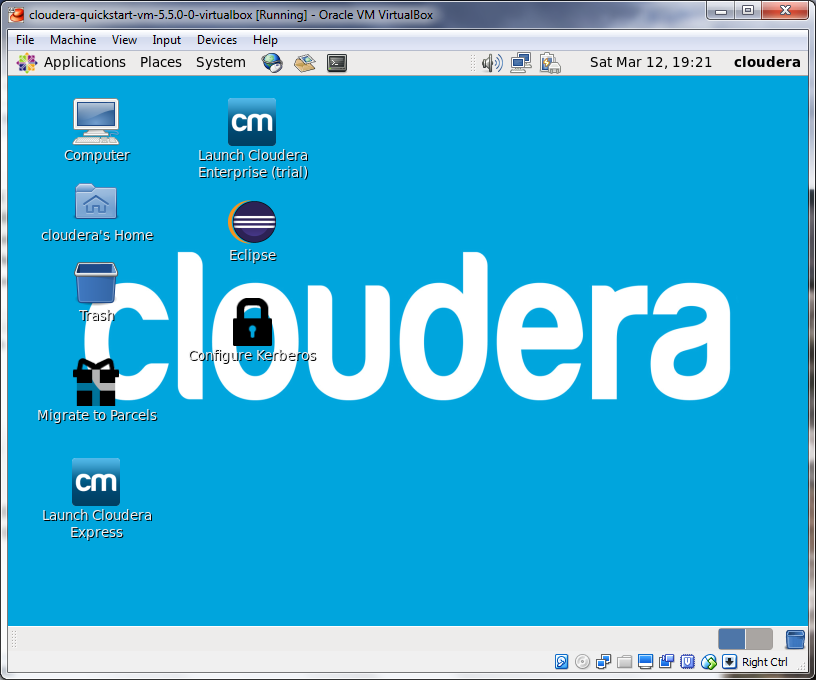
**Big Data Management Session 01 Lab**

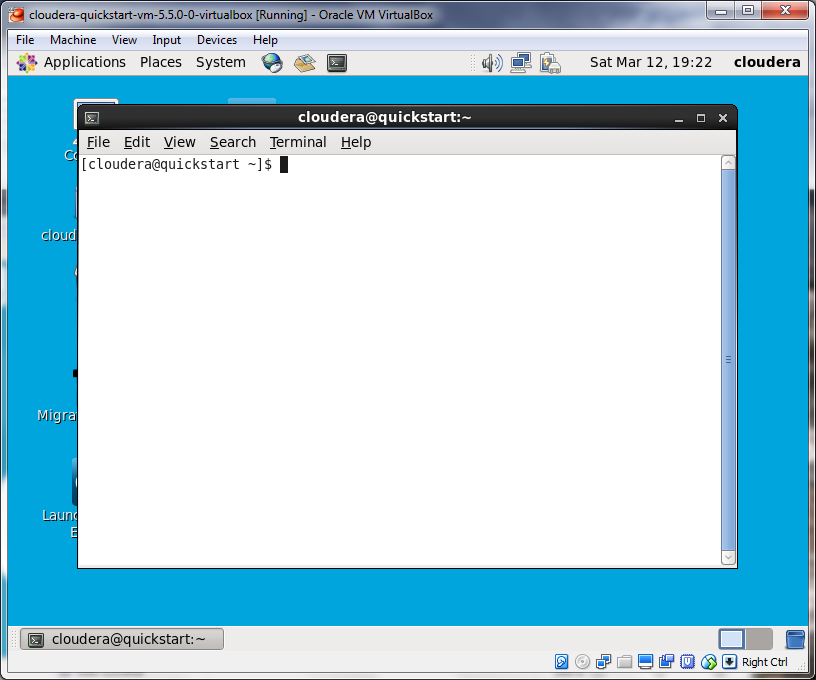
The hands-on exercise for today is a brief introduction to working in a Unix environment, especially with the command line. I will use the Python programming language a little bit, but I will not spend much time teaching Python. If you want to learn more about Python, there are many lessons and tutorials online for free. (Many of my students recommend *Learn Python the Hard Way*, which is found at <http://learnpythonthehardway.org/>.)

When you launch Cloudera Quick Start virtual machine for the first time, close the Firefox browser and you will see the desktop which looks like this:



This is a simulated Linux computer, which we can use as a controlled environment to experiment with Python, Hadoop, and some other big data tools. The platform is CentOS Linux, which is related to Red Hat Linux. I chose this VM because the author of our textbook works for Cloudera; therefore, his examples will probably work in this VM. Cloudera is one major distributor of Hadoop. Others include Hortonworks, MapR, IBM, and Teradata.

