CompTIA Linux+

Using Shell Input and Output Redirections

Exercise 1 - Using Shell Input and Output Redirections

Exercise 1 - Using Shell Input and Output Redirections

When commands are executed on a Linux system, it consists of three different parts referred to as streams:

- Standard input (stdin): This is the source of the data, which by default, is the text entered using a keyboard.
- Standard output (stdout): After the command is executed, it is the text that is displayed on the screen.
- Standard error (stderr): The error message that is displayed after a command has been executed.

In this exercise, you will understand how to redirect the Input and Output of these streams to files using a CentOS operating system.

Learning Outcomes

After completing this exercise, you will be able to:

- Log into a Linux system
- Redirect Output
- Redirect Input
- Discard the Output
- Use the tee Command

Your Devices

You will be using the following device in this lab. Please power these on now.

- PLABLINUX01 (CentOS Server)

Task 1 - Redirect Output

The default output of a command is displayed on the monitor. However, the user can also choose to redirect the output to a file.

For example, the user may need the entire directory listing to be embedded into another file. Rather than manually typing the entire directory listing, the user can simply use the directory listing command and then redirect the output to a file. The method of redirecting the output to a file is known as output redirection.

To redirect output, perform the following steps:

Step 1

On the desktop, right-click and select Open Terminal.

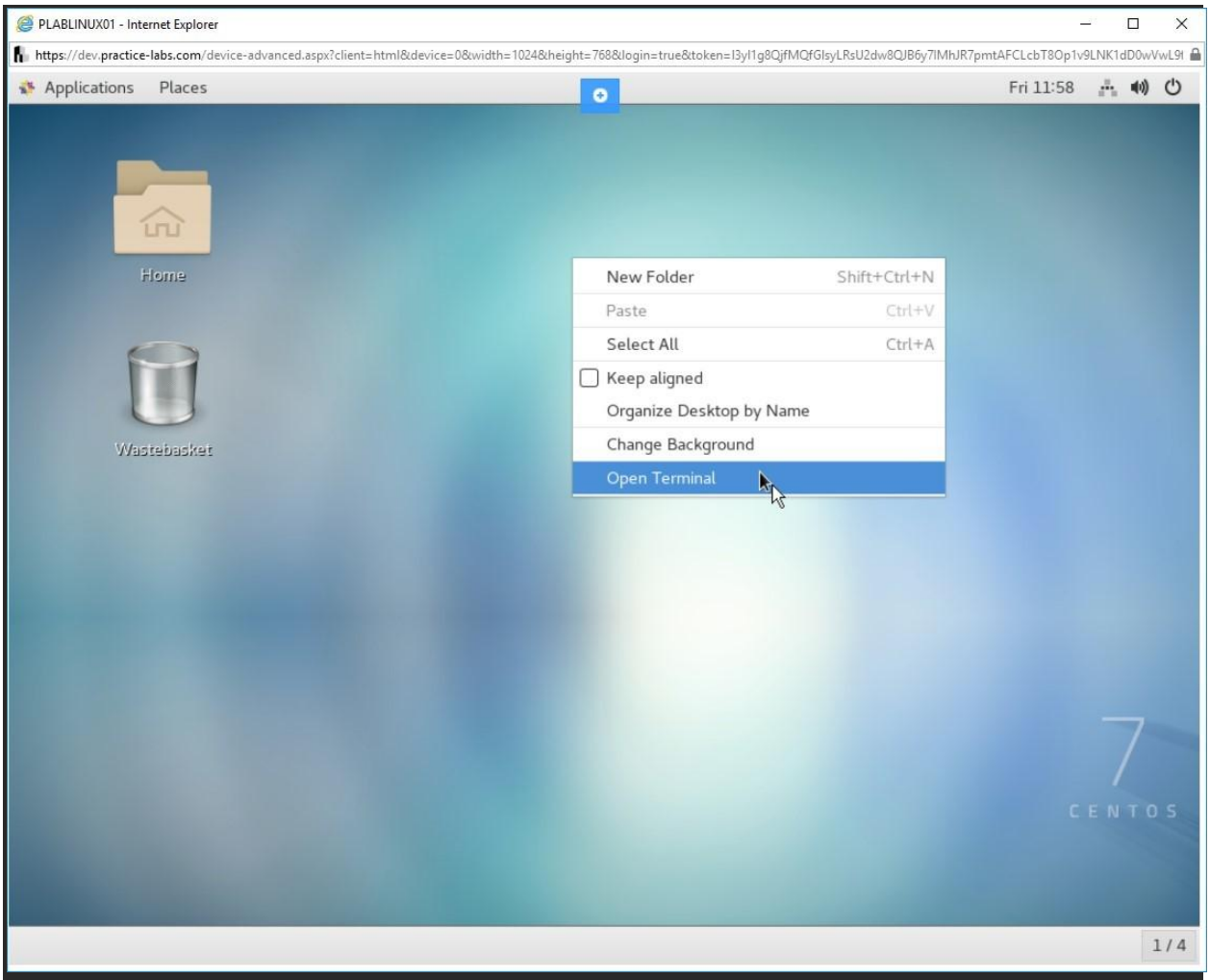

Figure 1.1 Screenshot of PLABLINUX01: Selecting the Open Terminal option from the context menu.

Step 2

The terminal window is displayed. The output of a command is redirected to a file with the help of the > operator. Type the following command:

```
who > user
```

Press Enter. After the user redirects the output, there is no output displayed on the monitor. Rather, the output is now saved into a file named user.

