Ryan Ballenger
CSCIE63 Big Data Analytics
Assignment 2

Assignment 2 Solutions

HU Extension School E-63 Big Data Analytics

Assignment 02

Handed out: 02/06/2016 Due by 11:30 PM on Friday, 02/12/2016

Capture all steps of your implementation with comments indicating what are accomplishing with every step. Place those in this MS Word document bellow the problem statement. Please send comments and questions to the Discussion Forum on the class site.

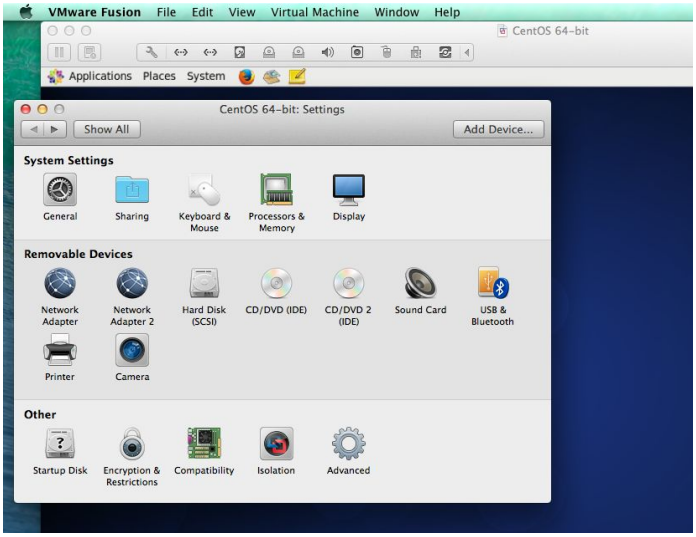
Problem 1) Please, download and install VMware Workstation 11 on your 64 bit Windows PC or VMWare Fusion 7, if you are on a MAC. Please download 64 bit CentOS6.7 and create a 64 bit VM. If you know what you are doing and you work with another flavor of Linux supported by CDH5.5.1, please be free to create a virtual machine based on your favorite Linux flavor. Provide your virtual machine with some 40GB of disk space, if you can spare it. For whatever reasons, Hadoop installation appears to prefer to have more than 20 GB of available space. Name the main user of your VM `cloudera`. Do not use name `hadoop`. "`hadoop`" is a bad name for a user, since Hadoop framework has an executable called `hadoop` and it creates many directories with that same name and those would not necessarily be owned by the VM user called `hadoop`. On that VM create yet another user called `joe`. Make both users `sudo` users. Once your CentOS is fully installed, please shut the VM down and make a copy of the entire directory containing that VM. Name the folder containing that copy differently. Two VMs are identical and you could even run them simultaneously if your machine has enough memory. In the folder of each VM add a text file describing OS on your VM, `usernames` and passwords of important users. This little file will make your VMs useful long into the future. The reason you are creating the backup VM is to save time, if you damage the one VM on which you are installing your software.

Please do not capture installation of Workstation 12 or Fusion. Please do not capture every step in creation of VM.

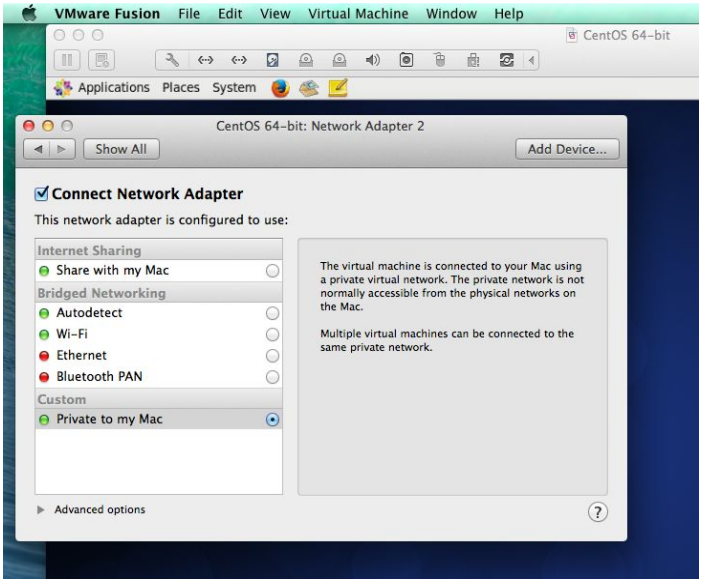
Show addition of the second network adapter.

