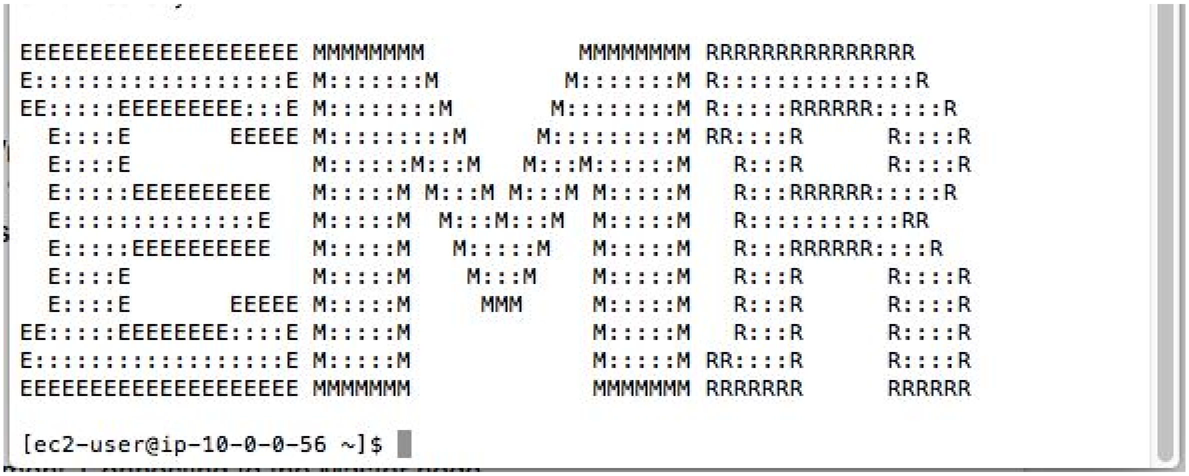
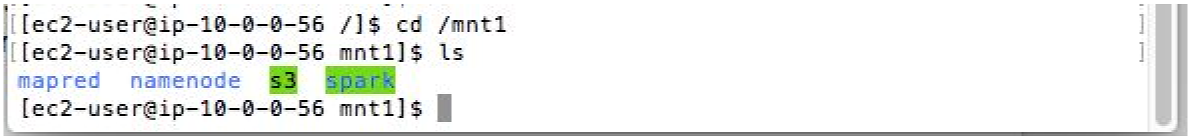
Uploading data to S3 via command line

1. First, use the instructions from Module 2, Session 3 to create an empty S3 bucket to store your data. Here is the link for the same.
2. Next, you need to connect to your Master node.



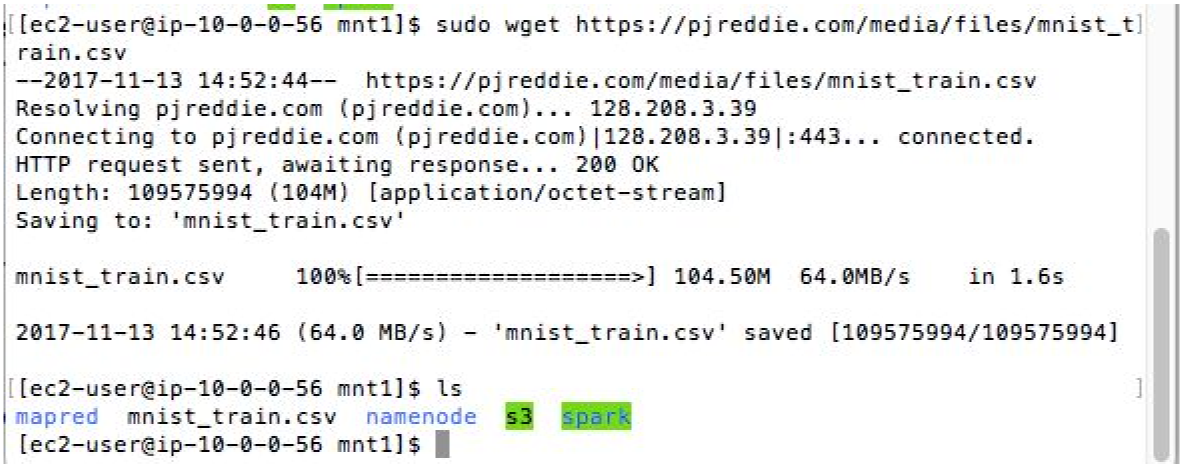
```
EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   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
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[ec2-user@ip-10-0-0-56 ~]$
```