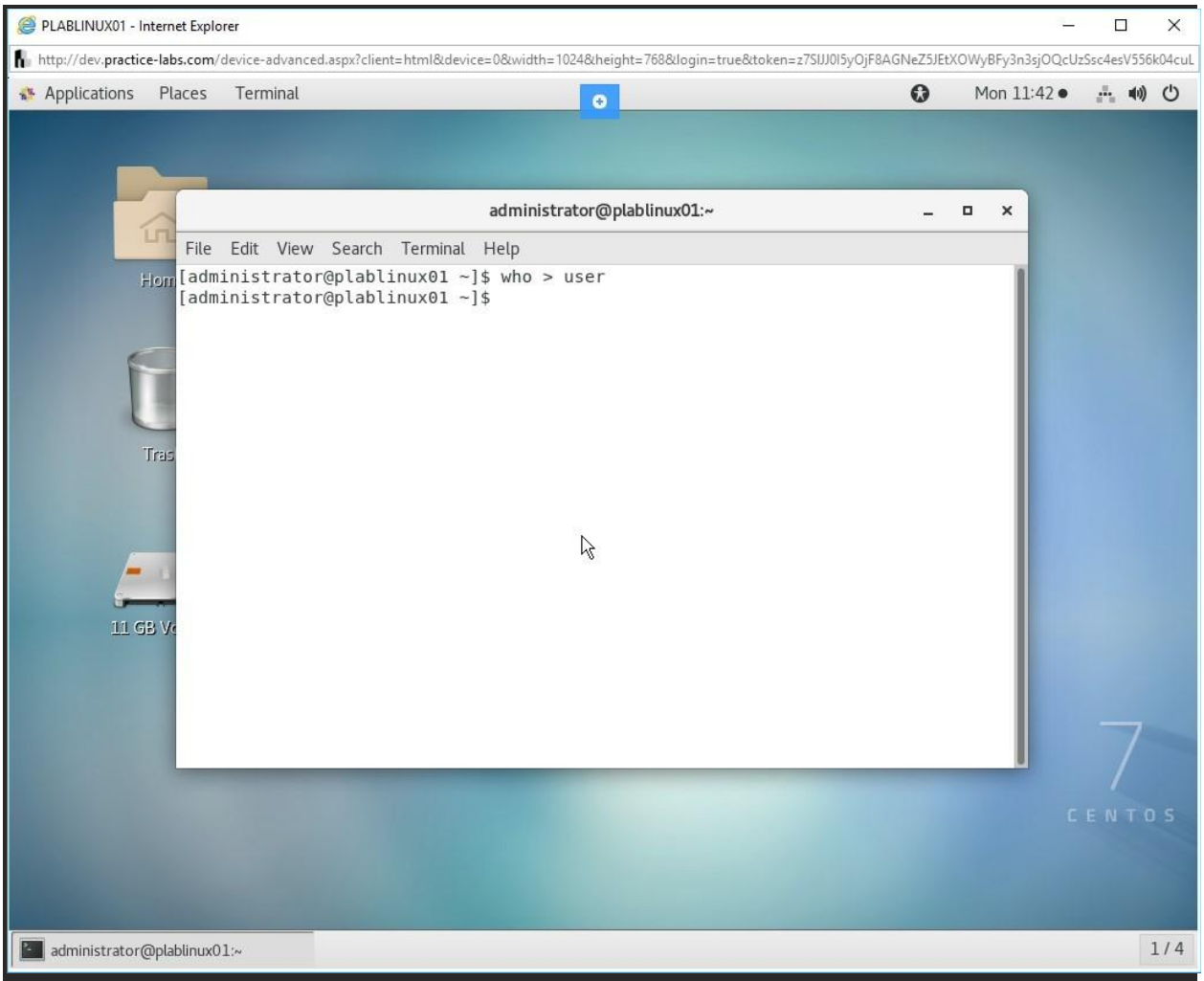

Figure 1.2 Screenshot of PLABLINUX01: Redirecting the output of the who command to a file named user.

Step 3

With the use of the output redirection, a new file named user is created. You can verify the command output by viewing the file using the cat command along with the file name.

Type the following command:

```
cat user
```

Press Enter.

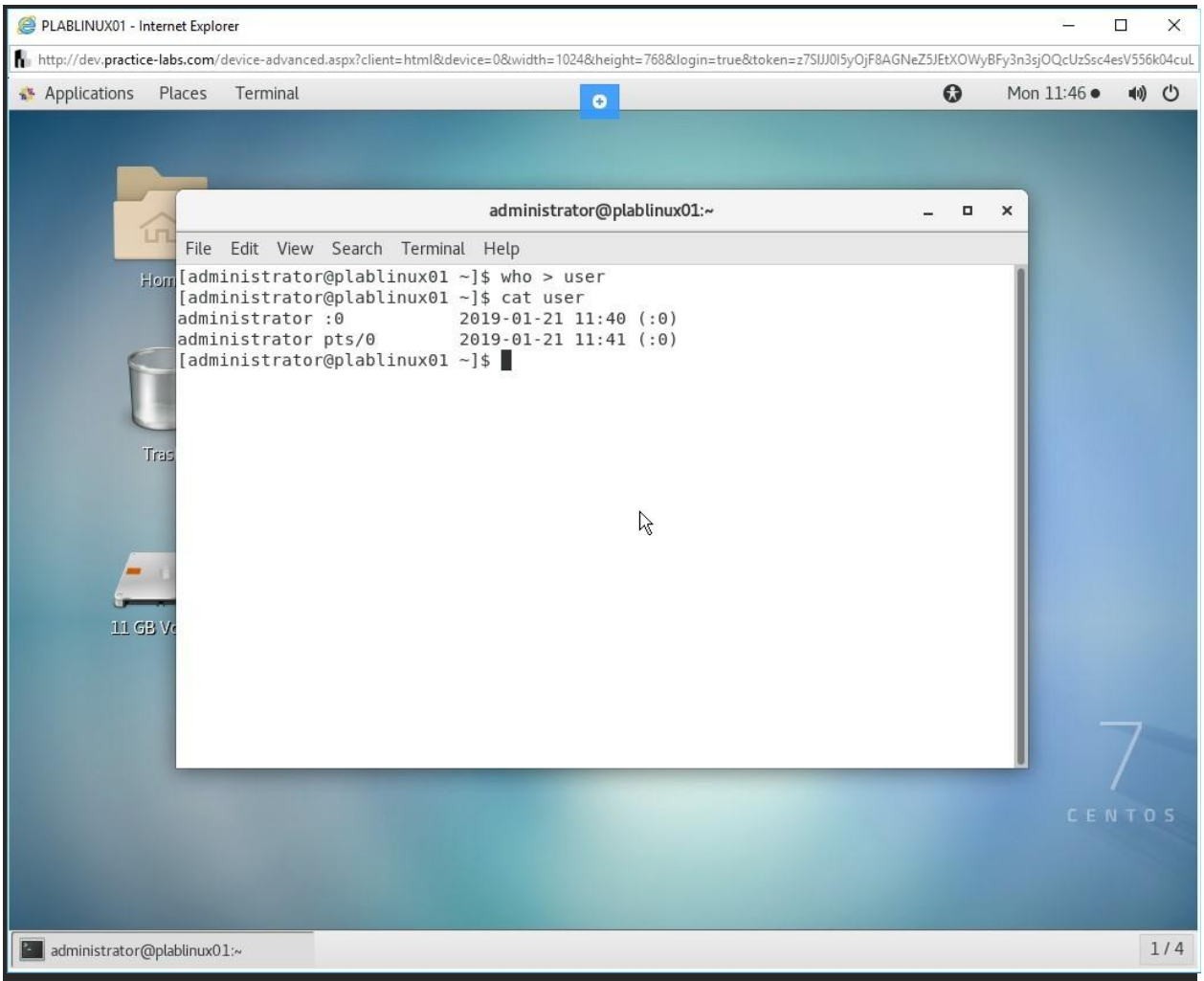

Figure 1.3 Screenshot of PLABLINUX01: Verifying the output of the user file.

Step 4

Clear the screen by entering the following command:

```
clear
```

You can redirect output to a file as many times as you want. However, with the > operator, the file will be overwritten every time.

Type the following command:

```
whoami > user
```

Press Enter.

