Guideline to Using Hadoop on CSE Department's Hadoop Cluster

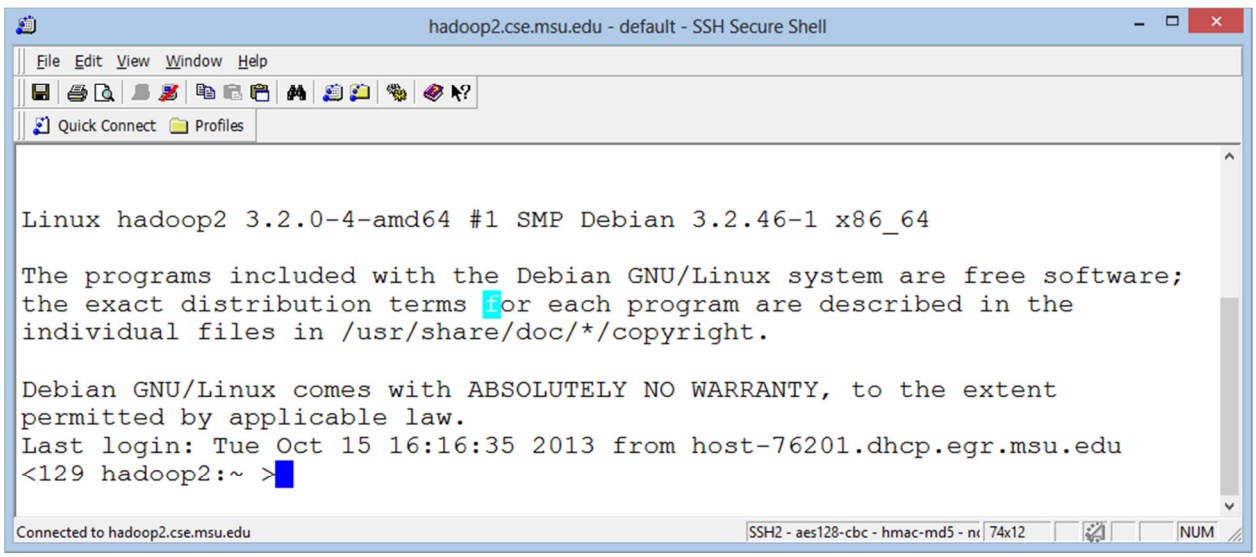
This documentation provides a step-by-step guideline on how to use the Hadoop cluster in the CSE department. This document is divided into 2 parts:

