2.txt
[ec2-user@ip-172-31-43-66 ~]$ python lucky.py
  File "lucky.py", line 1
    (Write a simple Python program)
           ^
SyntaxError: invalid syntax
[ec2-user@ip-172-31-43-66 ~]$ emacs lucky.py

[2]+  Stopped                 emacs lucky.py
[ec2-user@ip-172-31-43-66 ~]$ python lucky.py
*************************
    My Lucky Numbers
*************************
My lucky number is 2!
My lucky number is 4!
My lucky number is 6!
My lucky number is 8!
My lucky number is 10!
My lucky number is 12!
My lucky number is 14!
My lucky number is 16!
My lucky number is 18!
My lucky number is 20!
[ec2-user@ip-172-31-43-66 ~]$
```

Nikhil Bhalala
Dsc -333
Assignment-2

```
with open("myfile.txt", 'r') as file_as:
    f =file_as.read()
    f =f.replace('\n',' ')
    read_line=file_as.readlines()


    line=f.split(" ")
    line =line[:-1]
    d={}
    for x in line:
        d[x]=line.count(x)
    print(d)
   # print(len(d.keys()))
```

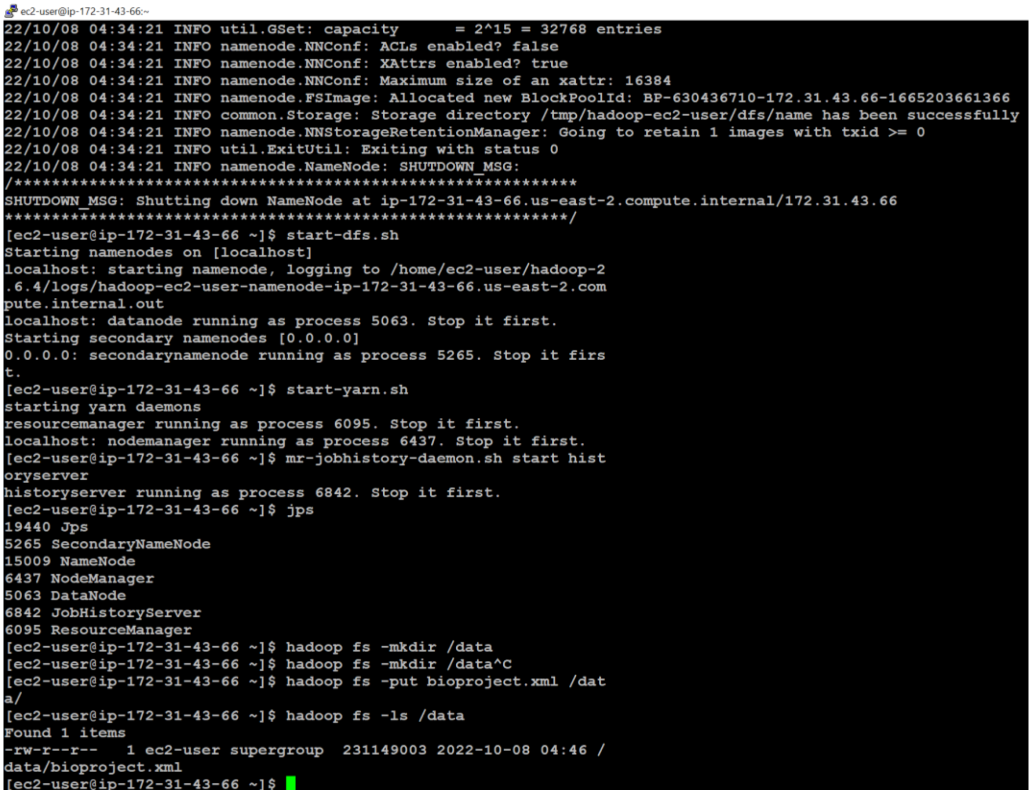