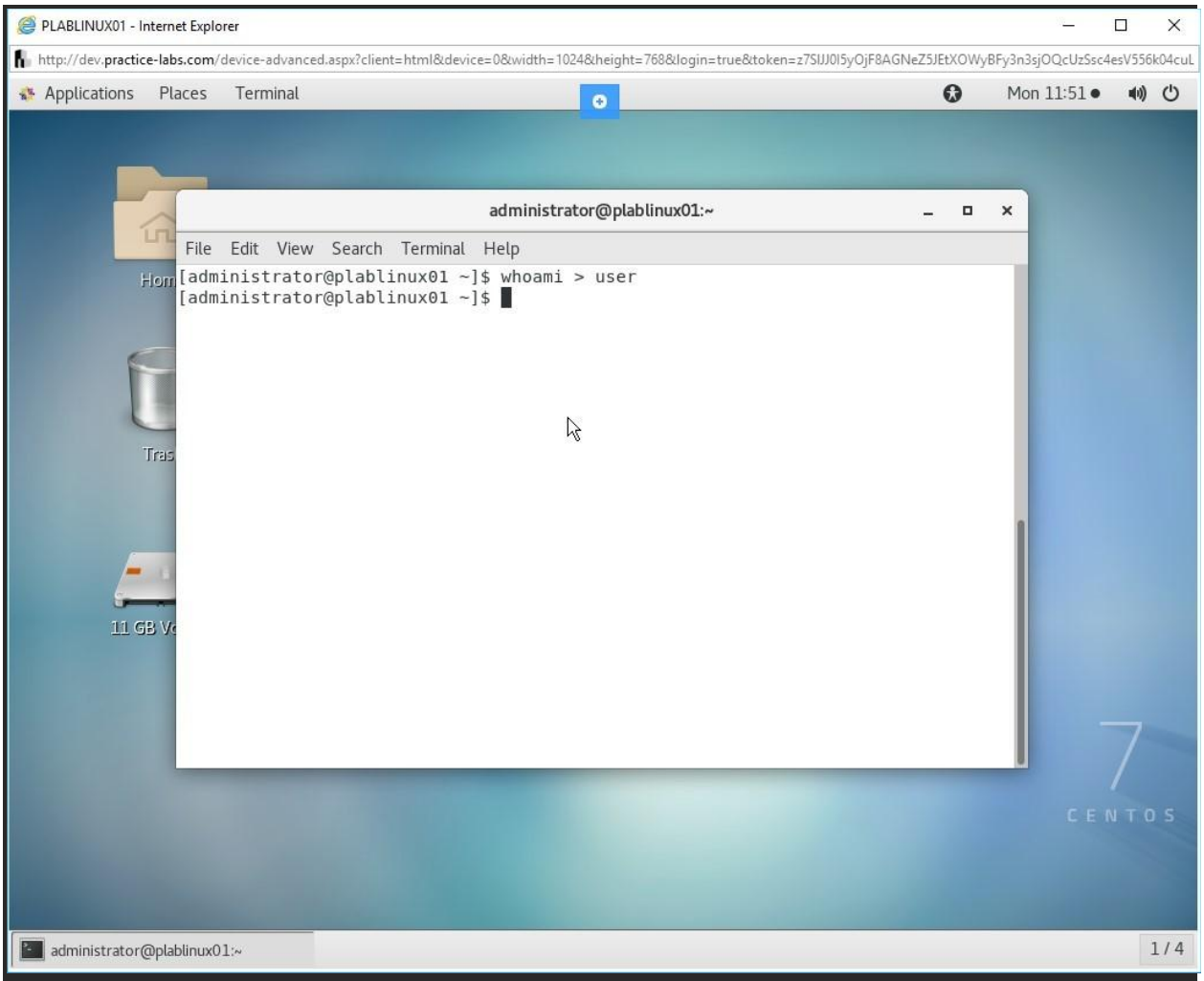

Figure 1.4 Screenshot of PLABLINUX01: Redirecting the output of the whoami command to a file named user.

Step 5

You need to verify the contents of the file named “user”. Type the following command:

```
cat user
```

Press Enter.

Notice that the file named “user” contains the output of the whoami command.

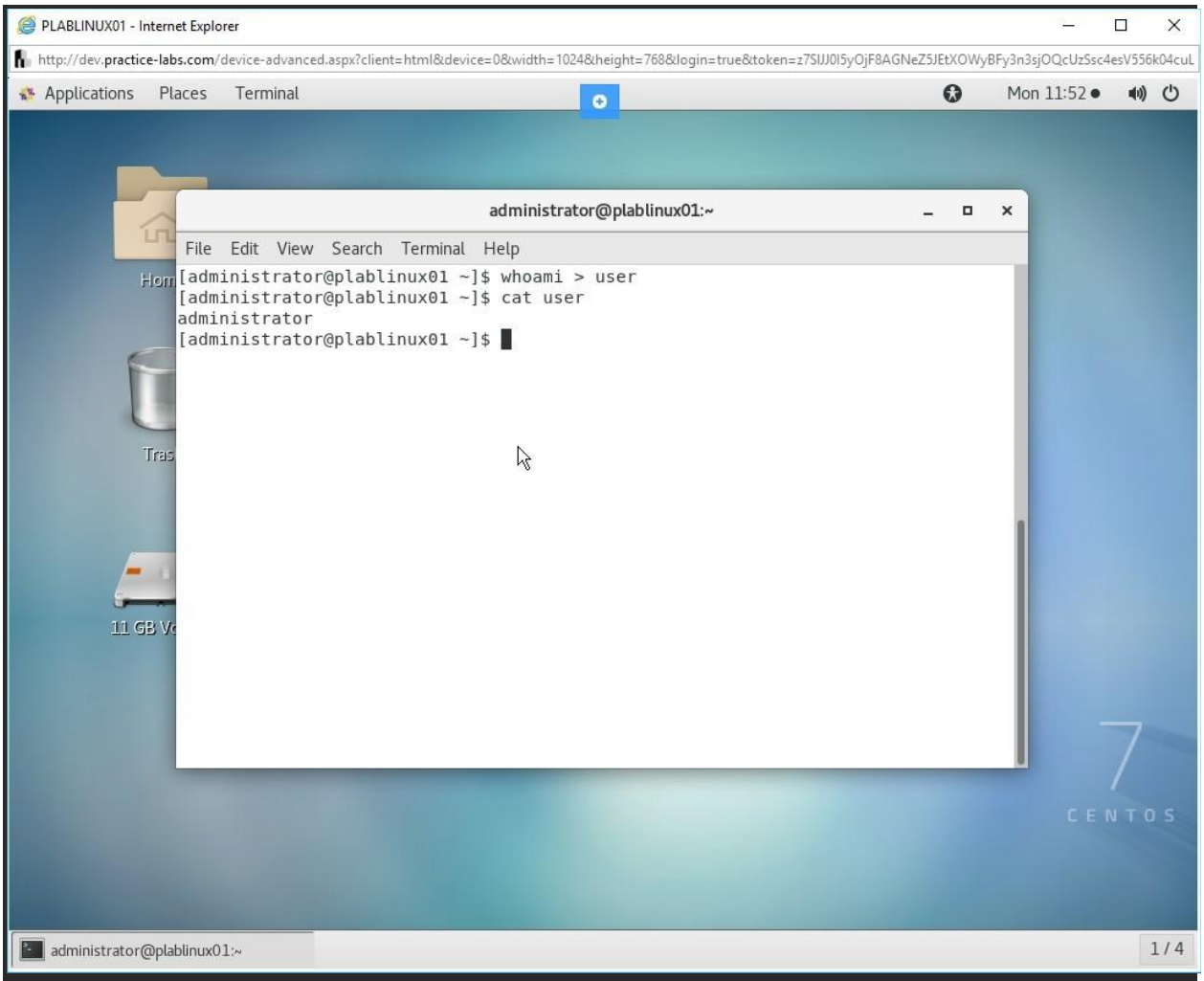

Figure 1.5 Screenshot of PLABLINUX01: Verifying the output of the user file.

Step 6

You can also append the output to the same file without overwriting it. Type the following command:

```
who >> user
```

Press Enter.