Open a command-line terminal from the menu: Accessories 🡪 System Tools 🡪 Terminal



At the command prompt, you can enter Unix commands and hit ENTER to execute them, one at a time. The prompt itself may give you some helpful information. In this case, it shows you the name of the user (cloudera), the name of the computer (quickstart), and the directory where you’re working (“~” which is an abbreviation for your home directory). Try these commands:

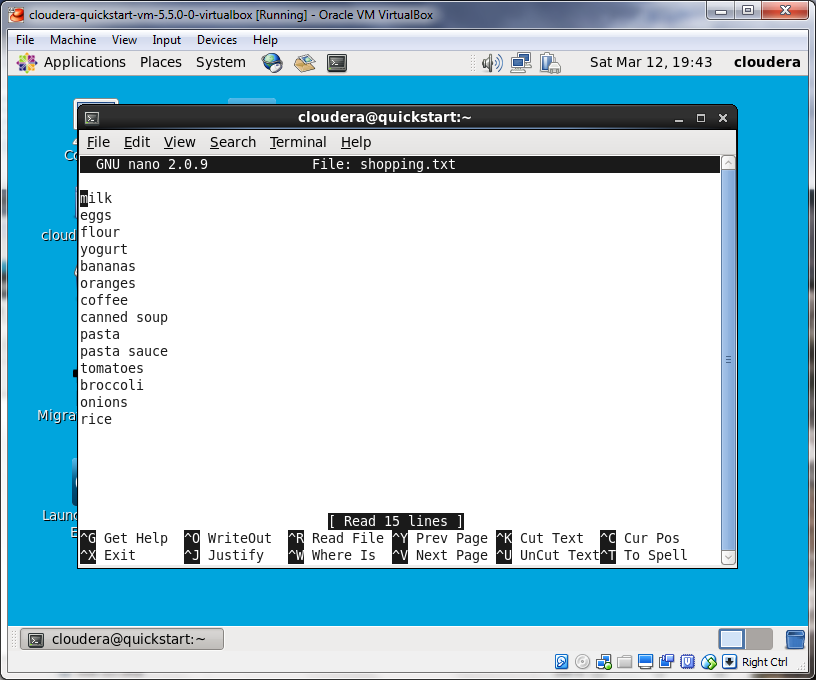
|  |  |
| --- | --- |
| echo hello world | Print “hello world” to the standard output. |
| pwd | Print the working directory. We’re in “~” which we know is the user’s home directory, but this will give you the full address: /home/cloudera |
| ls | List files and folders in the working directory. |
| ls -l | The same, but with details. |
| ls -a | The same, but with hidden files included. |
| ls -al | The same, but with details *and* hidden files. |
| ls / | List files in the top level directory of the file system. |
| cd / | Change directory to “/”. |
| cd home  cd cloudera | From the top level, cd into the “home” directory and then the “cloudera” subdirectory. These are *relative* moves because you didn’t enter the full address of either directory. |
| mkdir testdir | Make a directory called “testdir” within the current working directory. |
| cd /home/cloudera/testdir | Move into testdir by providing the *absolute* location of it. You could also have just used “cd testdir”. |
| cd ~ | Move back to the home directory. |

Next, let’s write and read some text files. There’s a GUI text editor provided with the VM, found under the Applications menu: Applications 🡪 Accessories 🡪 gedit Text Editor. Try it now. Write something with several lines. I’m going to write a shopping list. Save it as “shopping.txt” in the “cloudera” directory. Remember, this is the home directory or “~”.

Let’s also try a way to edit text right from the terminal. There’s a built-in text editor called “nano” which is pretty easy to use. Type:

nano shopping.txt

nano works pretty much like any text editor, except that you don’t use the mouse. The commands you can use are shown at the bottom of the screen. Use CTRL+O to write out (save) the file, and CTRL+X to quit nano.



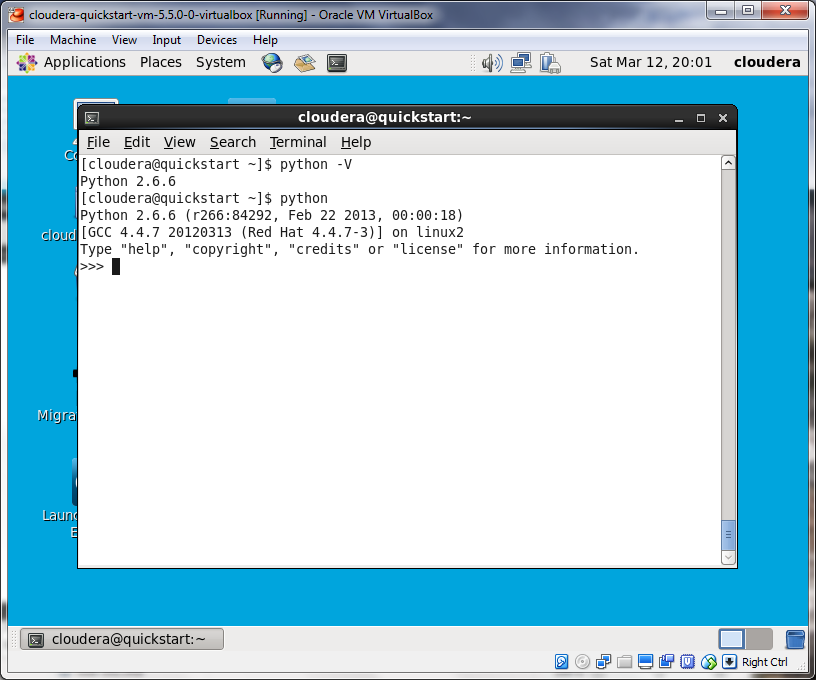
Try the following:

|  |  |
| --- | --- |
| mv shopping.txt list.txt | Rename the file to “list.txt”. “mv” is the *move* command. |
| mv list.txt testdir/ | Move list.txt into the testdir directory. |
| cd testdir | Switch to the testdir directory. |
| cat list.txt | “cat” reads the file and sends it to standard output. It is printed to the screen. |
| sort list.txt | “sort” sorts the lines in the file alphabetically and outputs them to the screen. |
| cat list.txt | sort | The “|” or “pipe” character takes the output of one program (cat) and sends it as input to another program (sort). This is a very useful tool when creating a data pipeline! You will see the output of the sort program applied to the data in list.txt. |
| sort --help | This is a way to get more information about how to use a Unix program. Notice that the output is more than you can see on one screen. |
| sort --help | less | Pipe the “help” output into less, a program that allows you to page up and page down through a large body of text. Hit “q” when done reading. |
| cat list.txt | sort > sortedlist.txt | The “>” character takes the output of a program (or in this case, the output of our pipeline) and saves it in a file. If you use “ls” you’ll see that a new file has been created in this folder. |
| cat sortedlist.txt | This will demonstrate that the new file contains the output of cat | sort. |
| rm sortedlist.txt | Remove (delete) the new file. |
| cd .. | Change to the parent directory. |
| rm -rf testdir | Remove the ‘testdir’ directory. “-rf” means “recursive” and “force”: delete the directory itself *and* also everything within it *without* asking us about every individual file. |

Depending on time, I’ll now demonstrate the Python interpreter. A version of Python is already installed in this virtual machine. Which version? Type the following to find out:

python -V

It’s Python 2.6.6 for me. Type ‘python’ by itself to open the interactive interpreter.

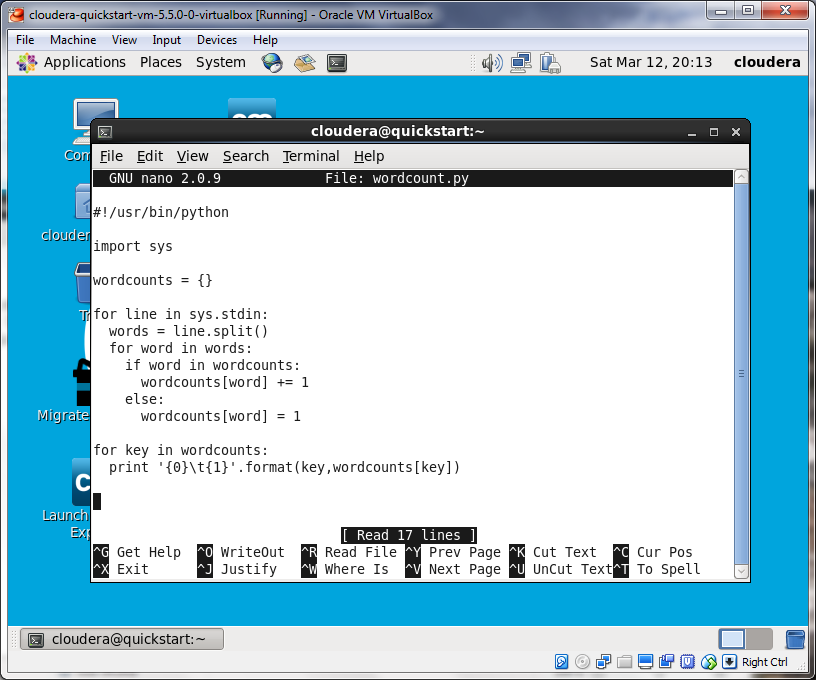


Here you can type one-line Python commands and immediately get the results. Try:

print ("hello world")

Type quit() when you want to exit the Python interpreter.

The other main way to use Python is to write code and save it in a text file with a “.py” suffix. Here’s a little program that will count the words in a text file:



After writing and saving this file as “wordcount.py”, do the following to make it an executable program:

chmod a+x wordcount.py

Now, you’ll need a file of text. You can make one quickly by dumping the “help” text of a program you’re interested in:

hadoop --help > hadoophelp.txt

To pipe your file into the word count program, use the following:

cat hadoophelp.txt | ./wordcount.py

The “./” is necessary here. It means that wordcount.py is located in your current working directory.

Now make a slightly longer pipeline so that you can sort it and read it all on one screen at a time:

cat hadoophelp.txt | ./wordcount.py | sort | less