```
[ec2-user@ip-172-31-43-66 ~]$ emacs myfile.py
[ec2-user@ip-172-31-43-66 ~]$ python myfile.py
{'': 4, 'Nikhil': 2, 'CSC333..!..': 2, 'for': 2, 'from': 1, 'This': 5, 'text': 2, 'is': 5,
'original': 1, 'file': 3, 'Bhalala!...': 2, 'file........!!': 1, 'copy': 1, 'my': 2, 'Hello
': 2}
[ec2-user@ip-172-31-43-66 ~]$
```

Nikhil Bhalala
Dsc -333
Assignment-2

Part 3: Wordcount

⇨ hadoop fs -put bioproject.xml /data/

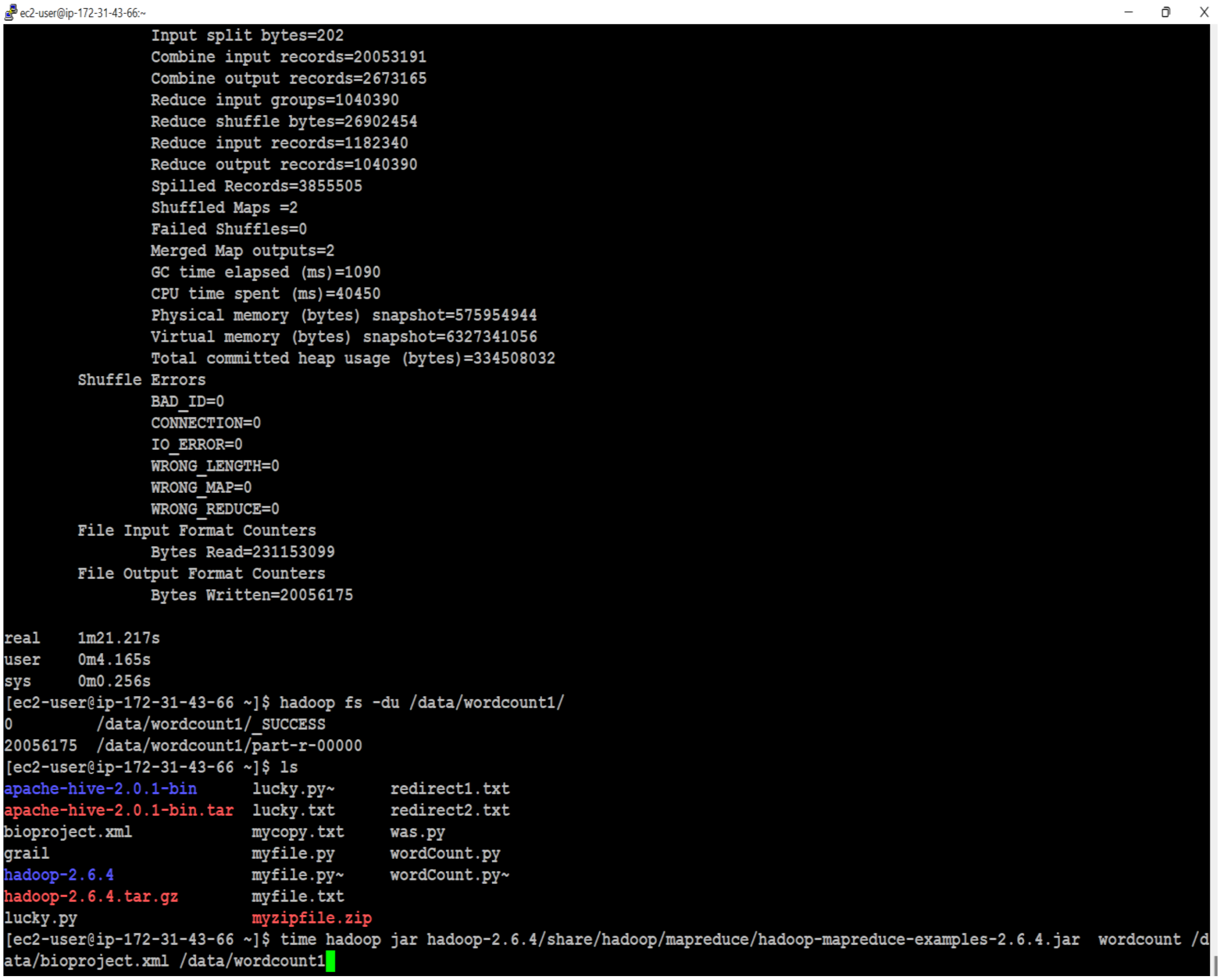


ec2-user@ip-172-31-43-66:~