PART 1: Getting Started

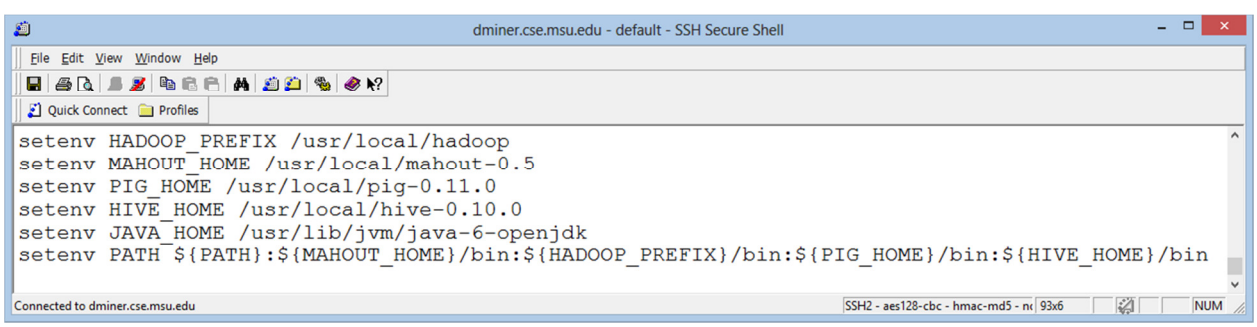
PART 2: Running a Hadoop Program

PART 1: GETTING STARTED

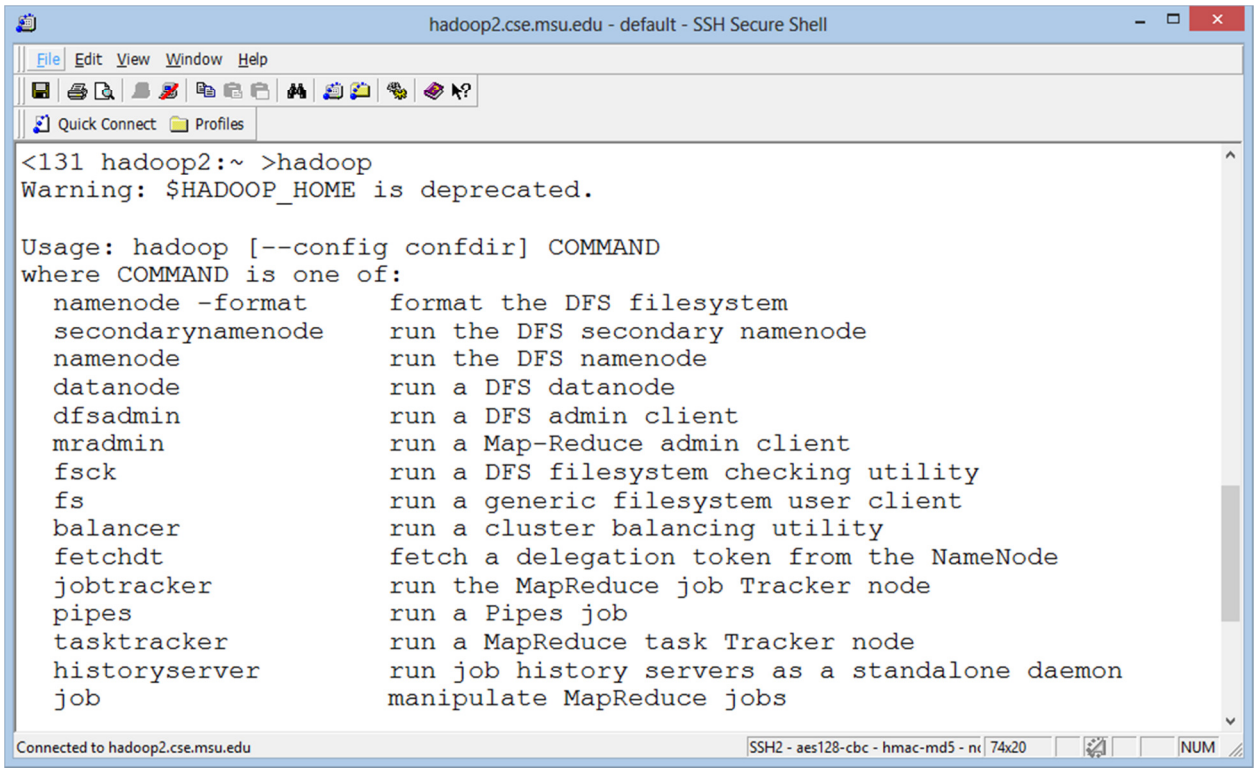
STEP 1: Open an SSH connection to hadoop1.cse.msu.edu or hadoop2.cse.msu.edu. You should be able to login using your CSE account. The disk drives are NSF-mounted, so you're basically logged in to your CSE home directory.