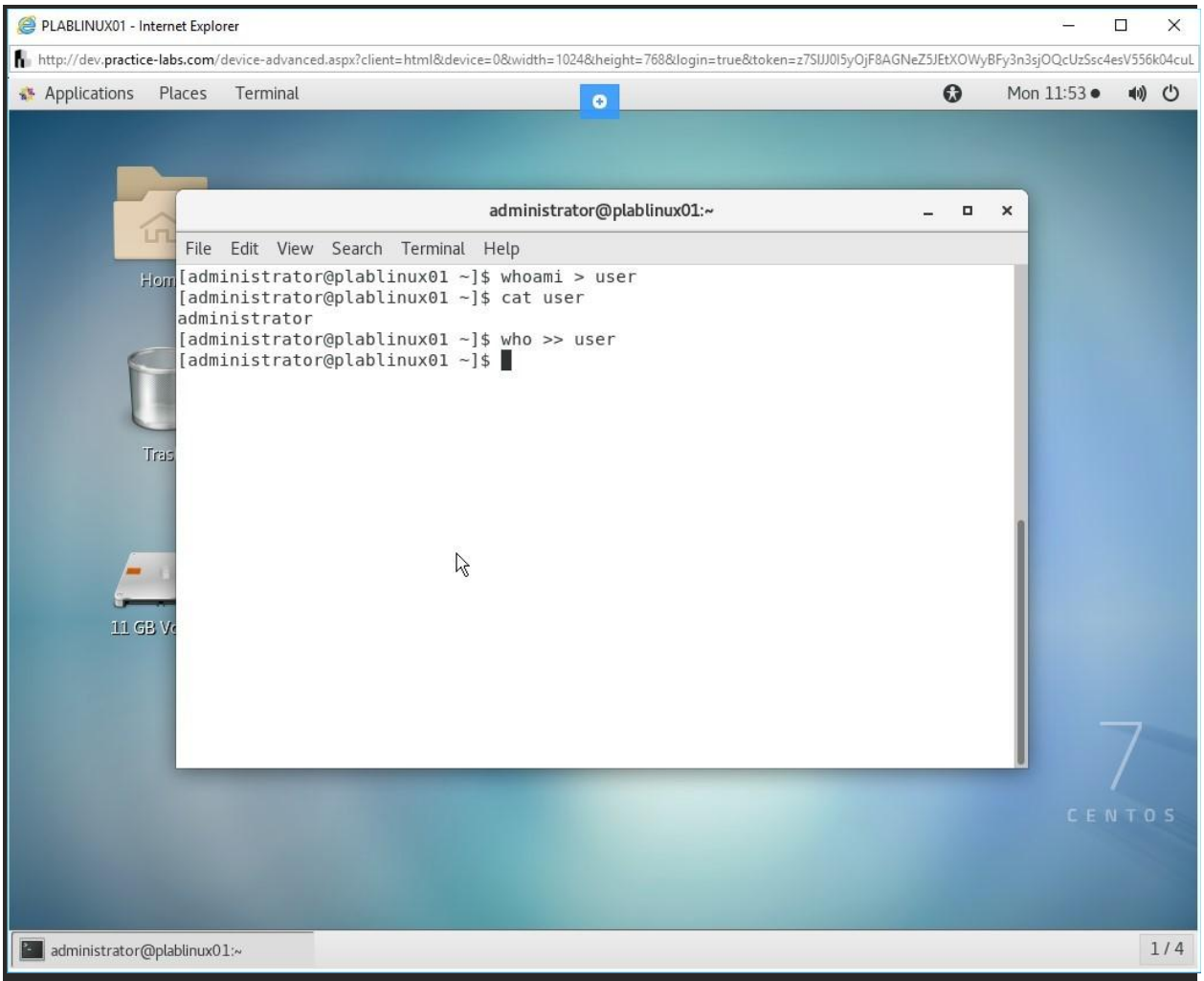

Figure 1.6 Screenshot of PLABLINUX01: Appending the output of the who command to the user file.

Step 7

You need to verify the contents of the file, user. Type the following command:

```
cat user
```

Press Enter. Notice that the file, user, contains the output of both the commands.

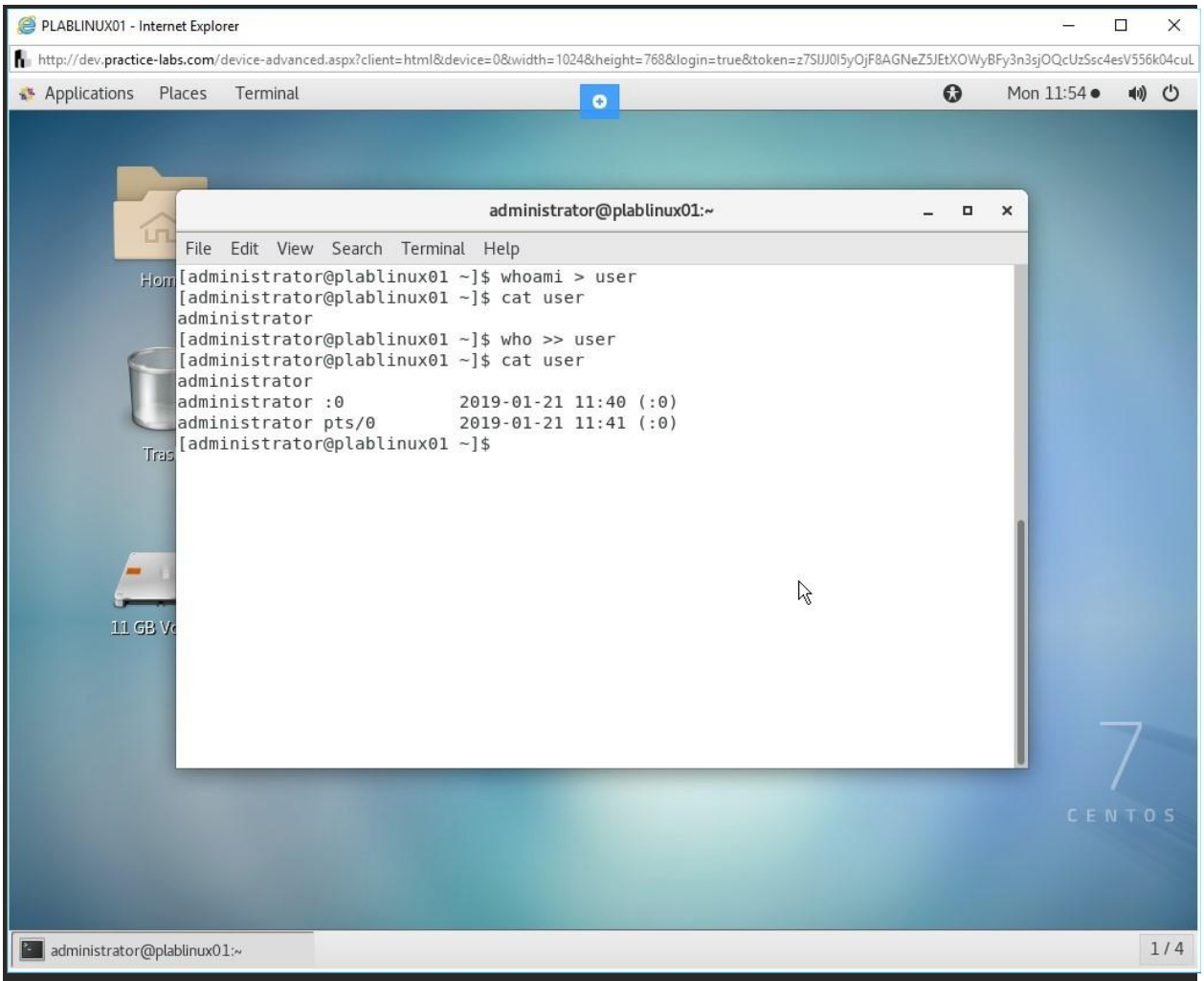

Figure 1.7 Screenshot of PLABLINUX01: Verifying the output of the user file.

Task 2 - Redirect Input

Similar to the output redirection, the user can also perform the input redirection. In the input redirection, the input from the file is used by the specified command. The input redirection uses the < operator.

To search for specific criteria through a file content or filesystem, perform the following steps:

Step 1

Clear the screen by entering the following command:

```
clear
```

With the file that you had created earlier with the name user, you can use the wc -l (word count) command which counts the lines (users) in the file and outputs the total number.

