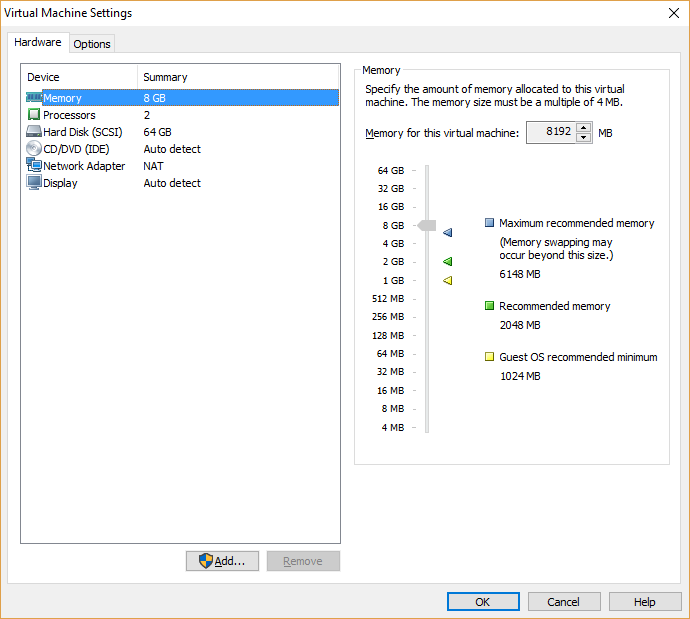
HU Extension Assignment 03 E63 Big Data Analytics

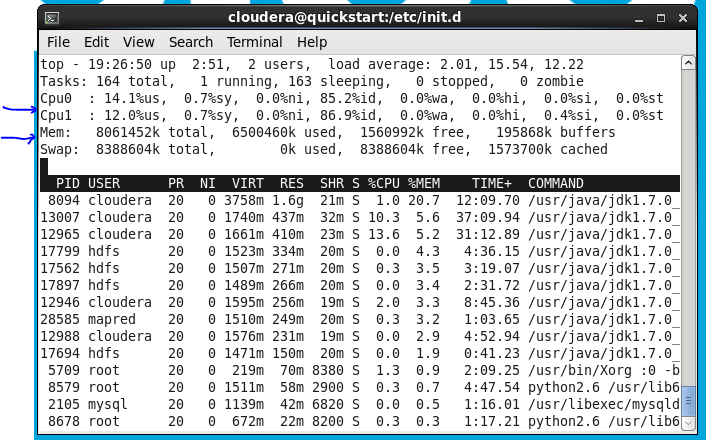
**Problem 1.** Download Quick Start VM for CDH 5.8 from <https://www.cloudera.com/downloads/quickstart_vms/5-8.html>.

Start the VM. Please assign to the VM as much memory as you can. Examine whether hadoop-hdfs-\* ,hadoop-mapreduce-\* and hadoop-yarn-\* daemons are running. If those daemons are not running start all of them.

Cloudera VM settings 8 GB memory, 2 cores, 64GB hard disk.



Linux ‘top’ command showing system resources that was indeed allocated - 2 cores, 8GB RAM etc.



We will use sudo command to check the status of various hadoop daemon services.

sudo will execute a command as another user.

|  |
| --- |
| **[cloudera@quickstart conf.pseudo]$** *cd /etc/init.d*  **[cloudera@quickstart init.d]$** *for i in `ls hadoop-hdfs-\*`; do sudo service $i status; done*  Hadoop datanode is not running [FAILED]  Hadoop journalnode is not running [FAILED]  Hadoop namenode is not running [FAILED]  Hadoop secondarynamenode is not running [FAILED]  **[cloudera@quickstart init.d]$** *for i in `ls hadoop-mapreduce-\*`; do sudo service $i status; done*  Hadoop historyserver is not running [FAILED]  **[cloudera@quickstart init.d]$** f*or i in `ls hadoop-yarn-\*`; do sudo service $i status; done*  Hadoop nodemanager is not running [FAILED]  Hadoop proxyserver is not running [FAILED]  Hadoop resourcemanager is not running [FAILED] |