STEP 2: Try typing hadoop on the Linux command line. If it doesn't work, you'll need to setup the environment variables for PATH, HADOOP_PREFIX, MAHOUT_HOME, PIG_HOME, JAVA_HOME, and HIVE_HOME. You can set it up in your profile to make sure these variables are set each time you logged in to the cluster. For tcsh shell, you can modify the variables by adding the following lines to your .personal file in your home directory:



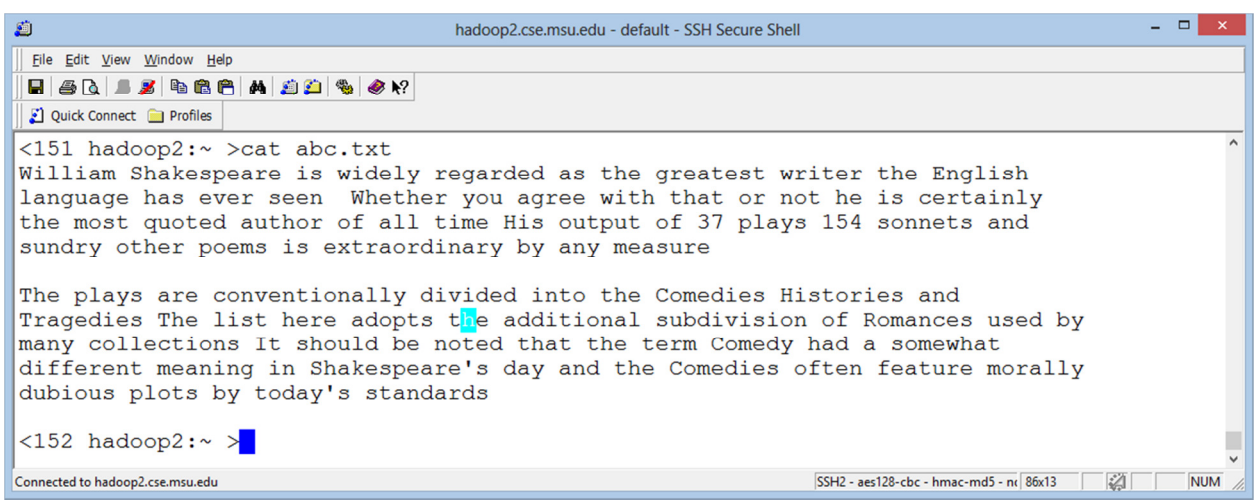
STEP 3: Now you're ready to run hadoop. Type hadoop on command line and it will display a list of commands you can use.



PART 2: RUNNING A HADOOP PROGRAM

The hadoop installation comes with example programs and datasets. In this example, we will illustrate how to run the Hadoop program for counting words in a collection of documents.

STEP 1: In this example, you will count the number of times each word appears in a given input file named abc.txt. You can replace this with any files that you want.