Type the following command:

```
wc -l user
```

Press Enter. Notice that the total number of users are displayed.

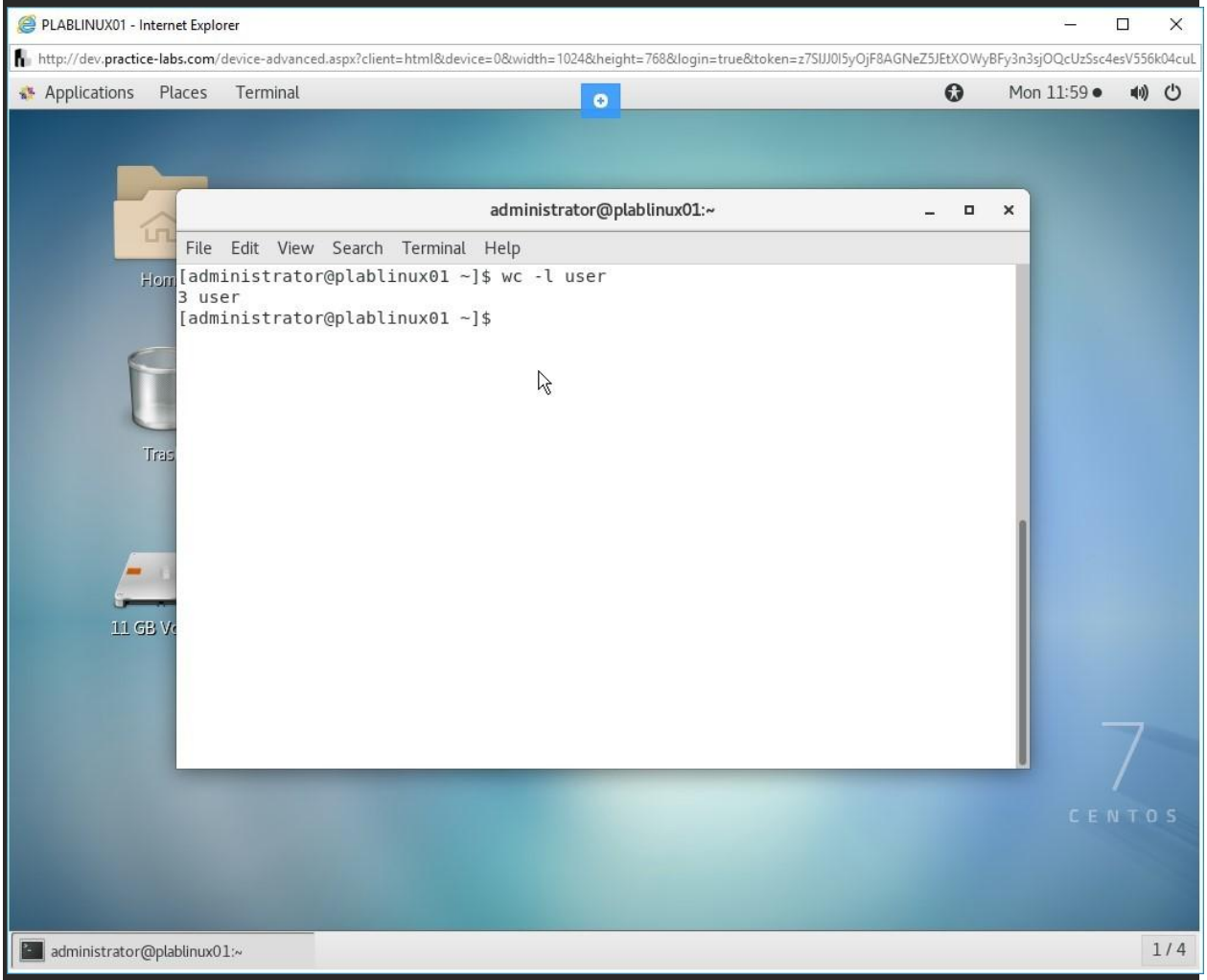

Figure 1.8 Screenshot of PLABLINUX01: Using the user file as input to the wc command.

Step 2

You serve the user file as the input to the wc -l command and get the total number of lines. Type the following command:

```
wc -l < user
```

Press Enter. Notice that the output is slightly different. The output displays the number of lines in the file, user.

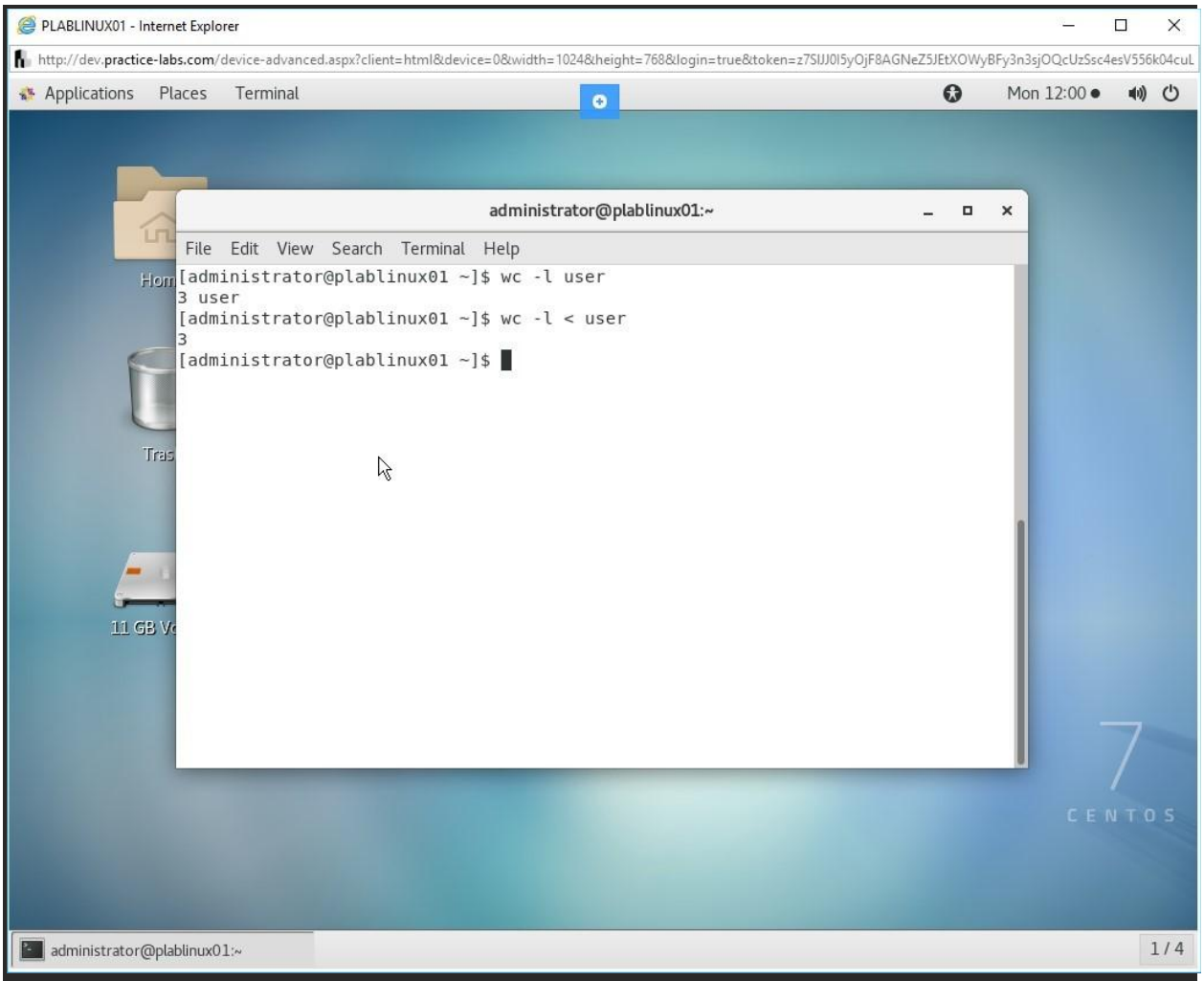

Figure 1.9 Screenshot of PLABLINUX01: Using the user file as input to the wc command.

