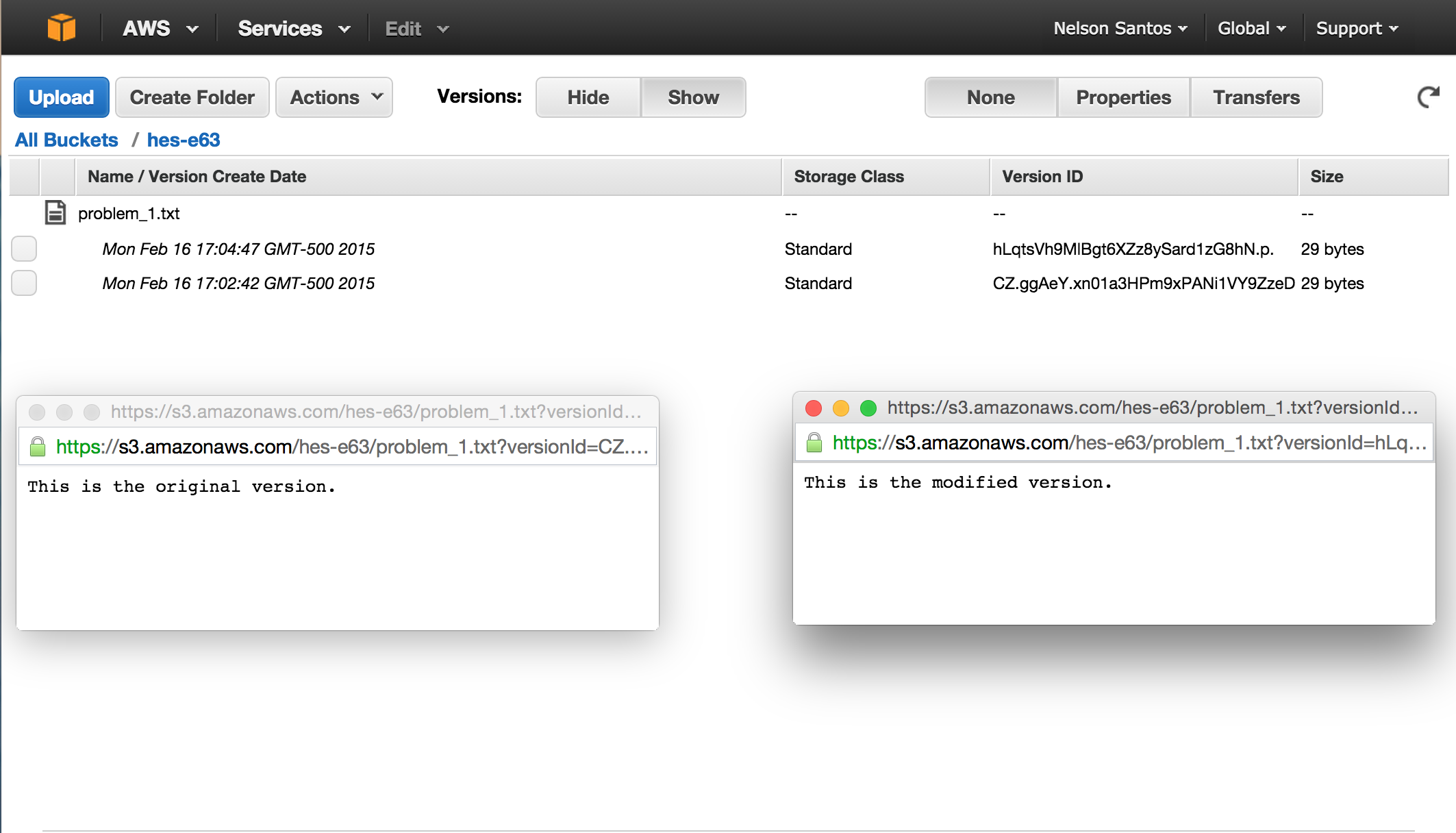
# Homework 03

Nelson Santos

[nsantos01@g.harvard.edu](mailto:nsantos01@g.harvard.edu)

## Problem 1.

The screenshot bellow shows that once versioning was enabled, I was able to access different versions of a modified file by selecting the **Show** option for the versions.