```
22/10/08 04:34:21 INFO util.GSet: capacity      = 2^15 = 32768 entries
22/10/08 04:34:21 INFO namenode.NNConf: ACLs enabled? false
22/10/08 04:34:21 INFO namenode.NNConf: XAttrs enabled? true
22/10/08 04:34:21 INFO namenode.NNConf: Maximum size of an xattr: 16384
22/10/08 04:34:21 INFO namenode.FSImage: Allocated new BlockPoolId: BP-630436710-172.31.43.66-1665203661366
22/10/08 04:34:21 INFO common.Storage: Storage directory /tmp/hadoop-ec2-user/dfs/name has been successfully
22/10/08 04:34:21 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
22/10/08 04:34:21 INFO util.ExitUtil: Exiting with status 0
22/10/08 04:34:21 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at ip-172-31-43-66.us-east-2.compute.internal/172.31.43.66
************************************************************/
[ec2-user@ip-172-31-43-66 ~]$ start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/ec2-user/hadoop-2
.6.4/logs/hadoop-ec2-user-namenode-ip-172-31-43-66.us-east-2.com
pute.internal.out
localhost: datanode running as process 5063. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 5265. Stop it firs
t.
[ec2-user@ip-172-31-43-66 ~]$ start-yarn.sh
starting yarn daemons
resourcemanager running as process 6095. Stop it first.
localhost: nodemanager running as process 6437. Stop it first.
[ec2-user@ip-172-31-43-66 ~]$ mr-jobhistory-daemon.sh start hist
oryserver
historyserver running as process 6842. Stop it first.
[ec2-user@ip-172-31-43-66 ~]$ jps
19440 Jps
5265 SecondaryNameNode
15009 NameNode
6437 NodeManager
5063 DataNode
6842 JobHistoryServer
6095 ResourceManager
[ec2-user@ip-172-31-43-66 ~]$ hadoop fs -mkdir /data
[ec2-user@ip-172-31-43-66 ~]$ hadoop fs -mkdir /data^C
[ec2-user@ip-172-31-43-66 ~]$ hadoop fs -put bioproject.xml /dat
a/
[ec2-user@ip-172-31-43-66 ~]$ hadoop fs -ls /data
Found 1 items
-rw-r--r--   1 ec2-user supergroup  231149003 2022-10-08 04:46 /
data/bioproject.xml
[ec2-user@ip-172-31-43-66 ~]$ 
```

Nikhil Bhalala
Dsc -333
Assignment-2

- ⇨ time hadoop jar hadoop-2.6.4/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.4.jar wordcount /data/bioproject.xml /data/wordcount1



```
                Input split bytes=202
                Combine input records=20053191
                Combine output records=2673165
                Reduce input groups=1040390
                Reduce shuffle bytes=26902454
                Reduce input records=1182340
                Reduce output records=1040390
                Spilled Records=3855505
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=1090
                CPU time spent (ms)=40450
                Physical memory (bytes) snapshot=575954944
                Virtual memory (bytes) snapshot=6327341056
                Total committed heap usage (bytes)=334508032
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=231153099
        File Output Format Counters
                Bytes Written=20056175

