## Lab 1: Test your first map reduce job In this lab you are going to get more familiar with the Map/ Reduce paradigm. Your first program will be a simple python program. Python is a really simple programming language. In the folder Module 1 you will find 3 files

## First download a pdf file. For that we will need to install wget on your instance and then you will want to convert the pdf file into a txt file to be able to count the number of words inside it. To do so execute the following commands with the sudo user.

| sudo yum -y install wget sudo yum install poppler-utils |
| --- |

Create a folder for this first lab and get inside this folder :

| mkdir lab1 cd lab1 |
| --- |

Download the last script of the movie the joker.

| wget https://script-pdf.s3-us-west-2.amazonaws.com/dear-white-people-script-pdf.pdf |
| --- |

Try to open it, you shouldn’t be able to see anything, only unicodes.

| head dear-white-people-script-pdf.pdf |
| --- |

We are now going to transform this pdf into a text file:  
To do so you need to execute the following command:

| pdftotext dear-white-people-script-pdf.pdf dear-white-people-script-pdf.txt head head dear-white-people-script-pdf.txt |
| --- |

Create two files one mapper.py and reducer.py. For that you can use vim or your favorite tool.

| vi mapper.py |
| --- |

| #!/usr/bin/env python  import sys    # Get input lines from stdin  for line in sys.stdin:  # Remove spaces from beginning and end of the line  line = line.strip()  # Split it into words  words = line.split()  # Output tuples on stdout  for word in words:  print '%s\t%s' % (word, "1") |
| --- |

To save you only need to execute :wq.  
  
Then create the file reducer.py :

| vi reducer.py |
| --- |

| #!/usr/bin/env python  import sys    # Create a dictionary to map words to counts  wordcount = {}    # Get input from stdin  for line in sys.stdin:  #Remove spaces from beginning and end of the line  line = line.strip()    # parse the input from mapper.py  word, count = line.split('\t', 1)  # convert count (currently a string) to int  try:  count = int(count)  except ValueError:  continue    try:  wordcount[word] = wordcount[word]+count  except:  wordcount[word] = count    # Write the tuples to stdout  # Currently tuples are unsorted  for word in wordcount.keys():  print '%s\t%s'% ( word, wordcount[word] ) |
| --- |

You need to make the two files executable. For that execute the following command :

| chmod +x mapper.py  chmod +x reducer.py |
| --- |

Check that the mapper is producing the good result. Execute the following command:

| cat dear-white-people-script-pdf.txt | python mapper.py |
| --- |

**Why are we using the character | ?**

Try also the following command:

| cat dear-white-people-script-pdf.txt | python mapper.py | sort |
| --- |

What are we using “sort” ?

Execute the following command :

| cat dear-white-people-script-pdf.txt | python mapper.py | sort | python reducer.py |
| --- |

Check that this small program is producing the expected result. If it works, congratulations, you just realized your first Map / Reduce program using the command line.

Test the performances of your first program.

| time dear-white-people-script-pdf.txt | python mapper.py | sort | python reducer.py |
| --- |

To make sure your program is working properly, you can also create your own file with your own content and play with it.

## Lab 2.2: Write your first map reduce job in python You have the csv file Groceries\_file.csv. You need to write a mapReduce job in python to get the number of items bought.