Start various hadoop services

|  |
| --- |
| **[cloudera@quickstart init.d]$** *for i in `ls hadoop-hdfs-\*`; do sudo service $i start; done*  starting datanode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-datanode-quickstart.cloudera.out  log4j:WARN No appenders could be found for logger (org.apache.hadoop.hdfs.server.datanode.DataNode).  log4j:WARN Please initialize the log4j system properly.  log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.  Started Hadoop datanode (hadoop-hdfs-datanode): [ OK ]  starting journalnode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-journalnode-quickstart.cloudera.out  log4j:WARN No appenders could be found for logger (org.apache.hadoop.hdfs.qjournal.server.JournalNode).  Started Hadoop journalnode: [ OK ]  starting namenode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-namenode-quickstart.cloudera.out  Started Hadoop namenode: [ OK ]  starting secondarynamenode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-secondarynamenode-quickstart.cloudera.out  Started Hadoop secondarynamenode: [ OK ]  **[cloudera@quickstart init.d]**$ *for i in `ls hadoop-mapreduce-\*`; do sudo service $i start; done*  starting historyserver, logging to /var/log/hadoop-mapreduce/mapred-mapred-historyserver-quickstart.cloudera.out  Started Hadoop historyserver: [ OK ]  **[cloudera@quickstart init.d]**$ *for i in `ls hadoop-yarn-\*`; do sudo service $i start; done*  starting nodemanager, logging to /var/log/hadoop-yarn/yarn-yarn-nodemanager-quickstart.cloudera.out  Started Hadoop nodemanager: [ OK ]  starting proxyserver, logging to /var/log/hadoop-yarn/yarn-yarn-proxyserver-quickstart.cloudera.out  Started Hadoop proxyserver: [ OK ]  starting resourcemanager, logging to /var/log/hadoop-yarn/yarn-yarn-resourcemanager-quickstart.cloudera.out  Started Hadoop resourcemanager: [ OK ] |

Status of hadoop services

|  |
| --- |
| **[cloudera@quickstart init.d]$** *for i in `ls hadoop-hdfs-\*`; do sudo service $i status; done*  Hadoop datanode is running [ OK ]  Hadoop journalnode is running [ OK ]  Hadoop namenode is running [ OK ]  Hadoop secondarynamenode is running [ OK ]  **[cloudera@quickstart init.d]**$ *for i in `ls hadoop-mapreduce-\*`; do sudo service $i status; done*  Hadoop historyserver is running [ OK ]  **[cloudera@quickstart init.d]**$ *for i in `ls hadoop-yarn-\*`; do sudo service $i status; done*  Hadoop nodemanager is running [ OK ]  Hadoop proxyserver is dead and pid file exists [FAILED]  Hadoop resourcemanager is running [ OK ] |

**Problem 2**. Examine whether there are HDFS directories for users: spark, hive, oozie. If the directories are present find the content of those directories. If the directories are not present on your VM, create them. In that case you first have to format the namenode first.

Those user accounts don’t exist on the fresh VM. So we would format the namenode first using hdfs command before creating those user accounts using init-hdfs.sh script.

*sudo -u hdfs hdfs namenode -format*

|  |
| --- |
| **[cloudera@quickstart init.d]**$ *sudo /usr/lib/hadoop/libexec/init-hdfs.sh*  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /tmp'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chmod -R 1777 /tmp'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /var'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /var/log'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chmod -R 1775 /var/log'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown yarn:mapred /var/log'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /tmp/hadoop-yarn'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown -R mapred:mapred /tmp/hadoop-yarn'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /tmp/hadoop-yarn/staging/history/done\_intermediate'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown -R mapred:mapred /tmp/hadoop-yarn/staging'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chmod -R 1777 /tmp'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /var/log/hadoop-yarn/apps'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chmod -R 1777 /var/log/hadoop-yarn/apps'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown yarn:mapred /var/log/hadoop-yarn/apps'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /hbase'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown hbase /hbase'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /benchmarks'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chmod -R 777 /benchmarks'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/history'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown mapred /user/history'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/jenkins'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chmod -R 777 /user/jenkins'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown jenkins /user/jenkins'  *+ su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p* ***/user/hive****'*  *+ su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chmod -R 777 /user/hive'*  *+ su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown hive /user/hive'*  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/root'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chmod -R 777 /user/root'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown root /user/root'  *+ su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p* ***/user/hue****'*  *+ su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chmod -R 777 /user/hue'*  *+ su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -chown hue /user/hue'*  *+ su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p* ***/user/oozie****'*  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/oozie/share'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/oozie/share/lib'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/oozie/share/lib/hive'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/oozie/share/lib/mapreduce-streaming'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/oozie/share/lib/distcp'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/oozie/share/lib/pig'  + su -s /bin/bash hdfs -c '/usr/bin/hadoop fs -mkdir -p /user/oozie/share/lib/sqoop'  ... |