I added the second network adapter as directed. To do this, I clicked the following.
Virtual Machine -> Settings… -> Network Adapter ->
Add Device -> Network Adapter -> Add
Then, I selected **Private to my Mac** and closed the window. Resulting screenshots below illustrate that the second network adapter was added.

The network adapter is shown in the **Settings…** menu.



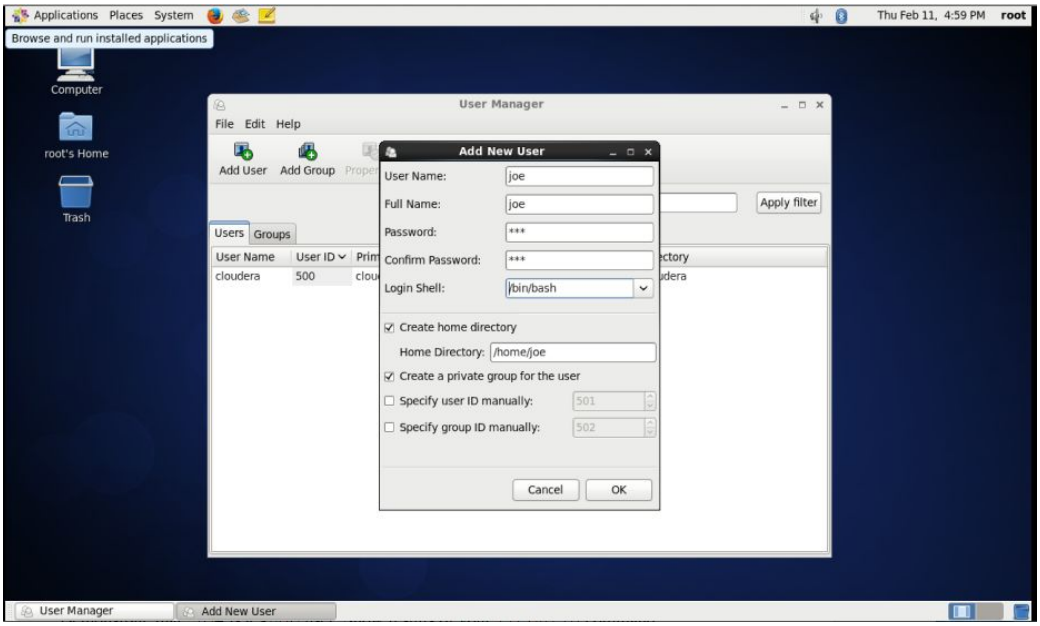
The details of the second network adapter are shown including the custom setting Private to my Mac.



Show steps in creation of user `joe`.

To add Joe, I selected Add User through the following menus.
System -> **Administration** -> **Users and Groups**, and finally, **Add User**

The screenshot below shows the form for Joe's information and final step of adding this user.



To give Jim sudo permissions, first I access the root account where I can
allocate sudo permissions to other users.
$su - root

Next, I change the permissions on the "sudoers" file in order to access it.
$chmod a+w /etc/sudoers