3. After you are connected to your Master node, type ‘df’ to get a directory listing.
4. Find a directory which has enough space (typically, a < 10% value will accommodate most data-sets), and go into that directory (e.g. here, we used cd /mnt1)



```
[ec2-user@ip-10-0-0-56 /]$ cd /mnt1
[ec2-user@ip-10-0-0-56 mnt1]$ ls
mapred  namenode  s3  spark
[ec2-user@ip-10-0-0-56 mnt1]$
```

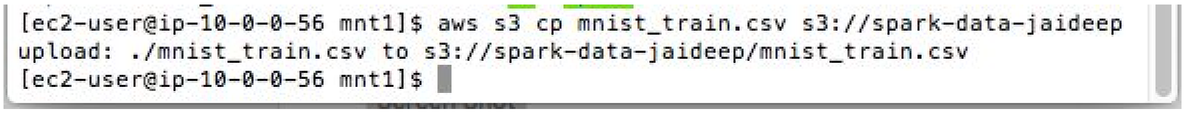
5. Now, use the following command on your Master terminal:
   sudo wget <link to file>
   (the sudo is necessary to override permissions. You can use ‘ls’ to verify that the file has been downloaded. As an example, we’ll download the MNIST data-set)



```
[ec2-user@ip-10-0-0-56 mnt1]$ sudo wget https://pjreddie.com/media/files/mnist_t
rain.csv
--2017-11-13 14:52:44--  https://pjreddie.com/media/files/mnist_train.csv
Resolving pjreddie.com (pjreddie.com)... 128.208.3.39
Connecting to pjreddie.com (pjreddie.com)|128.208.3.39|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 109575994 (104M) [application/octet-stream]
Saving to: 'mnist_train.csv'

mnist_train.csv     100%[===================>] 104.50M  64.0MB/s    in 1.6s

2017-11-13 14:52:46 (64.0 MB/s) - 'mnist_train.csv' saved [109575994/109575994]

[ec2-user@ip-10-0-0-56 mnt1]$ ls
mapred  mnist_train.csv  namenode  s3  spark
[ec2-user@ip-10-0-0-56 mnt1]$
```

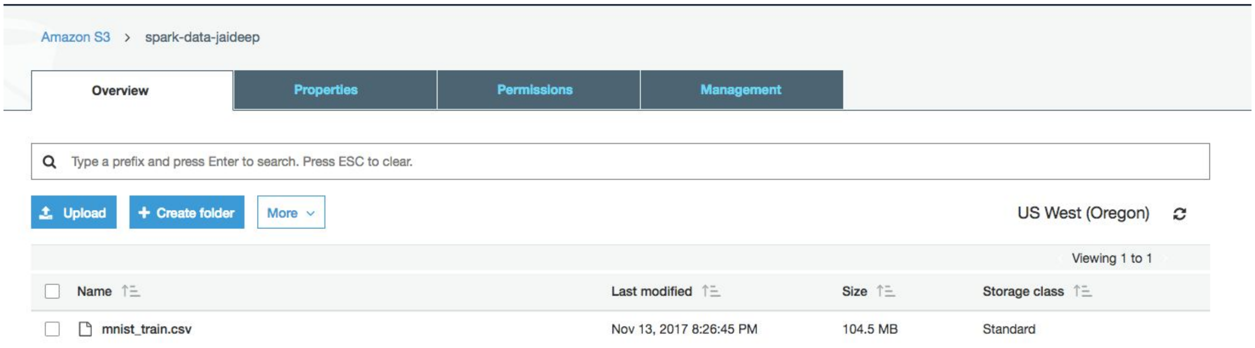
6. Sometimes, your data will be in .gz format. To unzip, use the command:
   sudo gzip -d filename
   (You can use ‘ls’ to verify that the file has been unzipped)

7. Now for the actual upload to S3. AWS provides a very powerful command line interface for all of their services. Run the following command:
   aws s3 cp filename s3://bucketname/



```
[ec2-user@ip-10-0-0-56 mnt1]$ aws s3 cp mnist_train.csv s3://spark-data-jaideep
upload: ./mnist_train.csv to s3://spark-data-jaideep/mnist_train.csv
[ec2-user@ip-10-0-0-56 mnt1]$
```

8. You’re all set! If you go to your S3 console, you will be able to see your data file there.



Amazon S3 › spark-data-jaideep

□	Name	Last modified	Size	Storage class
□	mnist_train.csv	Nov 13, 2017 8:26:45 PM	104.5 MB	Standard

Uploading data to S3 via command line

9. For copying a whole directory, use the following command:

   ```
   aws s3 cp -R <filename>
   ```

   (here, the -R stands for ‘recursive’, which means the command repeats itself through the entire directory and all its subdirectories)