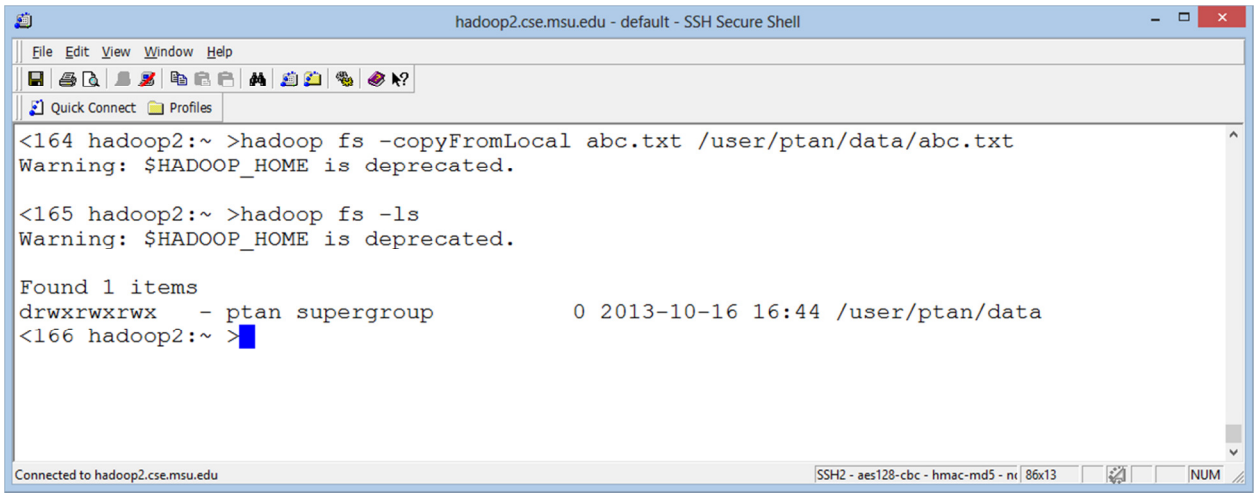


STEP 2: Upload the data from your local directory to Hadoop Distributed File System (HDFS). If you're using HDFS for the first time, make sure you type the full path of the destination directory. The command for copying the file is:

> hadoop fs –copyFromLocal <source> <destination>

In the example below, the file abc.txt will be uploaded to HDFS and stored in the directory named /user/ptan/data. To make sure the file is copied correctly, you can use the *ls* command to list the content of your Hadoop working directory

> hadoop fs –ls



STEP 3: We will use a sample Hadoop program called wordcount to do this. The Java program is archived in a file called hadoop-examples-1.1.1.jar located in /usr/local/hadoop directory. This program expects two input arguments: (1) source directory (in HDFS) that contains the input data, and (2) destination directory (in HDFS) that contains the results. The typical syntax for executing a Hadoop java program (stored in a jar file) is as follows:

> hadoop jar <name-of-jar-file> <name-of-java-program> <input arguments>

In the example shown below, the wordcount program will read input documents stored in /user/ptan/data directory and write the output to /user/ptan/output directory.

```
<167 hadoop2:~ >hadoop jar ${HADOOP_HOME}/hadoop-examples-1.1.1.jar wordcount /user/pt
an/data /user/ptan/output
Warning: $HADOOP_HOME is deprecated.

13/10/16 16:46:29 INFO input.FileInputFormat: Total input paths to process : 1
13/10/16 16:46:29 INFO util.NativeCodeLoader: Loaded the native-hadoop library
13/10/16 16:46:29 WARN snappy.LoadSnappy: Snappy native library not loaded
13/10/16 16:46:30 INFO mapred.JobClient: Running job: job_201308211055_0058
13/10/16 16:46:31 INFO mapred.JobClient:  map 0% reduce 0%
13/10/16 16:46:36 INFO mapred.JobClient:  map 100% reduce 0%
13/10/16 16:46:44 INFO mapred.JobClient:  map 100% reduce 33%
13/10/16 16:46:45 INFO mapred.JobClient:  map 100% reduce 100%
13/10/16 16:46:46 INFO mapred.JobClient: Job complete: job_201308211055_0058
13/10/16 16:46:46 INFO mapred.JobClient: Counters: 29
13/10/16 16:46:46 INFO mapred.JobClient:   Job Counters
13/10/16 16:46:46 INFO mapred.JobClient:     Launched reduce tasks=12
13/10/16 16:46:46 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=5380
13/10/16 16:46:46 INFO mapred.JobClient:     Total time spent by all reduces waiting a
fter reserving slots (ms)=0
13/10/16 16:46:46 INFO mapred.JobClient:     Total time spent by all maps waiting afte
r reserving slots (ms)=0
```