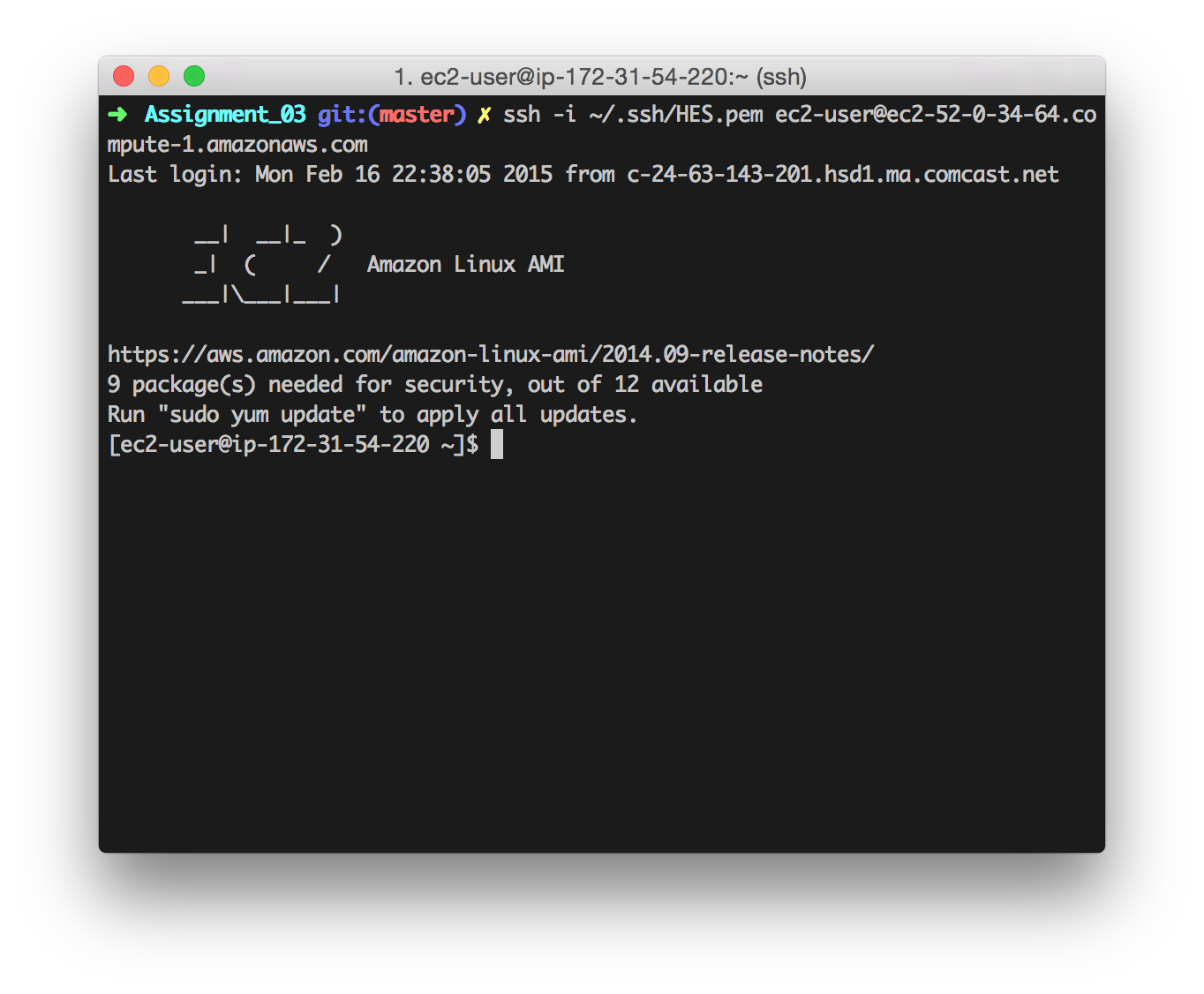


The screenshot bellow shows the result of navigating to <http://s3.amazonaws.com/hes-e63/problem_1.jpg>.



## Problem 2.

The screenshot bellow shows an SSH connection to my EC2 instance.



The screenshot bellow shows an SCP operation from local to EC2 (local\_ec2.txt), and from EC2 to local (ec2\_local.txt).



## Problem 3.

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; Clojure MapReduce word counter ;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(ns map-reduce.master

(:gen-class)

(:require [clojure.string :as str]

[clojure.pprint :as pp]))

(declare mapper combiner sum reducer)

(defn -main [& args]

(with-open [rdr (clojure.java.io/reader "Jabberwocky.txt")]

(->> (doall (line-seq rdr))

(map mapper)

(combiner)

(reducer)

(sort)

(pp/pprint))))

(defn mapper [line]

(map #(vector % 1) (str/split (str/lower-case line) #"\W")))

(defn combiner [mapped]

(->> (apply concat mapped)

(group-by first)

(map (fn [[k v]]

{k (map second v)}))))

(defn sum [word]

{(apply key word) (apply + (apply val word))})

(defn reducer [collected-values]

