Hands-on Lab: Wrangling Text Files at the Command Line



Estimated time needed: 40 minutes

Consider that any text file can potentially be interpreted as some sort of dataset. For example, your text file might contain the following:

- The kind of text that you might find in a news article or blog post
- The source HTML code that defines a web page
- A table-like structure consisting of text fields separated by a delimiter such as a comma or Tab
- The data stream that encodes a song or a movie
- A completely random sequence of characters, such as:
 - o an encryption key or password
 - o a digitized random noise sequence

Whatever your text file may be, you can benefit from being able to perform some basic text processing to munge, wrangle, clean, and integrate your data the way you need it.

Learning Objectives

Working through the exercises in this lab will enable you to perform some basic but essential text wrangling operations. These operations will allow you to work with text files by:

- Displaying and exploring file contents
- Extracting and displaying first or last N lines of text
- Displaying counts of lines, words, and characters in text
- Sorting lines (rows) of text
- Dropping consecutively duplicated lines of text
- Extracting lines of text containing a pattern match
- · Extracting fields from lines of text
- · Merging text files as aligned columns of text

These are some of the building blocks of filtering text files. Later in this course, you will learn how to combine such operations. This will empower you to start engineering sophisticated views of your data by creating complex data-processing flows called *data pipelines*.

About Skills Network Cloud IDE

Skills Network Cloud IDE (based on Theia and Docker) provides an environment for hands-on labs for course and project related labs. Theia is an open source IDE (Integrated Development Environment), that can be run on desktop or on the cloud. To complete this lab, you will be using the Cloud IDE based on Theia.

Important notice about this lab environment

Please be aware that sessions for this lab environment are not persisted. Thus, every time you connect to this lab, a new environment is created for you and any data or files you may have saved in a previous session will be lost. To avoid losing your data, plan to complete these labs in a single session.

Exercise 1 - Viewing file contents

cat more

In this exercise, you will learn how to explore file contents using the cat, more, and less commands to display the file contents in your terminal window.

Begin by changing directories to your default home directory, ~, or \home\theia:

cd ~

Using the 1s command, you should see a file called entrypoint.sh. The .sh is a convention used to identify a text file as being a shell script.

Next, let's take a look inside this file.

about:blank 1/8

1.1. Viewing file content with the cat command

The cat command displays the contents of the file and exits back to the command prompt as follows:

cat entrypoint.sh

It only displays the tail end of the file, so if the file is too long to fit on the terminal, you won't be able to see some of its contents.

Although the cat command may not be the best way to view the contents of a file, especially larger files, it is quite useful for shell scripting applications. For example, it is often used to *concatenate*, or append one file onto another.

1.2. Viewing file content with the more command

A better alternative to the cat command for viewing file contents is the more command. By entering the following command:

more entrypoint.sh

you will see the top portion of the file first.

Tip: The first line of this particular file, #!/bin/bash, is called a *shebang*. Basically, this shebang line makes the file a *bash script* by invoking the bash shell. You will learn more about shebang lines later in this course.

When using the more command, you can see only as many lines as will fit on your terminal window at once.

To see the next portion of the file, just press your spacebar. You can keep *paging* this way, tapping the spacebar until you reach the end of the file. Once you reach the last *page*, you will exit back to the command prompt.

Another way to exit is simply to type q, which quits and returns to the command prompt.

1.3. Scrolling through file content with the less command

What if you want to move up and down through the file, not just downward? In this case, you can use the 1ess command:

less entrypoint.sh

Just like more, the less command displays the first page of the file. What's useful about less is that you can use it to move around the file, page by page, using the Page Up and Page Down keys.

You can also scroll up and down through the file line-by-line, using the Up Arrow and Down Arrow keys, ↑ and ↓.

Unlike more, less does not automatically exit when you reach the end of a file, allowing you the option to continue scrolling around. You can quit at any time by typing q.

Exercise 2 - Viewing text file contents

In this exercise, you will work with a few more commands for viewing the content of text files.

To begin, run the following commands:

```
cd /home/project
wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/usdoi.txt
```

about:blank 2/8

The wget command downloads a text file called usdoi.txt from the provided URL. You'll see this command again later in the context of networking commands. You can check to see if you successfully downloaded the usdoi.txt by using the 1s command.

2.1. Display the first N lines of a file

head

By default, head will print the first 10 lines of a file. To use it with usdoi.txt, enter the following:

head usdoi.txt

You can also specify the number of lines to be printed. Print only the first 3 lines of text from the file usdoi.txt by entering:

head -3 usdoi.txt

2.2. Display the last N lines of a file

tai]

By default, tail will print the last 10 lines of the file usdoi.txt:

tail usdoi.txt

Just like with head, you can specify the number of lines to be printed. Print the last 2 lines of the file usdoi.txt by entering the following:

tail -2 usdoi.txt

Exercise 3 - Getting basic text file stats

3.1. Count lines, words, or characters from a text file

wc

If you want to find the number of lines, words, and characters in a file like usdoi.txt, enter the following command:

about:blank 3/8



The output contains the number of lines, followed by the number of words, followed by the number of characters in the file.