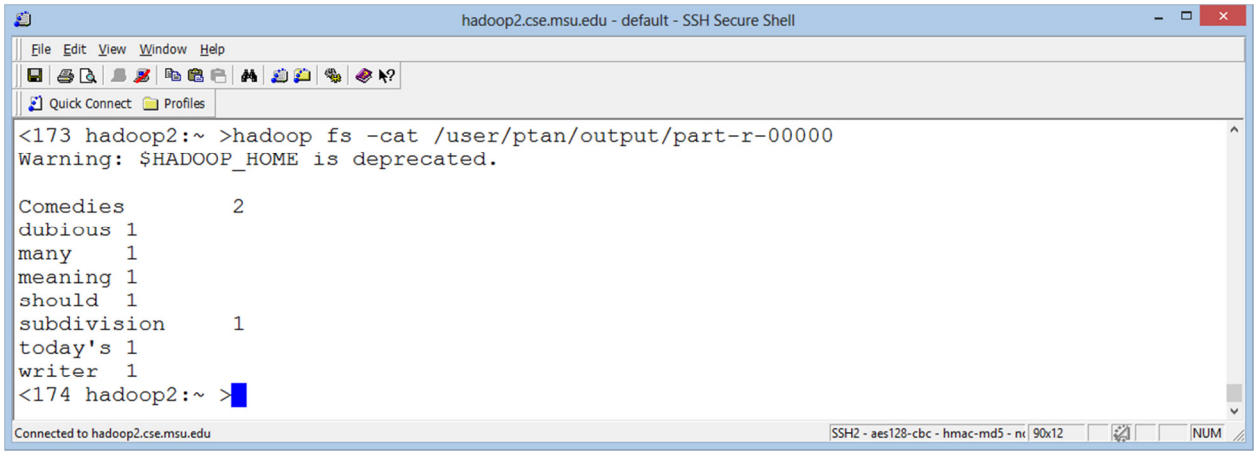
STEP 4: To verify that the job has been completed, list the content of /user/ptan/output directory and check whether there is a file named _SUCCESS. If it is successful, the results will be stored in the files called part-r-XXXXX (where X is a digit from 0 to 9). The number of output files generated depends on the number of reducers used to run the Hadoop job.

```
<171 hadoop2:~ >hadoop fs -ls /user/ptan/output
Warning: $HADOOP_HOME is deprecated.

Found 14 items
-rw-r--r--   2 ptan supergroup          0 2013-10-16 16:46 /user/ptan/output/_SUCCESS
drwxrwxrwx   - ptan supergroup          0 2013-10-16 16:46 /user/ptan/output/_logs
-rw-r--r--   2 ptan supergroup         80 2013-10-16 16:46 /user/ptan/output/part-r-00000
-rw-r--r--   2 ptan supergroup         87 2013-10-16 16:46 /user/ptan/output/part-r-00001
-rw-r--r--   2 ptan supergroup         85 2013-10-16 16:46 /user/ptan/output/part-r-00002
-rw-r--r--   2 ptan supergroup         52 2013-10-16 16:46 /user/ptan/output/part-r-00003
-rw-r--r--   2 ptan supergroup         68 2013-10-16 16:46 /user/ptan/output/part-r-00004
-rw-r--r--   2 ptan supergroup         52 2013-10-16 16:46 /user/ptan/output/part-r-00005
-rw-r--r--   2 ptan supergroup         29 2013-10-16 16:46 /user/ptan/output/part-r-00006
-rw-r--r--   2 ptan supergroup         63 2013-10-16 16:46 /user/ptan/output/part-r-00007
-rw-r--r--   2 ptan supergroup         76 2013-10-16 16:46 /user/ptan/output/part-r-00008
-rw-r--r--   2 ptan supergroup         64 2013-10-16 16:46 /user/ptan/output/part-r-00009
-rw-r--r--   2 ptan supergroup         20 2013-10-16 16:46 /user/ptan/output/part-r-00010
-rw-r--r--   2 ptan supergroup         17 2013-10-16 16:46 /user/ptan/output/part-r-00011
<172 hadoop2:~ >
```

STEP 5: You can view the content of a file by using the following command

> hadoop fs –cat <filename>



STEP 6: You can merge the outputs of the program and write the output back to your local directory using the getmerge command.

> hadoop fs –getmerge <source-directory> <destination-local-filename>

For example, the following command will combine all the part-r-XXXXX files in /user/ptan/output directory and store them into a single file called output.