Verifying the presence of user accounts:

|  |
| --- |
| **[cloudera@quickstart init.d]**$ *cat /etc/passwd | grep -i hive*  hive:x:492:491:Hive:/var/lib/hive:/sbin/nologin  **[cloudera@quickstart init.d]**$ *cat /etc/passwd | grep -i spark*  spark:x:486:485:Spark:/var/lib/spark:/sbin/nologin  **[cloudera@quickstart init.d]**$ *cat /etc/passwd | grep -i oozie*  oozie:x:483:483:Oozie User:/var/lib/oozie:/bin/false  **[cloudera@quickstart init.d]$** *hadoop fs -ls /user*  Found 7 items  drwxr-xr-x - mapred supergroup 0 2017-02-14 19:19 /user/history  drwxrwxrwx - hive supergroup 0 2017-02-13 19:56 /user/hive  drwxrwxrwx - hue supergroup 0 2017-02-13 19:56 /user/hue  drwxrwxrwx - jenkins supergroup 0 2017-02-13 19:55 /user/jenkins  drwxrwxrwx - oozie supergroup 0 2017-02-13 19:57 /user/oozie  drwxrwxrwx - root supergroup 0 2017-02-13 19:56 /user/root  drwxr-xr-x - hdfs supergroup 0 2017-02-13 20:01 /user/spark |