To get just the count of lines in usdoi.txt, use the -1 option:

```
wc -l usdoi.txt
```

Similarly, for the count of words in usdoi.txt, use the -w option:

```
wc -w usdoi.txt
```

To print the number of characters in usdoi.txt, use the -c option:

wc -c usdoi.txt

Exercise 4 - Basic text wrangling: sorting lines and dropping duplicates

4.1. Sort and display lines of file alphanumerically

sort

You can use the sort command to display the lines of a file sorted alphanumerically.

To view the lines of usdoi.txt sorted alphanumerically, enter:

sort usdoi.txt

To view those lines sorted in reverse order, enter:

```
sort -r usdoi.txt
```

about:blank 4/8

4.2.	Drop	consecutive	duplicated	lines and	display	result

uniq

First download the following file:

wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/zoo.txt

View the raw contents of zoo.txt with the cat command:

cat zoo.txt

View the contents of zoo.txt with identical, consecutive lines dropped using the uniq command:

uniq zoo.txt

The uniq line will drop any lines in the file that are identical *and* consecutive. This is similar to what is known as "dropping duplicates". As you can see from this example, however, there can still be duplicated lines left over if these lines are not repeated right after the other.

Exercise 5 - Basic text wrangling: extracting lines and fields

5.1. Extract lines matching a specified criterion

grep

The grep command allows you to specify a pattern and search for lines within a file that match that pattern.

For example, the following command prints all lines in the file usdoi.txt which contain the word people:

grep people usdoi.txt

Some frequently used options for grep include:

Option	Description	
-n	Along with the matching lines, also print the line numbers	

about:blank 5/8

Option	Description	
-c	Get the count of matching lines	
-i	Ignore the case of the text while matching	
-v	Print all lines which do not contain the pattern	
-w	Match only if the pattern matches whole words	

You can use these options to print all the lines from the /etc/passwd file which do not contain the pattern login:

grep -v login /etc/passwd

5.2. Extract fields from lines of text

cut

The cut command allows you to view only specific fields from each line of text in a file.

For example, you can use cut with the -c option to view only the *first two* characters of each line:

cut -c -2 zoo.txt

Or to view each line *starting from* the second character:

cut -c 2- zoo.txt

The cut command can also be used to extract a field from a delimited file.

To demonstrate this, start by downloading and taking a look at the following comma-separated file:

wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/v4_new_content/labs/names_and_numbers.csv cat names_and_numbers.csv

Now you can extract just the phone numbers for each person listed in the file using the -d (delimiter) and f (field) options as follows:

cut -d "," -f2 names_and_numbers.csv

about:blank 6/8

-d "," tells the command that the delimiter is a comma, and -f2 tells it to extract the second field.

Exercise 6 - Basic text wrangling: merging lines as fields

6.1. Merge text files line-by-line, aligned as columns

paste

Use the paste command to merge lines of multiple files together.

Download the following file:

wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/zoo_ages.txt

Then use the paste command to view the two files merged together, line-by-line, as columns delimited by a Tab character:

paste zoo.txt zoo_ages.txt

Try changing the delimiter. Instead of the default Tab delimiter, you can specify a comma, as follows:

paste -d "," zoo.txt zoo_ages.txt

Practice Exercises

Before you begin, ensure you're working in your home directory by entering:

cd ~

- 1. Display the number of lines in the /etc/passwd file.
- ► Click here for Hint
- ► Click here for Solution
- $\textbf{2. Display the lines that contain the string "not installed" in $$/\text{var/log/bootstrap.log.}$}$
- ▶ Click here for Hint
- ► Click here for Solution

about:blank 7/8

3. The text file at https://cf-courses-data.sa.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/top-sites.txt contains a list of popular websites. Find all the websites on the list that have the word "org" in them.

- ▶ Click here for Hint
- ► Click here for Solution
- ▶ Alternative Solution
- 4. Print the first seven lines of top-sites.txt.
- ▶ Click here for Hint
- ► Click here for Solution
- 5. Print the last seven lines of top-sites.txt.
- ▶ Click here for Hint
- ► Click here for Solution
- 6. Print the first three characters of each line from top-sites.txt.
- ► Click here for Hint
- ▶ Click here for Solution
- $7.\ Extract\ and\ view\ only\ the\ names, without\ their\ phone\ numbers, from\ the\ file\ names_and_numbers.csv.$
- ▶ Click here for Hint
- ► Click here for Solution

Summary

In this lab, you learned how to:

- View file contents with cat, more, and less
- See the first and last N lines of a file using head and tail
- Find the number of lines, words, and characters in a file with wc
- Sort lines and drop duplicates using sort and uniq
- Extract lines and fields from a file with grep and cut
- Merge text files using paste

Authors

Jeff Grossman Ramesh Sannareddy Sam Prokopchuk

Other contributors

Rav Ahuja

© IBM Corporation. All rights reserved.

about:blank 8/8