real    1m21.217s
user    0m4.165s
sys     0m0.256s
[ec2-user@ip-172-31-43-66 ~]$ hadoop fs -du /data/wordcount1/
0         /data/wordcount1/_SUCCESS
20056175  /data/wordcount1/part-r-00000
[ec2-user@ip-172-31-43-66 ~]$ ls
apache-hive-2.0.1-bin      lucky.py~      redirect1.txt
apache-hive-2.0.1-bin.tar  lucky.txt      redirect2.txt
bioproject.xml             mycopy.txt     was.py
grail                      myfile.py      wordCount.py
hadoop-2.6.4               myfile.py~     wordCount.py~
hadoop-2.6.4.tar.gz        myfile.txt
lucky.py                   myzipfile.zip
[ec2-user@ip-172-31-43-66 ~]$ time hadoop jar hadoop-2.6.4/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.4.jar  wordcount /d
ata/bioproject.xml /data/wordcount1
```

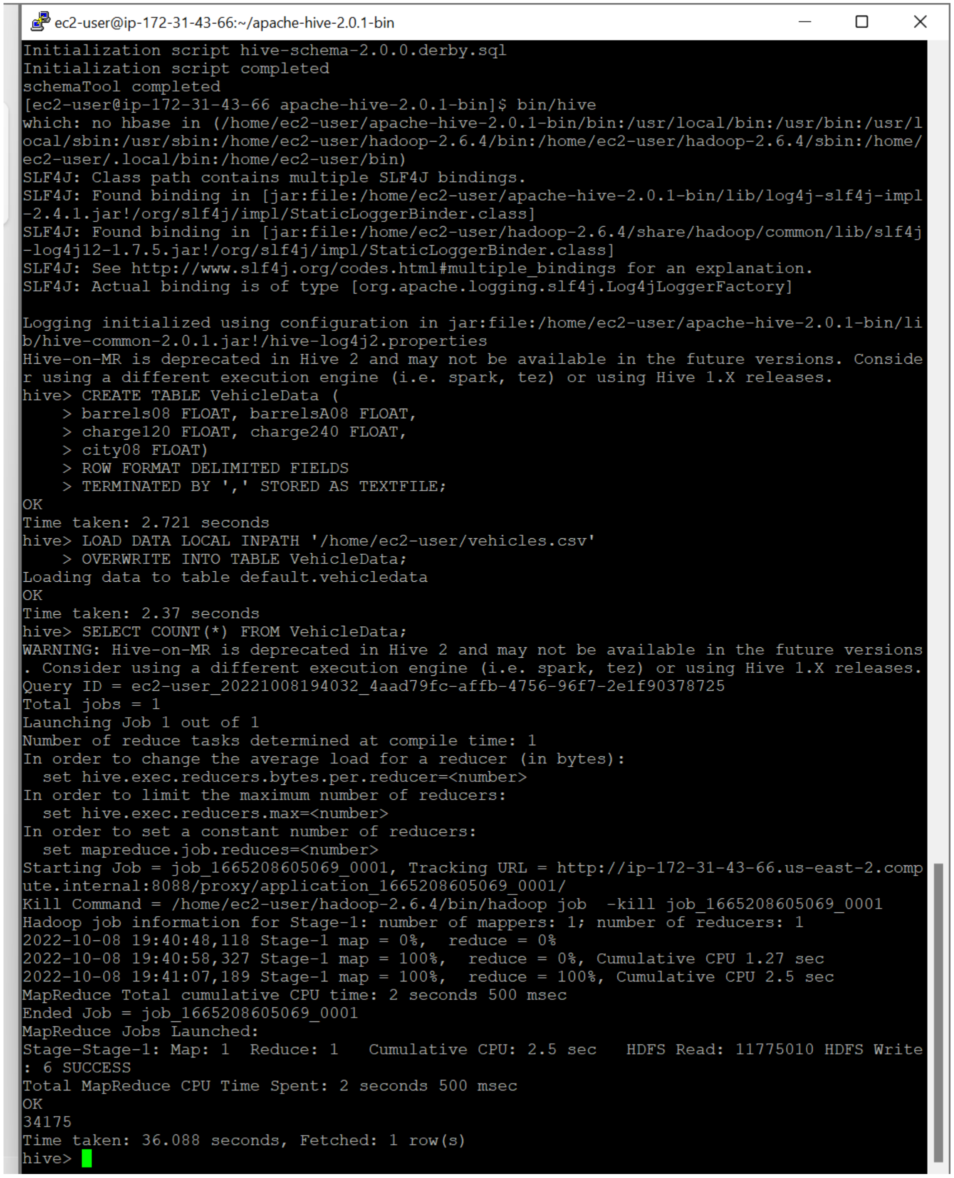
Nikhil Bhalala
Dsc -333
Assignment-2

Part 4: Hive Intro

1. **Verify that your table had successfully loaded by running**
   SELECT COUNT(*) FROM VehicleData;
   (Copy the query output and report how many rows you got as an answer.)

⇨ 34175 rows

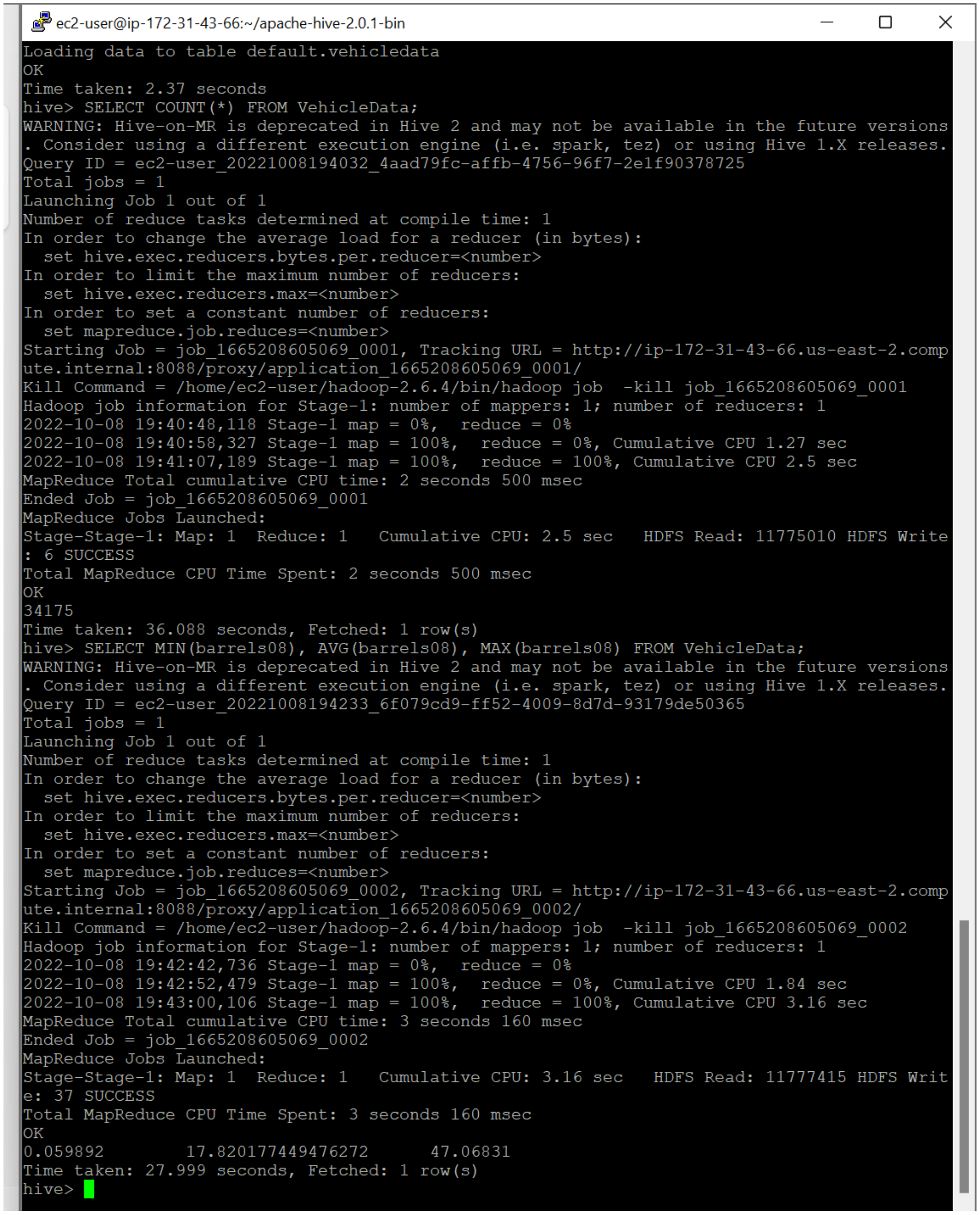
Nikhil Bhalala
Dsc -333
Assignment-2