Contents of user HDFS directories

|  |
| --- |
| **[cloudera@quickstart init.d]$** *sudo -u hdfs hadoop fs -ls -R /user/hive*  **[cloudera@quickstart init.d]$** *sudo -u hdfs hadoop fs -ls -R /user/spark*  drwxr-xr-x - spark supergroup 0 2017-02-13 20:01 /user/spark/applicationHistory  **[cloudera@quickstart init.d]$** *sudo -u hdfs hadoop fs -ls -R /user/oozie*  drwxrwxrwx - oozie supergroup 0 2017-02-13 19:57 /user/oozie/share  drwxrwxrwx - oozie supergroup 0 2017-02-13 19:57 /user/oozie/share/lib  drwxrwxrwx - oozie supergroup 0 2017-02-13 19:58 /user/oozie/share/lib/distcp  -rwxrwxrwx 1 oozie supergroup 1953236 2017-02-13 19:58 /user/oozie/share/lib/distcp/hadoop-distcp-2.6.0-cdh5.8.0.jar  -rwxrwxrwx 1 oozie supergroup 1953236 2017-02-13 19:58 /user/oozie/share/lib/distcp/hadoop-distcp.jar  drwxrwxrwx - oozie supergroup 0 2017-02-13 19:58 /user/oozie/share/lib/hive  -rwxrwxrwx 1 oozie supergroup 236660 2017-02-13 19:58 /user/oozie/share/lib/hive/ST4-4.0.4.jar  -rwxrwxrwx 1 oozie supergroup 4368200 2017-02-13 19:57 /user/oozie/share/lib/hive/accumulo-core-1.6.0.jar  -rwxrwxrwx 1 oozie supergroup 102069 2017-02-13 19:57 /user/oozie/share/lib/hive/accumulo-fate-1.6.0.jar  -rwxrwxrwx 1 oozie supergroup 57420 2017-02-13 19:57 /user/oozie/share/lib/hive/accumulo-start-1.6.0.jar  -rwxrwxrwx 1 oozie supergroup 117409 2017-02-13 19:57 /user/oozie/share/lib/hive/accumulo-trace-1.6.0.jar  -rwxrwxrwx 1 oozie supergroup 62983 2017-02-13 19:57 /user/oozie/share/lib/hive/activation-1.1.jar  -rwxrwxrwx 1 oozie supergroup 1997485 2017-02-13 19:57 /user/oozie/share/lib/hive/ant-1.9.1.jar  -rwxrwxrwx 1 oozie supergroup 18336 2017-02-13 19:57 /user/oozie/share/lib/hive/ant-launcher-1.9.1.jar  -rwxrwxrwx 1 oozie supergroup 445288 2017-02-13 19:57 /user/oozie/share/lib/hive/antlr-2.7.7.jar  -rwxrwxrwx 1 oozie supergroup 164368 2017-02-13 19:57 /user/oozie/share/lib/hive/antlr-runtime-3.4.jar  -rwxrwxrwx 1 oozie supergroup 448794 2017-02-13 19:57 /user/oozie/share/lib/hive/apache-log4j-extras-1.2.17.jar  -rwxrwxrwx 1 oozie supergroup 43398 2017-02-13 19:57 /user/oozie/share/lib/hive/asm-3.2.jar  -rwxrwxrwx 1 oozie supergroup 32693 2017-02-13 19:57 /user/oozie/share/lib/hive/asm-commons-3.1.jar  -rwxrwxrwx 1 oozie supergroup 21879 2017-02-13 19:57 /user/oozie/share/lib/hive/asm-tree-3.1.jar  -rwxrwxrwx 1 oozie supergroup 476769 2017-02-13 19:57 /user/oozie/share/lib/hive/avro.jar  -rwxrwxrwx 1 oozie supergroup 110600 2017-02-13 19:57 /user/oozie/share/lib/hive/bonecp-0.8.0.RELEASE.jar  -rwxrwxrwx 1 oozie supergroup 216853 2017-02-13 19:57 /user/oozie/share/lib/hive/calcite-avatica-1.0.0-incubating.jar  -rwxrwxrwx 1 oozie supergroup 3295091 2017-02-13 19:57 /user/oozie/share/lib/hive/calcite-core-1.0.0-incubating.jar  -rwxrwxrwx 1 oozie supergroup 432351 2017-02-13 19:57 /user/oozie/share/lib/hive/calcite-linq4j-1.0.0-incubating.jar  -rwxrwxrwx 1 oozie supergroup 188671 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-beanutils-1.7.0.jar  -rwxrwxrwx 1 oozie supergroup 206035 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-beanutils-core-1.8.0.jar  -rwxrwxrwx 1 oozie supergroup 41123 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-cli-1.2.jar  -rwxrwxrwx 1 oozie supergroup 58160 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-codec-1.4.jar  -rwxrwxrwx 1 oozie supergroup 588337 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-collections-3.2.2.jar  -rwxrwxrwx 1 oozie supergroup 30595 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-compiler-2.7.6.jar  -rwxrwxrwx 1 oozie supergroup 241367 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-compress-1.4.1.jar  -rwxrwxrwx 1 oozie supergroup 298829 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-configuration-1.6.jar  -rwxrwxrwx 1 oozie supergroup 160519 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-dbcp-1.4.jar  -rwxrwxrwx 1 oozie supergroup 143602 2017-02-13 19:57 /user/oozie/share/lib/hive/commons-digester-1.8.jar  …. |

**Problem 3**. On your VM create new Linux user. Call that user whatever you feel like.Please record the password of the new user. Make the user a member of group mapred.Create a HDFS directory for that user. Transfer ownership to the directory to new user.Also assign full permissions 1777 to that directory.