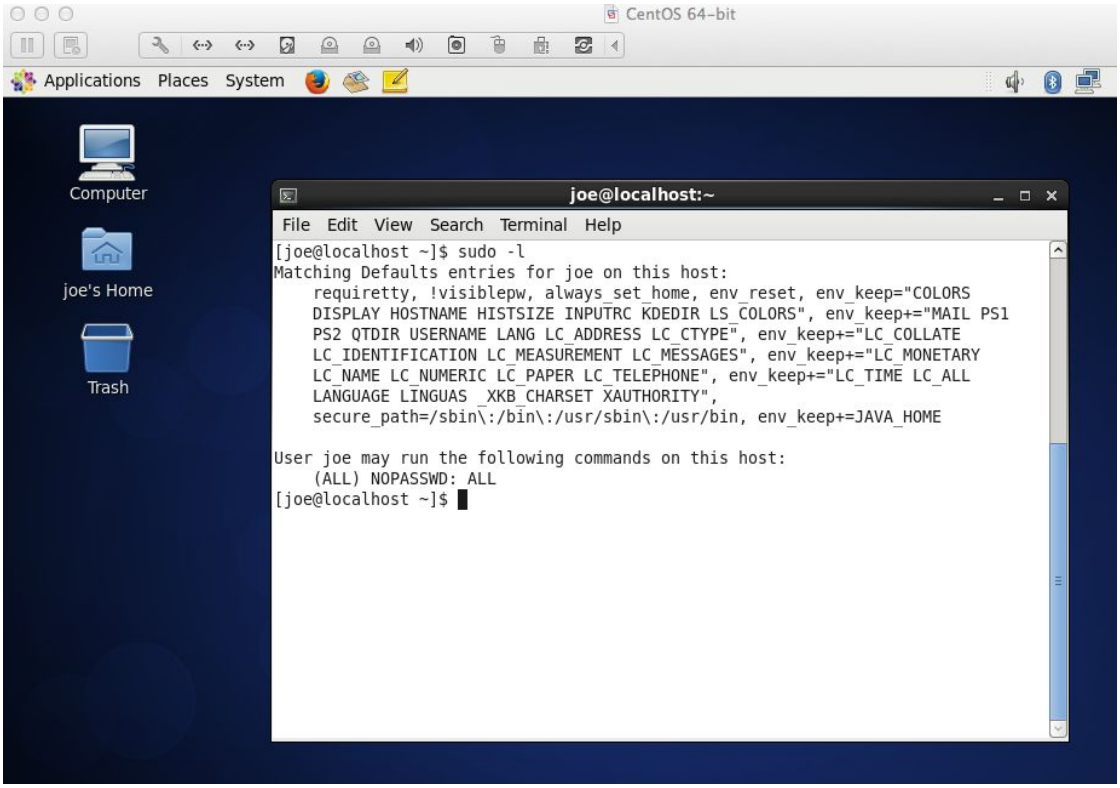
# I use visudo to edit the "sudoers" file which checks syntax since mistakes
# in this file could make the system inoperable.
$ visudo

I add the following line to the bottom on the "sudoers" file to give joe access
to sudo permissions. This gives him sudo permissions without needing a password.
joe ALL=(ALL) NOPASSWD: ALL

# After exiting and saving with :x, I reapply the priviledges to the "sudoers"
# file as instructed. This completes the steps to give sudo permission.
$ chmod –w /etc/sudoers

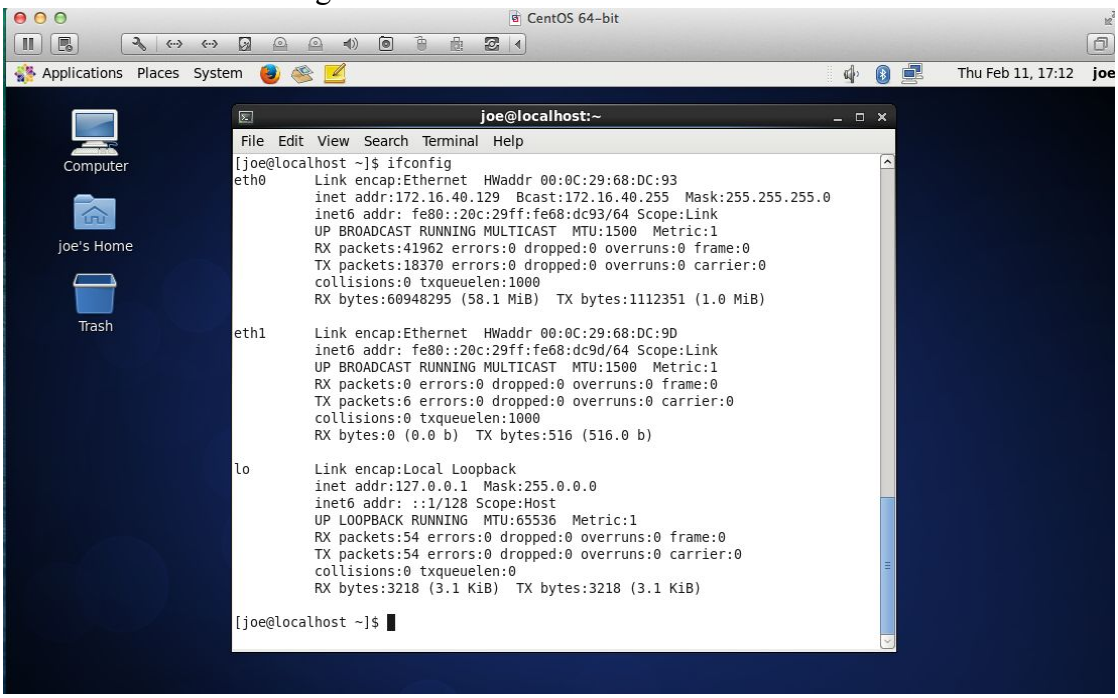
Demonstrate that `joe` is a `sudo` user.