```
Initialization script hive-schema-2.0.0.derby.sql
Initialization script completed
schemaTool completed
[ec2-user@ip-172-31-43-66 apache-hive-2.0.1-bin]$ bin/hive
which: no hbase in (/home/ec2-user/apache-hive-2.0.1-bin/bin:/usr/local/bin:/usr/bin:/usr/l
ocal/sbin:/usr/sbin:/home/ec2-user/hadoop-2.6.4/bin:/home/ec2-user/hadoop-2.6.4/sbin:/home/
ec2-user/.local/bin:/home/ec2-user/bin)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/ec2-user/apache-hive-2.0.1-bin/lib/log4j-slf4j-impl
-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/ec2-user/hadoop-2.6.4/share/hadoop/common/lib/slf4j
-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/ec2-user/apache-hive-2.0.1-bin/li
b/hive-common-2.0.1.jar!/hive-log4j2.properties
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Conside
r using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> CREATE TABLE VehicleData (
    > barrels08 FLOAT, barrelsA08 FLOAT,
    > charge120 FLOAT, charge240 FLOAT,
    > city08 FLOAT)
    > ROW FORMAT DELIMITED FIELDS
    > TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 2.721 seconds
hive> LOAD DATA LOCAL INPATH '/home/ec2-user/vehicles.csv'
    > OVERWRITE INTO TABLE VehicleData;
Loading data to table default.vehicledata
OK
Time taken: 2.37 seconds
hive> SELECT COUNT(*) FROM VehicleData;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions
. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = ec2-user_20221008194032_4aad79fc-affb-4756-96f7-2e1f90378725
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1665208605069_0001, Tracking URL = http://ip-172-31-43-66.us-east-2.comp
ute.internal:8088/proxy/application_1665208605069_0001/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1665208605069_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-10-08 19:40:48,118 Stage-1 map = 0%,  reduce = 0%
2022-10-08 19:40:58,327 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.27 sec
2022-10-08 19:41:07,189 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.5 sec
MapReduce Total cumulative CPU time: 2 seconds 500 msec
Ended Job = job_1665208605069_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.5 sec   HDFS Read: 11775010 HDFS Write
: 6 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 500 msec
OK
34175
Time taken: 36.088 seconds, Fetched: 1 row(s)
hive> 
```

2. **SELECT MIN(barrels08), AVG(barrels08), MAX(barrels08) FROM VehicleData;**
   (copy the output from that query)

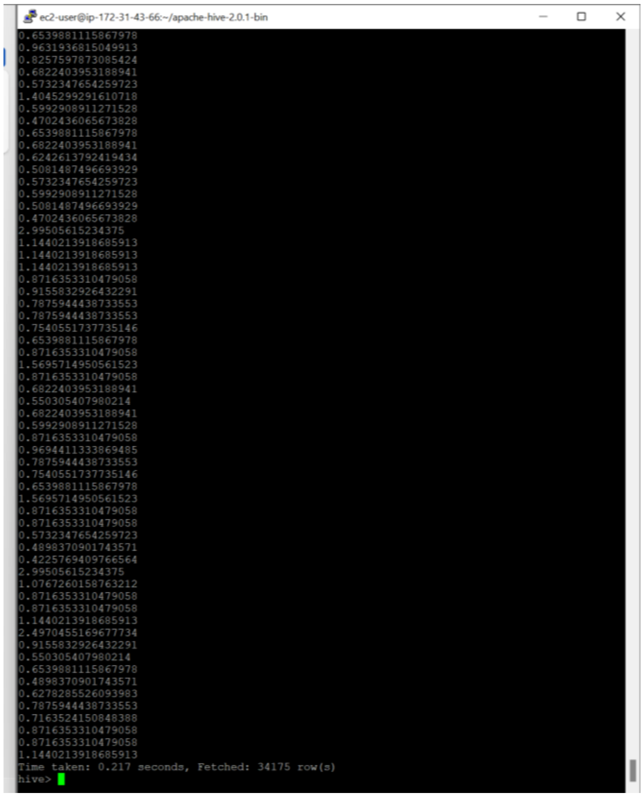
Nikhil Bhalala
Dsc -333
Assignment-2