Create a new Linux user (eg: mpatnam)

|  |
| --- |
| # First login as root using su command.  **[cloudera@quickstart init.d]$** *su -*  Password:  # add new user  **[root@quickstart ~]#** *useradd mpatnam*  # add password for the new user  **[root@quickstart ~]#** *passwd mpatnam*  Changing password for user mpatnam.  New password:  Retype new password:  passwd: all authentication tokens updated successfully.  # Verify presence of new user  **[root@quickstart ~]#** *grep mpatnam /etc/passwd*  mpatnam:x:502:504::/home/mpatnam:/bin/bash  # make new user member of group mapred  **[root@quickstart ~]#** *usermod -g mpatnam mapred*  # login to the new user account using su command.  **[cloudera@quickstart init.d]$** *su - mpatnam*  Password:  **[mpatnam@quickstart ~]$** *whoami*  mpatnam  **[mpatnam@quickstart ~]$** *pwd*  /home/mpatnam |

Create HDFS directory for the new user

|  |
| --- |
| #create HDFS directory for the new user  **[cloudera@quickstart init.d]$** *sudo -u hdfs hadoop fs -mkdir -p /user/mpatnam*  #change permissions 777 (read/write/exec for owner/group/other)  **[cloudera@quickstart init.d]$** *sudo -u hdfs hadoop fs -chmod -R 777 /user/mpatnam*  #change ownership to the new user  **[cloudera@quickstart init.d]$** *sudo -u hdfs hadoop fs -chown mpatnam:mapred* /user/mpatnam  #verify HDFS dir created  **[cloudera@quickstart init.d]$** *sudo -u hdfs hadoop fs -ls -R /user/mpatnam* |

**Problem 4**. Take any simple ASCII file and copy of that file into a directory in HDFS.Delete the copy of the file on the host OS. Next cat the content of the file in HDFS and convince yourself that the file is the same as the one you originally had on the host OS.Then copy that file back to the host OS and examine its content. Show that the content is the same as the content of the original file. Finally, delete that file in the HDFS.[20%]

**Approach**: Performing this problem under the new user account (mpatnam) created as part of problem 3. Copying a local file /home/mpatnam/ascii.txt to HDFS folder: /user/mpatnam

|  |
| --- |
| **[mpatnam@quickstart ~]$** *pwd*  /home/mpatnam  **[mpatnam@quickstart ~]$** *ls -l \*.txt*  -rwx------ 1 mpatnam mpatnam 221 Feb 16 19:04 ascii.txt  **[mpatnam@quickstart ~]$** *cat ascii.txt*  "New England", "Massachusetts", "Boston", "SuperMart",  "Feb" , 2000000"New England", "Massachusetts", "Springfield", "SuperMart",  "Feb" , 1400000"New England", "Massachusetts", "Worcester", "SuperMart",  "Feb" , 2200000  # put the file into HDFS: /user/mpatnam  **[mpatnam@quickstart ~]$** *hadoop fs -put /home/mpatnam/ascii.txt*  # remove local file from host OS.  **[mpatnam@quickstart ~]$** *rm -f ascii.txt*  **[mpatnam@quickstart ~]$** *ls -l*  total 0  -rw-rw-r-- 1 mpatnam mpatnam 0 Feb 14 20:07 test  **[mpatnam@quickstart ~]$** *hadoop fs -ls -R /user/mpatnam*  -rw-r--r-- 1 mpatnam mapred 221 2017-02-16 19:10 /user/mpatnam/ascii.txt  # cat the file from hdfs location  **[mpatnam@quickstart ~]$** *hadoop fs -cat /user/mpatnam/ascii.txt*  "New England", "Massachusetts", "Boston", "SuperMart",  "Feb" , 2000000"New England", "Massachusetts", "Springfield", "SuperMart",  "Feb" , 1400000"New England", "Massachusetts", "Worcester", "SuperMart",  "Feb" , 2200000  # copy file back to host OS  **[mpatnam@quickstart ~]$** *hadoop fs -get /user/mpatnam/ascii.txt*  **[mpatnam@quickstart ~]$** *ls -l*  -rw-r--r-- 1 mpatnam mpatnam 221 Feb 16 19:19 ascii.txt  # cat the file from local file system  **[mpatnam@quickstart ~]$** *cat ascii.txt*  "New England", "Massachusetts", "Boston", "SuperMart",  "Feb" , 2000000"New England", "Massachusetts", "Springfield", "SuperMart",  "Feb" , 1400000"New England", "Massachusetts", "Worcester", "SuperMart",  "Feb" , 2200000  # delete the file from HDFS location  **[mpatnam@quickstart ~]$** *hadoop fs -rm /user/mpatnam/ascii.txt*  17/02/16 19:22:36 INFO fs.TrashPolicyDefault: Moved: 'hdfs://quickstart.cloudera:8020/user/mpatnam/ascii.txt' to trash at: hdfs://quickstart.cloudera:8020/user/mpatnam/.Trash/Current/user/mpatnam/ascii.txt  # verify that file is gone in HDFS  **[mpatnam@quickstart ~]$** *hadoop fs -ls -R /user/mpatnam/ascii.txt*  ls: `/user/mpatnam/ascii.txt': No such file or directory |