User joe can run "sudo -l" which shows his commands and also proves he is now a sudo user. The screenshot below shows the results of joe successfully running "sudo -l".



Show results of your `ifconfig` command.

The results of the ifconfig command are shown in the screenshot below.



Problem 2) Use one of above VMs and follow closely steps in the CDH5.5.1 Quick Start Guide, or my notes. PDF and PPT formats and characters on PC do not always map well into Unix (Linux) characters. If you want to copy commands from the guide you are better off doing it from the HTML version of the CDH Quick Start Guide, which you could find at:
http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_qs_quick_start.html and open from with your VM.
Both my notes and the Quick Start Guide will lead you through a "semi-automated" process of installing Hadoop. Please install YARN version of Hadoop. My notes add a few explanations beyond what you can see in the Cloudera's guide. Read the notes and the guide very carefully. Do not execute commands for flavors of Linux other than RedHat (CentOS) unless you are working with another flavor purposefully. **You will know that you have successfully installed Hadoop if all of tests described in the guide work properly.**

```
# First, I installed Java which is required for Cloudera's Hadoop (CDH).
# After downloading the RPM, I installed it with the following command.
$ rpm –ivh jdk-8u60-linux-x64.rpm

# Then, I added JAVA_HOME to the root's bash profile in order to make the
# installation of java accessible at all times.
$ cd ~
$ vi /root/.bash_profile

        # After opening the bash profile, I added  or edited the following lines.
        JAVA_HOME=/usr/java/jdk1.8.0_60
        export JAVA_HOME
        PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
        export PATH

# To make the session aware of the changes, I ran source .bash_profile as directed.
$cd /root
$ source .bash_profile

# To begin installation of CDH, I downloaded the RPM and ran the following
# command from the downloads folder.
$ sudo yum --nogpgcheck localinstall cloudera-cdh-5-0.x86_64.rpm

# Then, I installed Hadoop with YARN in psuedo-distributed mode. For reference,
# I used the lecture slides' steps to install and not Cloudera's Quick Start guide.
$ sudo yum install hadoop-conf-pseudo

# Before starting any nodes, I formatted the HDFS filesystem. The CDH has its
# own filesystem accessible later through hadoop -fs … and it is initiated at this step.
```

```
$ sudo –u hdfs hdfs namenode –format

# Next, I locate the Hadoop executables and start them with the service command.
# This starts the HDFS filesystem.
$ cd /etc/init.d
$ do sudo service hadoop-hdfs-datanode start ;
$ do sudo service hadoop-hdfs-namenode start ;
$ do sudo service hadoop-hdfs-secondarynamenode start ;

# At this step, I created home directories and service directories. First I remove the
# /tmp folder with the recursive command. This gives users like hadoop-yarn the
# necessary directories to run.
$ sudo -u hdfs hadoop fs -rm -r /tmp
$ sudo /usr/lib/hadoop/libexec/init-hdfs.sh

# To start YARN, I run the following commands while still in the /etc/init.d
# directory. This starts YARN's resource manager, node manager, and history
# server services
$ sudo service hadoop-yarn-resourcemanager start
$ sudo service hadoop-yarn-nodemanager start
$ sudo service hadoop-mapreduce-historyserver start

# At this point, I switch to the user cloudera and create the necessary home directories.
# This enables this user to run YARN as well.
$ su - cloudera
$ sudo -u hdfs hadoop fs -mkdir /user/$USER
$ sudo -u hdfs hadoop fs -chown $USER /user/$USER

# To run the example program on YARN, I make an input directory in the filesystem
# and copy the XML files, noted in the lectures slides, to this directory. Also the
# HOME variable to give access to hadoop-mapreduce.
$ hadoop fs -mkdir input
$ hadoop fs -put /etc/hadoop/conf/*.xml input
$ export HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce

# Finally to run the Hadoop grep example, I do the following command.
$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce- examples.jar grep input
output23 'dfs[a-z.]+'

# The output of the program is shown with an "ls" command to the hadoop filesystem.
# Below is the result and the _SUCCESS shows it worked properly. Additionally,
# the head of the part-r-00000 file is outputted with the "cat" command to
# show the results.
[cloudera@localhost ~]$ hadoop fs -ls output23
Found 2 items
```

```
-rw-r--r--   1 cloudera supergroup          0 2016-02-11 20:59 output23/_SUCCESS
-rw-r--r--   1 cloudera supergroup        244 2016-02-11 20:59 output23/part-r-00000
[cloudera@localhost ~]$ hadoop fs -cat output23/part-r-00000 | head
1       dfs.safemode.min.datanodes
1       dfs.safemode.extension
1       dfs.replication
1       dfs.namenode.name.dir
1       dfs.namenode.checkpoint.dir
1       dfs.domain.socket.path
1       dfs.datanode.hdfs
1       dfs.datanode.data.dir
1       dfs.client.read.shortcircuit
1       dfs.client.file
```