(apply merge (map sum collected-values)))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

|  |  |
| --- | --- |
| egrep -oi "\w+" Jabberwocky.txt|tr '[:upper:]' '[:lower:]'|sort|uniq -c  1 1872  1 alice  2 all  15 and  1 arms  2 as  1 awhile  1 back  1 bandersnatch  1 beamish  2 beware  1 bird  1 bite  1 blade  2 borogoves  1 boy  2 brillig  1 burbled  2 by  1 callay  1 callooh  2 came  1 carroll  1 catch  1 chortled  1 claws  1 come  1 day  1 dead  2 did  1 eyes  1 flame  1 foe  1 found  1 frabjous  1 from  1 frumious  1 galumphing  2 gimble  1 glass  2 gyre  1 hand  1 has  7 he  1 head  2 his  6 in  2 it  1 its  3 jabberwock  1 jabberwocky  1 jaws  1 joy  1 jubjub  1 left  1 lewis  1 long  1 looking  1 manxome  2 mimsy  2 mome  3 my  1 o  1 of  2 one  2 outgrabe  2 raths  1 rested  1 shun  1 slain  2 slithy  1 snack  1 snicker  1 so  1 son  1 sought  2 stood  1 sword  2 that  20 the  1 there  1 thou  2 thought  4 through  1 time  1 to  1 took  2 toves  1 tree  1 tulgey  1 tumtum  2 twas  2 two  1 uffish  2 vorpal  2 wabe  2 went  2 were  1 what  1 whiffling  2 with  1 wood | java -jar target/uberjar/map-reduce-0.1.0-SNAPSHOT-standalone.jar  (["" 60]  ["1872" 1]  ["alice" 1]  ["all" 2]  ["and" 15]  ["arms" 1]  ["as" 2]  ["awhile" 1]  ["back" 1]  ["bandersnatch" 1]  ["beamish" 1]  ["beware" 2]  ["bird" 1]  ["bite" 1]  ["blade" 1]  ["borogoves" 2]  ["boy" 1]  ["brillig" 2]  ["burbled" 1]  ["by" 2]  ["callay" 1]  ["callooh" 1]  ["came" 2]  ["carroll" 1]  ["catch" 1]  ["chortled" 1]  ["claws" 1]  ["come" 1]  ["day" 1]  ["dead" 1]  ["did" 2]  ["eyes" 1]  ["flame" 1]  ["foe" 1]  ["found" 1]  ["frabjous" 1]  ["from" 1]  ["frumious" 1]  ["galumphing" 1]  ["gimble" 2]  ["glass" 1]  ["gyre" 2]  ["hand" 1]  ["has" 1]  ["he" 7]  ["head" 1]  ["his" 2]  ["in" 6]  ["it" 2]  ["its" 1]  ["jabberwock" 3]  ["jabberwocky" 1]  ["jaws" 1]  ["joy" 1]  ["jubjub" 1]  ["left" 1]  ["lewis" 1]  ["long" 1]  ["looking" 1]  ["manxome" 1]  ["mimsy" 2]  ["mome" 2]  ["my" 3]  ["o" 1]  ["of" 1]  ["one" 2]  ["outgrabe" 2]  ["raths" 2]  ["rested" 1]  ["shun" 1]  ["slain" 1]  ["slithy" 2]  ["snack" 1]  ["snicker" 1]  ["so" 1]  ["son" 1]  ["sought" 1]  ["stood" 2]  ["sword" 1]  ["that" 2]  ["the" 20]  ["there" 1]  ["thou" 1]  ["thought" 2]  ["through" 4]  ["time" 1]  ["to" 1]  ["took" 1]  ["toves" 2]  ["tree" 1]  ["tulgey" 1]  ["tumtum" 1]  ["twas" 2]  ["two" 2]  ["uffish" 1]  ["vorpal" 2]  ["wabe" 2]  ["went" 2]  ["were" 2]  ["what" 1]  ["whiffling" 1]  ["with" 2]  ["wood" 1]) |

The only difference between the two is that my solution uses a split approach that included non-word characters, represented by the empty string count.

## Problem 4.

EMR Log output:

Downloading 's3://elasticmapreduce/libs/pig/pig-script' to '/mnt/var/lib/hadoop/steps/s-18ZRNYHV5Q1NP/.'

2015-02-17 03:25:12 GMT - INFO java version: 1.7

hadoop version 2.4.0

java version: 1.7

pig versions: 0.12.0

2015-02-17 03:25:12 GMT - INFO Running: /home/hadoop/pig/bin/pig -l . '-f' 's3://elasticmapreduce/samples/pig-apache/do-reports2.pig' '-p' 'INPUT=s3://elasticmapreduce/samples/pig-apache/input' '-p' 'OUTPUT=s3://hes-e63/pig-apache/output'

It looks like the script parsed and analyzed web server logs. These logs are created and stored as single line events with multiple attributes, such as requester (client) IP, HTTP referer, request size, and more. The script parsed those logs (which could well be stored on multiple files, on different machines) and ranked (top 50’s) and aggregated (bytes/hour) the information.