```
Loading data to table default.vehicledata
OK
Time taken: 2.37 seconds
hive> SELECT COUNT(*) FROM VehicleData;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions
. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = ec2-user_20221008194032_4aad79fc-affb-4756-96f7-2e1f90378725
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1665208605069_0001, Tracking URL = http://ip-172-31-43-66.us-east-2.comp
ute.internal:8088/proxy/application_1665208605069_0001/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1665208605069_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-10-08 19:40:48,118 Stage-1 map = 0%,  reduce = 0%
2022-10-08 19:40:58,327 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.27 sec
2022-10-08 19:41:07,189 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.5 sec
MapReduce Total cumulative CPU time: 2 seconds 500 msec
Ended Job = job_1665208605069_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.5 sec   HDFS Read: 11775010 HDFS Write
: 6 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 500 msec
OK
34175
Time taken: 36.088 seconds, Fetched: 1 row(s)
hive> SELECT MIN(barrels08), AVG(barrels08), MAX(barrels08) FROM VehicleData;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions
. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = ec2-user_20221008194233_6f079cd9-ff52-4009-8d7d-93179de50365
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1665208605069_0002, Tracking URL = http://ip-172-31-43-66.us-east-2.comp
ute.internal:8088/proxy/application_1665208605069_0002/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1665208605069_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-10-08 19:42:42,736 Stage-1 map = 0%,  reduce = 0%
2022-10-08 19:42:52,479 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.84 sec
2022-10-08 19:43:00,106 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.16 sec
MapReduce Total cumulative CPU time: 3 seconds 160 msec
Ended Job = job_1665208605069_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.16 sec   HDFS Read: 11777415 HDFS Writ
e: 37 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 160 msec
OK
0.059892        17.820177449476272      47.06831
Time taken: 27.999 seconds, Fetched: 1 row(s)
hive>
```

Nikhil Bhalala
Dsc -333
Assignment-2

3. **SELECT (barrels08/city08) FROM VehicleData;**
   (you do not need to report the output from that query, but report "Time taken")

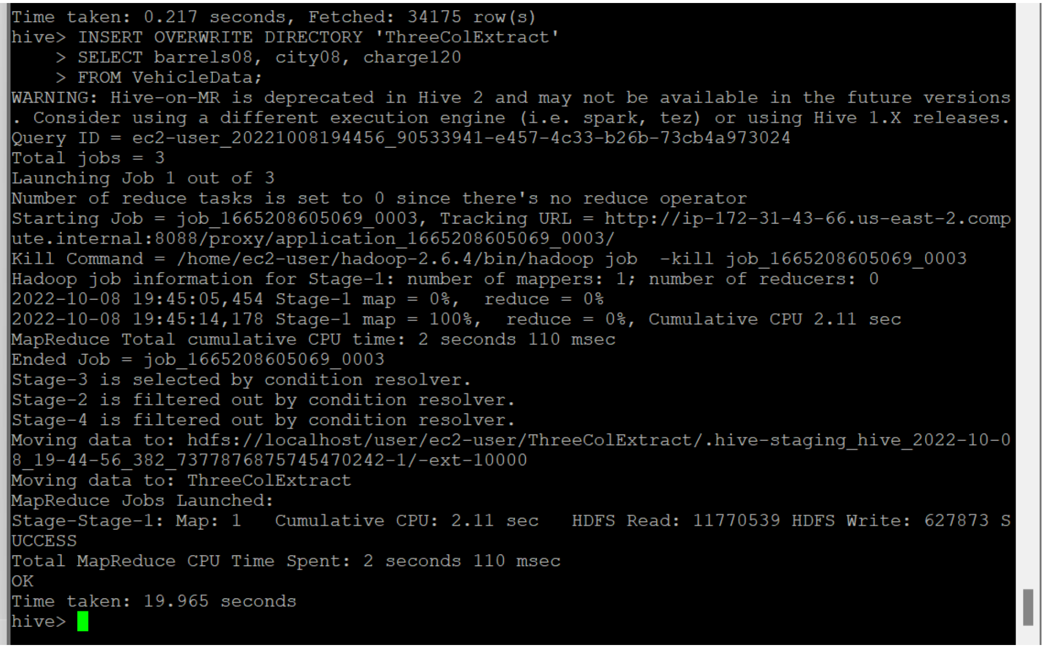


ec2-user@ip-172-31-43-66:~/apache-hive-2.0.1-bin