Step 3

You can also use the here document for redirecting input, which can be directed to either an interactive shell script or program. Type the following command:

```
wc -l << EOF
This is just a test.
EOF
```

Press Enter after each statement. Notice that after the wc -l << EOF command, the command prompt changes to a > sign. You can terminate the command with another EOF statement. Then, the output is displayed.

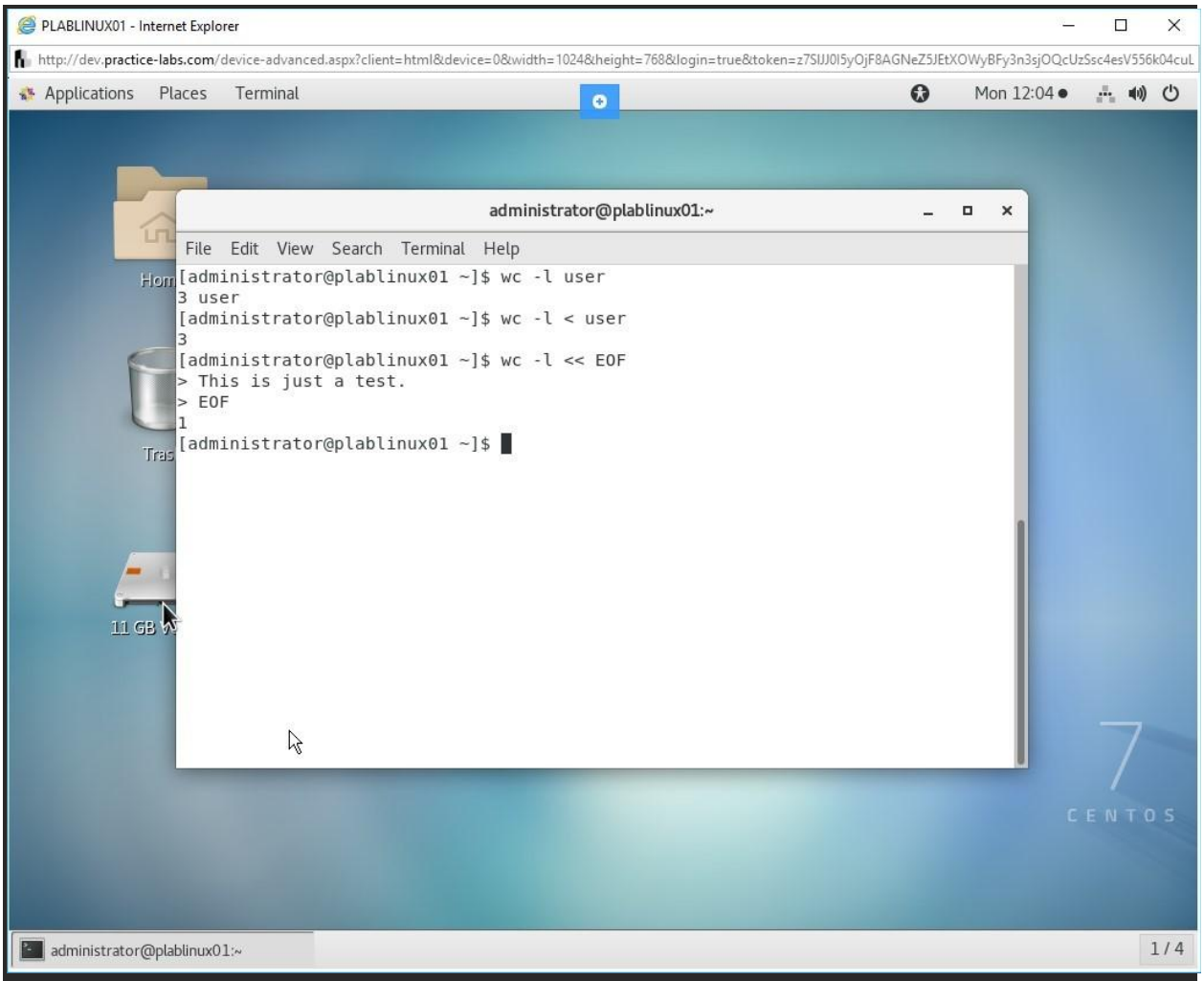

Figure 1.10 Screenshot of PLABLINUX01: Creating the here document.

Step 4

Clear the screen by entering the following command:

```
clear
```

You can also take the input from one file and then redirect the output to another file. Type the following command:

```
wc -l < user > count
```

Press Enter.