Problem 3) As your new Linux user joe fetch the .txt version of James Joyce's Ulysses by issuing the following command on the command prompt:

```
wget http://www.gutenberg.org/files/4300/4300.zip
```

Unzip the file. Open the resulting txt file with Vi and convince yourself that the life of Buck Mulligan is in front of you. **Create a HDFS directory called `ulysses` and copy the .txt file into that HDFS directory.**

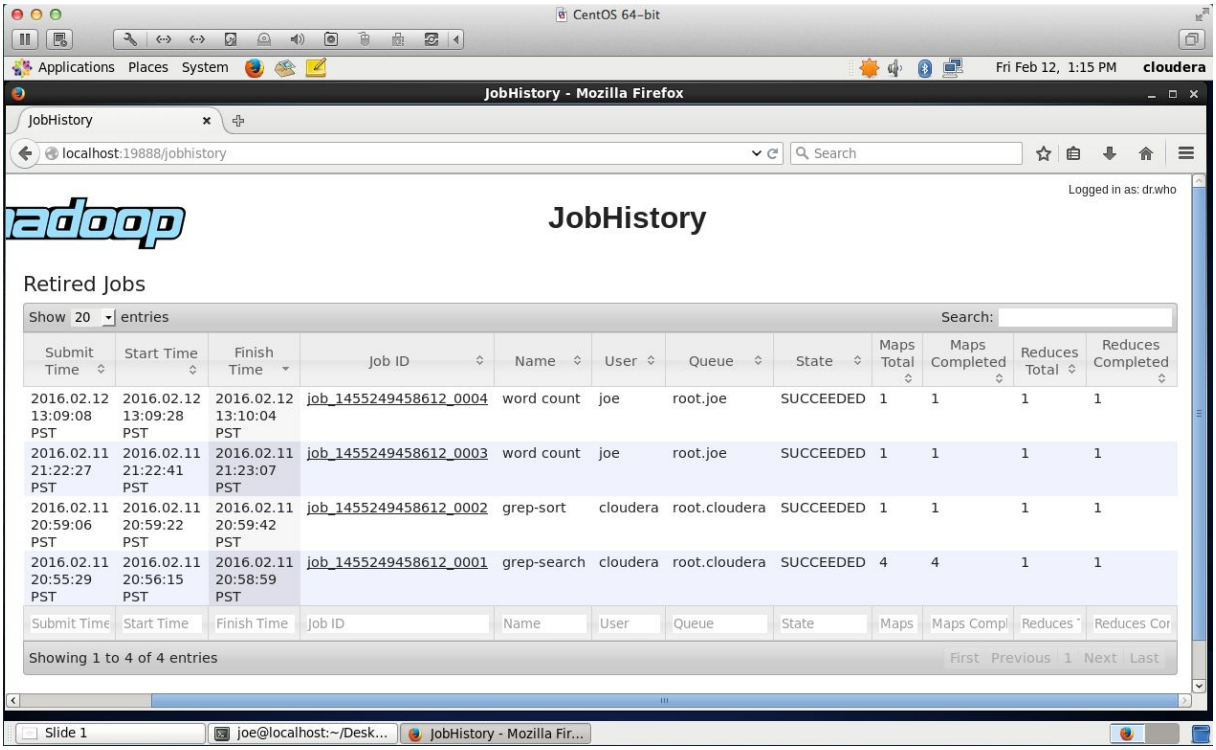
```
# After downloading the zip file, I unzip it as the user Joe. Then, I make the new
# directory ulysses in the hadoop filesystem and copy the .txt file there
# where it will be accessible for a hadoop-mapreduce program.
$ unzip /home/joe/Desktop/4300.zip
$ hadoop fs -mkdir ulysses
$ hadoop fs -put /home/joe/Desktop/4300.txt ulysses
```