```
0.6539881115867978
0.9631936815049913
0.8257597873085424
0.6822403953188941
0.5732347654259723
1.4045299291610718
0.5992908911271528
0.4702436065673828
0.6539881115867978
0.6822403953188941
0.6242613792419434
0.5081487496693929
0.5732347654259723
0.5992908911271528
0.5081487496693929
0.4702436065673828
2.99505615234375
1.1440213918685913
1.1440213918685913
1.1440213918685913
0.8716353310479058
0.9155832926432291
0.7875944438733553
0.7875944438733553
0.7540551737735146
0.6539881115867978
0.8716353310479058
1.5695714950561523
0.8716353310479058
0.6822403953188941
0.550305407980214
0.6822403953188941
0.5992908911271528
0.8716353310479058
0.9694411333869485
0.7875944438733553
0.7540551737735146
0.6539881115867978
1.5695714950561523
0.8716353310479058
0.8716353310479058
0.5732347654259723
0.4898370901743571
0.4225769409766564
2.99505615234375
1.0767260158763212
0.8716353310479058
0.8716353310479058
1.1440213918685913
2.4970455169677734
0.9155832926432291
0.550305407980214
0.6539881115867978
0.4898370901743571
0.6278285526093983
0.7875944438733553
0.7163524150848388
0.8716353310479058
0.8716353310479058
1.1440213918685913
Time taken: 0.217 seconds, Fetched: 34175 row(s)
hive>
```

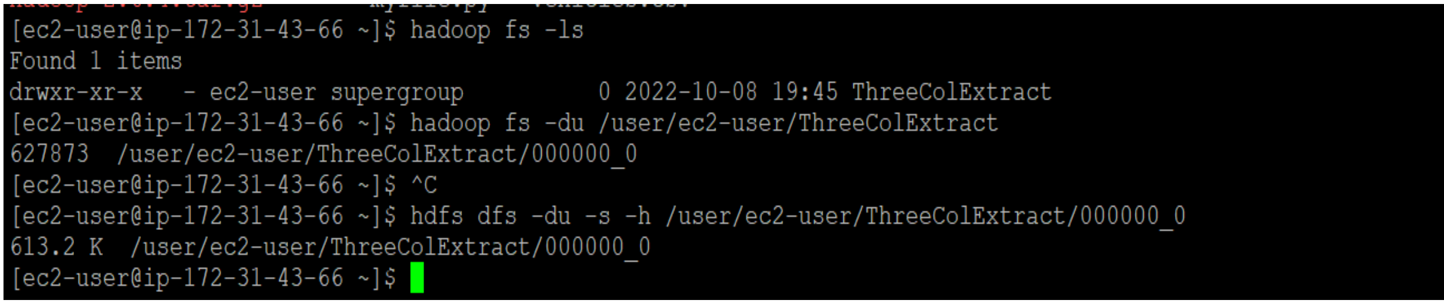
4. **INSERT OVERWRITE DIRECTORY 'ThreeColExtract'**
   SELECT barrels08, city08, charge120
   FROM VehicleData;

Nikhil Bhalala
Dsc -333
Assignment-2