**Problem 5**. Copy attached 4300.txt file into HDFS. Find the number of lines in the text that contain word “afternoon” or “night” or “morning”. Do this in two ways, first create a lambda function which will test whether a line contains any one of those 3 words.Second, create a named function in the language of choice that returns TRUE if a line passed to it contains any one of those three words. Demonstrate that the count is the same.

**Approach**: 4300.txt is a file in the desktop. We will first copy it to the VM local file system using scp (secure copy) command. SCP tool is available as part of cygwin toolset installed on my windows 10 box. SCP also needs target hostname or IP address. We will use ifconfig command on linux VM to get this information. Finally once file is transferred to VM local filesystem, we can then copy to HDFS.

# On Cloudera Linux VM, get the IP address.

|  |
| --- |
| **[mpatnam@quickstart ~]$** *ifconfig*  eth1 Link encap:Ethernet HWaddr 00:0C:29:B0:B6:F9  inet addr:**192.168.89.129** Bcast:192.168.89.255 Mask:255.255.255.0  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  RX packets:19735 errors:0 dropped:0 overruns:0 frame:0  TX packets:10029 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:1000  RX bytes:25470381 (24.2 MiB) TX bytes:852815 (832.8 KiB) |

# On Desktop, running following command from cygwin to copy file to VM filesystem.

|  |
| --- |
| **mpatnam@Mohan** /cygdrive/c/Users/mpatnam/Documents  $ *ls -al 4300.txt*  -rwx------+ 1 mpatnam mpatnam 4692498 Feb 12 21:48 4300.txt  # Obtained IP address of the VM by running  **mpatnam@Mohan** /cygdrive/c/Users/mpatnam/Documents  $ *scp 4300.txt mpatnam@192.168.89.129:/home/mpatnam*  mpatnam@192.168.89.129's password:  4300.txt 100% 4583KB 37.9MB/s 00:00 |

# Transfer file from VM local filesystem to HDFS

|  |
| --- |
| **[mpatnam@quickstart ~]$** pwd  /home/mpatnam  **[mpatnam@quickstart ~]$** ls -l  total 4592  -rwx------ 1 mpatnam mpatnam 4692498 Feb 16 19:33 4300.txt  -rw-r--r-- 1 mpatnam mpatnam 221 Feb 16 19:19 ascii.txt  -rw-rw-r-- 1 mpatnam mpatnam 221 Feb 16 19:28 test  **[mpatnam@quickstart ~]$** hadoop fs -put /home/mpatnam/4300.txt  **[mpatnam@quickstart ~]$** hadoop fs -ls -R /user/mpatnam/4300.txt  -rw-r--r-- 1 mpatnam mapred 4692498 2017-02-16 19:33 /user/mpatnam/4300.txt |

# Now that file is in HDFS, lets use pyspark to look for specific works using named function.

|  |
| --- |
| >>>def hasWords(line):  … return “afternoon” in line or “night” in line or “morning” in line  …  >>> lines = sc.textFile(“file:///user/mpatnam/4300.txt”)  >>> wordLines = lines.filter(hasWords)  >>> wordLines.count()  1251 |

# repeating the same using lambda function

|  |
| --- |
| >>> lines = sc.textFile(“file:///user/mpatnam/4300.txt”)  >>> wordLines = lines.filter(**lambda** line: “afternoon” in line or “night” in line or “morning” in line)  >>> wordLines.count()  1251 |