Do not create another HDFS directory called `counted`. The Map Reduce job you will run will create that directory for its output. Actually, if the directory preexists the job will raise an error. That same `hadoop-mapreduce-examples.jar` file mentioned in class notes and you used as the final proof that MapReduce works contains another program called `wordcount`. `wordcount` will tell you how many times a word appears in a provided text. Invoke `wordcount` by the following command:

```
$ hadoop jar
/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
wordcount ulysses counted
```

Once the job is finished visit site http://localhost:19888. **You will see some statistics on MapReduce jobs executed on your cluster.** There will not be much for your short job. In general that is a very useful site.

After running the program designated above, hadoop's job history is shown to illustrate that the job was run successfully. The screenshot below is from http://localhost:19888.



Copy results of word count analysis to the local file system.

# To copy the results to the local filesystem, I transfer the part-r-00000 file
# to the shared VM directory.
$ hadoop fs -copyToLocal counted/part-r-00000 /mnt/hgfs/VM_shared/

Write a small program in any language (or scripting tool) of your choice and order the counting results by the decreasing count. Present the portion of your final result which does not contain so called stop words (the, a, and, or, …) in your report. Submit top 200 words in separate .txt file with your report.

I used the Default English Stopwords list at http://www.ranks.nl/stopwords.

My sorter Sorter.java is part of my submission.

The top ten results are below excluding stopwords. The top 200 words are part of my submission and named top200.txt.

Word	**Count**
Mr	699
like	649

one	498
said.	480
says	450
Bloom	428
said	423
old	419
Stephen	315
two	302

Problem 4). Consider a symmetric matrix

A = [3 2 4 2 0 2 4 2 3]

Using R demonstrate that all three eigenvectors of that matrix are mutually orthogonal.

Below I show the dot products of each pair of eigenvectors of the given matrix are 0, which proves the three eigenvectors are mutually orthogonal.

```
# First, I calculate the eigenvectors of the given matrix.
> eigVects = eigen(matrix(c(3, 2, 4, 2, 0, 2, 4, 2, 3), nrow=3))$vectors
> eigVects
   [,1]       [,2]       [,3]
[1,] 0.6666667  0.7453560  0.0000000
[2,] 0.3333333 -0.2981424 -0.8944272
[3,] 0.6666667 -0.5962848  0.4472136
```

Dot product of eigenvector #1 and #2

```
> sum(eigVects[,1] * eigVects[,2])
[1] 3.053113e-16
# 3.053113e-16 ≈ 0
```

Dot product of eigenvector #1 and #3

```
> sum(eigVects[,1] * eigVects[,3])
[1] 5.551115e-17
# 5.551115e-17 ≈ 0
```

Dot product of eigenvector #2 and #3

```
> sum(eigVects[,2] * eigVects[,3])
[1] -1.110223e-16
# -1.110223e-16 ≈ 0
```

Let Λ be the matrix of eigenvectors of matrix A. Calculate product of tree matrices:

$\Lambda^T A \Lambda$

Symbol T indicates the transpose matrix. Google around for properties of eigenvectors and eigenvalues of real symmetric matrices. What is the general statement you can make about the observation on the value of the above product. Include copies of your R commands in your MS Word report.

```
> A = matrix(c(3, 2, 4, 2, 0, 2, 4, 2, 3), nrow=3)
> eigVects = eigen(A)$vectors
> t(eigVects) %*% A %*% eigVects
              [,1]          [,2]          [,3]
[1,] 8.000000e+00  2.664535e-15  1.332268e-15
[2,] 3.441691e-15 -1.000000e+00  2.220446e-16
[3,] 1.221245e-15  1.665335e-16 -1.000000e+00
```

The general statement that can be made is that the product is a diagonal matrix with the eigenvalues in the diagonal. The eigenvalues of matrix A are 8, -1, and -1 which are shown in that order in the diagonal.

A property of diagonal matrices is that their eigenvalues are real. This is also demonstrated in the product above where 8.000000e+00 = 8, -1.000000e+00 = -1, and -1.000000e+00 = -1 are listed in the diagonal. These are real numbers of course which typically is not a characteristic of eigenvalues.