```
Time taken: 0.217 seconds, Fetched: 34175 row(s)
hive> INSERT OVERWRITE DIRECTORY 'ThreeColExtract'
    > SELECT barrels08, city08, charge120
    > FROM VehicleData;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions
. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = ec2-user_20221008194456_90533941-e457-4c33-b26b-73cb4a973024
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1665208605069_0003, Tracking URL = http://ip-172-31-43-66.us-east-2.comp
ute.internal:8088/proxy/application_1665208605069_0003/
Kill Command = /home/ec2-user/hadoop-2.6.4/bin/hadoop job  -kill job_1665208605069_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-10-08 19:45:05,454 Stage-1 map = 0%,  reduce = 0%
2022-10-08 19:45:14,178 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.11 sec
MapReduce Total cumulative CPU time: 2 seconds 110 msec
Ended Job = job_1665208605069_0003
Stage-3 is selected by condition resolver.
Stage-2 is filtered out by condition resolver.
Stage-4 is filtered out by condition resolver.
Moving data to: hdfs://localhost/user/ec2-user/ThreeColExtract/.hive-staging_hive_2022-10-0
8_19-44-56_382_7377876875745470242-1/-ext-10000
Moving data to: ThreeColExtract
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.11 sec   HDFS Read: 11770539 HDFS Write: 627873 S
UCCESS
Total MapReduce CPU Time Spent: 2 seconds 110 msec
OK
Time taken: 19.965 seconds
hive> 
```

5. **verify that the new output file has been created (the file will be called 000000_0)**
   The file would be created in HDFS in user home directory (/user/ec2-user/ThreeColExtract)



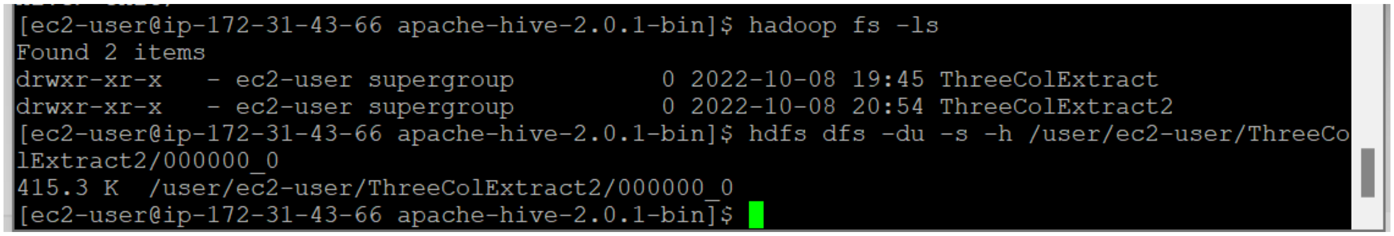
```
[ec2-user@ip-172-31-43-66 ~]$ hadoop fs -ls
Found 1 items
drwxr-xr-x   - ec2-user supergroup          0 2022-10-08 19:45 ThreeColExtract
[ec2-user@ip-172-31-43-66 ~]$ hadoop fs -du /user/ec2-user/ThreeColExtract
627873  /user/ec2-user/ThreeColExtract/000000_0
[ec2-user@ip-172-31-43-66 ~]$ ^C
[ec2-user@ip-172-31-43-66 ~]$ hdfs dfs -du -s -h /user/ec2-user/ThreeColExtract/000000_0
613.2 K  /user/ec2-user/ThreeColExtract/000000_0
[ec2-user@ip-172-31-43-66 ~]$ 
```

6. **create a new table that is going to load 8 columns instead of 5 in our example (i.e. create and load a new table that defines 8 columns by including columns city08U,cityA08,cityA08U) and**

Nikhil Bhalala
Dsc -333
Assignment-2

use Hive to generate a new output file containing only the city08U and cityA08U columns from the vehicles.csv file. Report the size of that output file as well.

```
CREATE TABLE VehicleData2(
barrels08 FLOAT, barrelsA08 FLOAT,
charge120 FLOAT, charge240 FLOAT,
city08 FLOAT,
city08U FLOAT,cityA08 FLOAT ,cityA08U FLOAT)
ROW FORMAT DELIMITED FIELDS
TERMINATED BY ',' STORED AS TEXTFILE;

LOAD DATA LOCAL INPATH '/home/ec2-user/vehicles.csv'
OVERWRITE INTO TABLE VehicleData2;

INSERT OVERWRITE DIRECTORY 'ThreeColExtract2'
SELECT city08U, cityA08, cityA08U
FROM VehicleData2;
```



```
[ec2-user@ip-172-31-43-66 apache-hive-2.0.1-bin]$ hadoop fs -ls
Found 2 items
drwxr-xr-x   - ec2-user supergroup          0 2022-10-08 19:45 ThreeColExtract
drwxr-xr-x   - ec2-user supergroup          0 2022-10-08 20:54 ThreeColExtract2
[ec2-user@ip-172-31-43-66 apache-hive-2.0.1-bin]$ hdfs dfs -du -s -h /user/ec2-user/ThreeCo
lExtract2/000000_0
415.3 K  /user/ec2-user/ThreeColExtract2/000000_0
[ec2-user@ip-172-31-43-66 apache-hive-2.0.1-bin]$
```