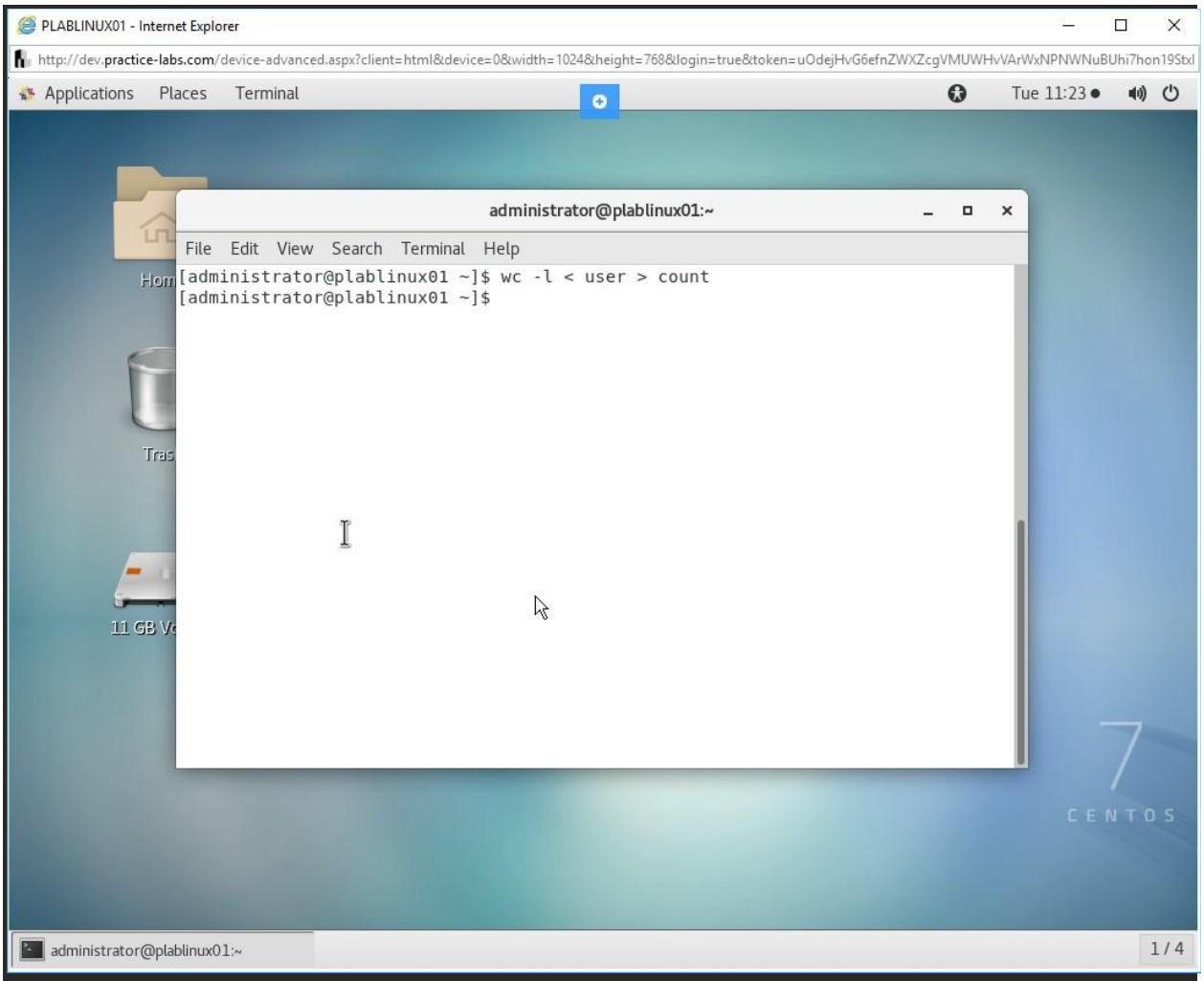

Figure 1.11 Screenshot of PLABLINUX01: Taking the input from one file and then redirecting the output to another file.

Step 5

Let's view the count file. Type the following command:

```
cat count
```

Press Enter.

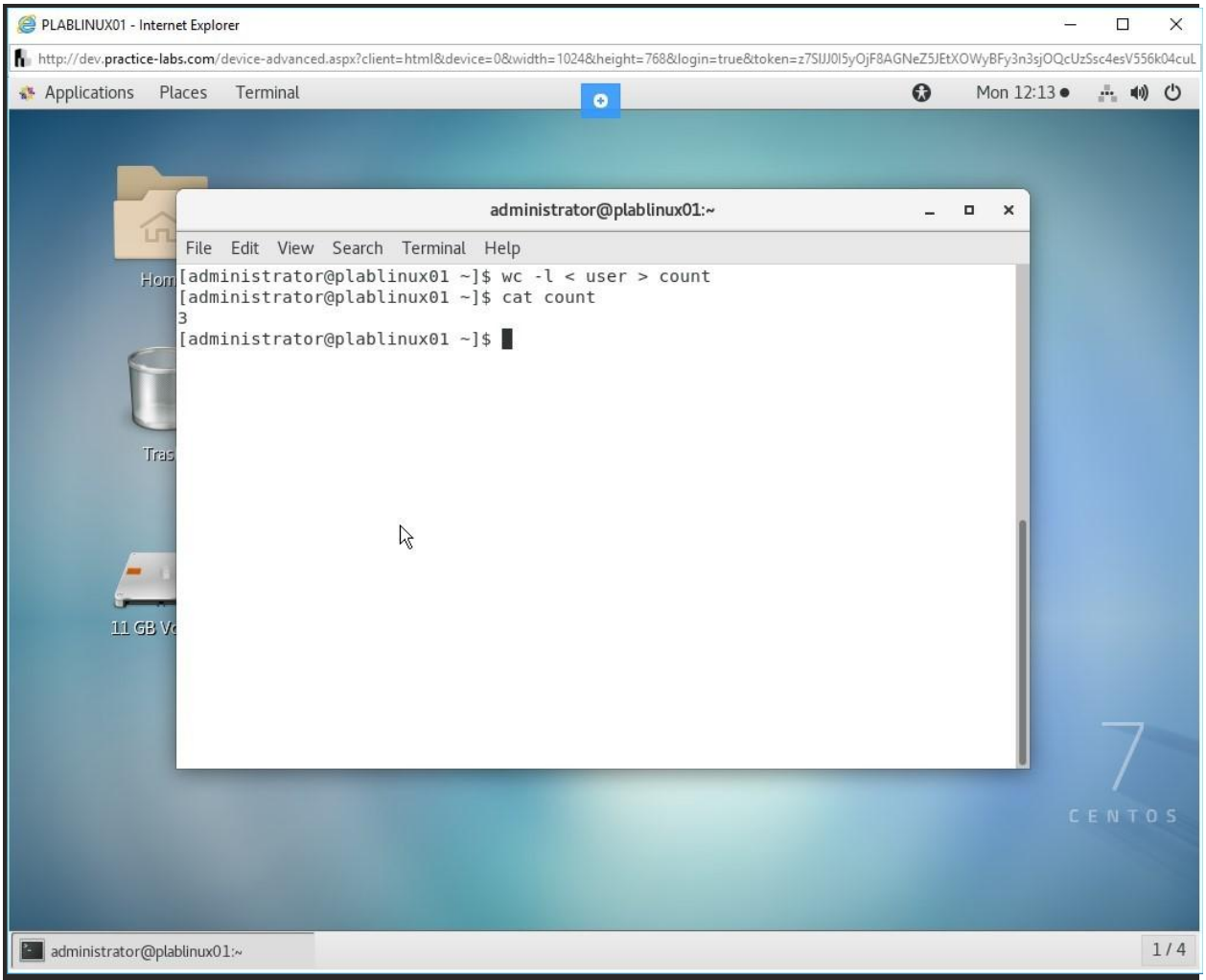

Figure 1.12 Screenshot of PLABLINUX01: Viewing the count file.

Task 3 - Discard the Output

There are many programs and applications that return the output as 0 and 1. You do not want to display the output on the screen, and therefore, you can choose to discard it. The redirection of an error can be done with the help of a file descriptor, which is a unique and pre-process value that refers to an open file. Each file is then assigned a unique and pre-process value or the file descriptor.

Assume the scenario in which the user wants to view a file on the monitor. In this case, when the user executes the command, the output of the command is sent to the file descriptor of the monitor screen, and the output is displayed. In a similar scenario, if the user executes a command to print a file, the output is sent to file descriptor of the associated printer.

By default, there are three file descriptors:

- Stdin: 0
- Stdout: 1
- Stderr: 2

To discard the output, perform the following steps:

Step 1

Clear the screen by entering the following command:

```
clear
```

Let's sent the output of the ls command to /dev/null. In a normal scenario, the ls command will display the output on the screen. Type the following command:

```
ls > /dev/null
```

Press Enter. Notice that the output is not displayed on the screen.

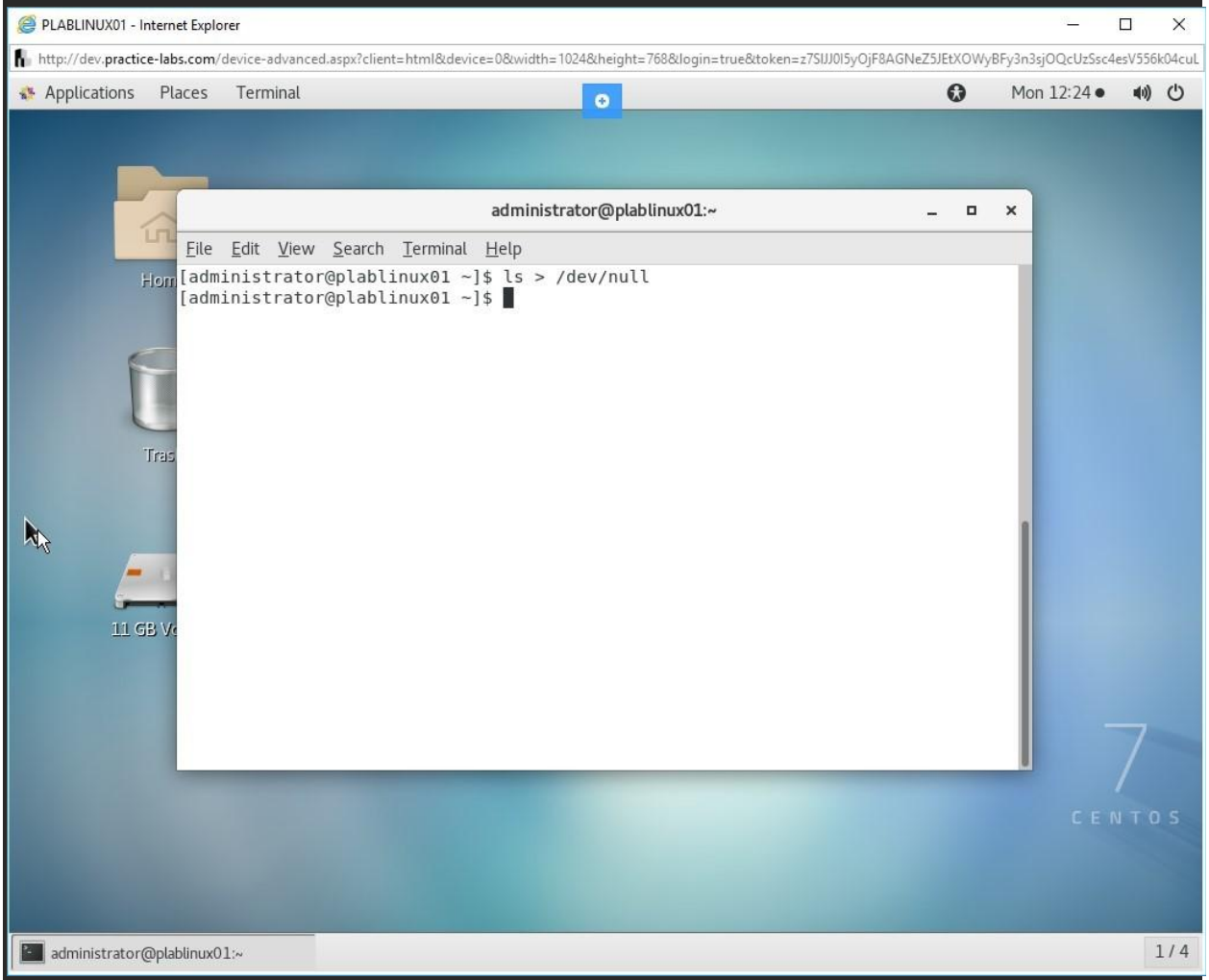

Figure 1.13 Screenshot of PLABLINUX01: Using /dev/null to discard output of the ls command.

Step 2

Clear the screen by entering the following command:

```
clear
```

You can also discard output of a command and the error output if required. You can use the standard redirection to redirect STDERR to STDOUT. Let's sent the output of the ls command to /dev/null. In a normal scenario, the ls command will display the output on the screen but in this scenario, it will not. Type the following command:

```
ls > /dev/null 2>&1
```

Press Enter. Notice that the output is not displayed on the screen.

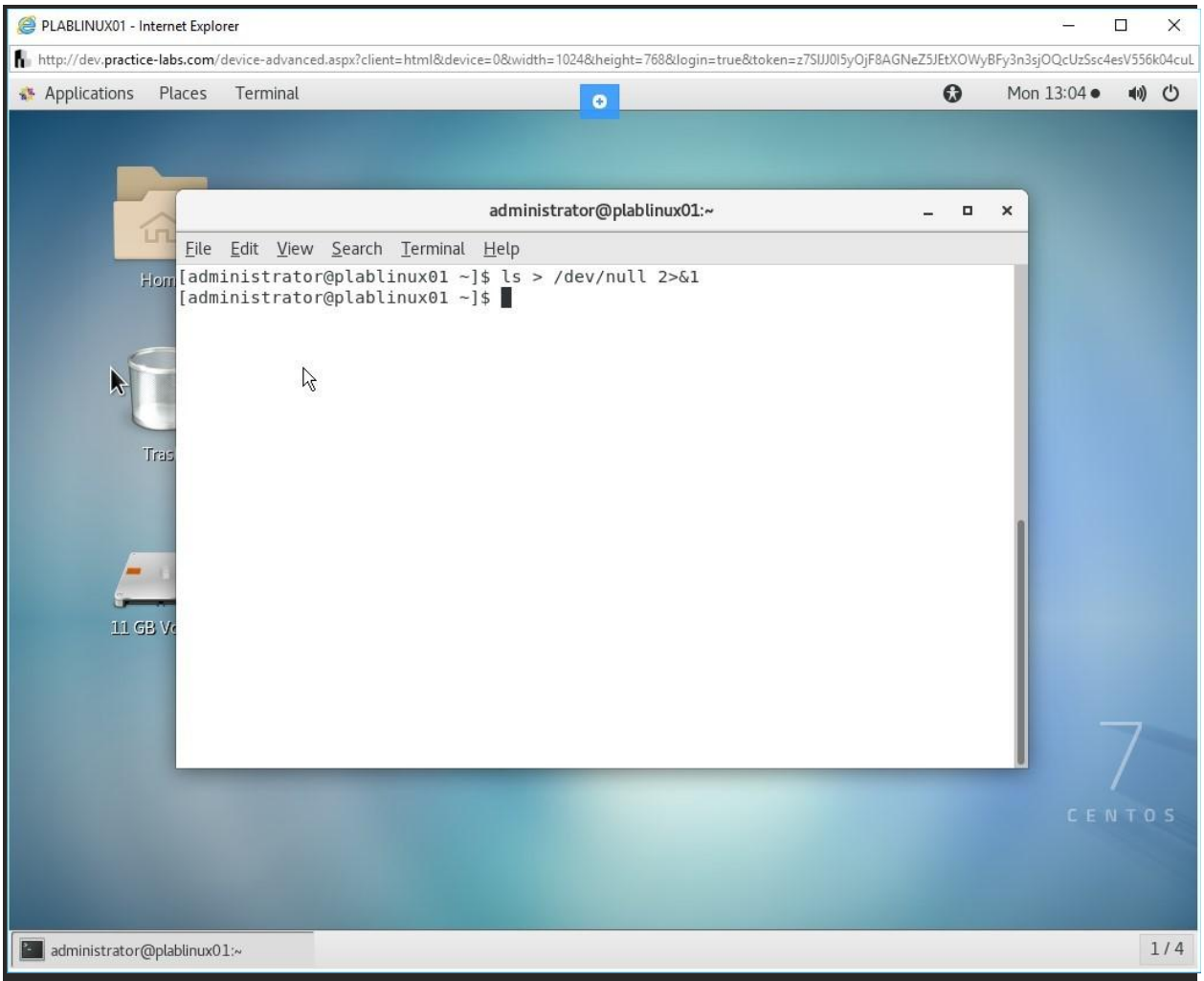

Figure 1.14 Screenshot of PLABLINUX01: Discarding output of a command and the error output if required.

Task 4 - Use the tee Command

There will be scenarios in which you need to display the output and also save it in a file. The tee command is used for this purpose. It will display the output and save it

To use the tee command, perform the following steps:

Step 1

Clear the screen by entering the following command:

```
clear
```

The wc command is taking the /etc/hosts file and counting its characters, lines, and words. The output is displayed on the screen as well as saved in the file named plab.txt.

It is interesting to note that if the file does not exist, it will be created. Type the following command:

```
wc /etc/hosts | tee plab.txt
```

Press Enter. Notice that the number of characters, lines, and words are displayed.

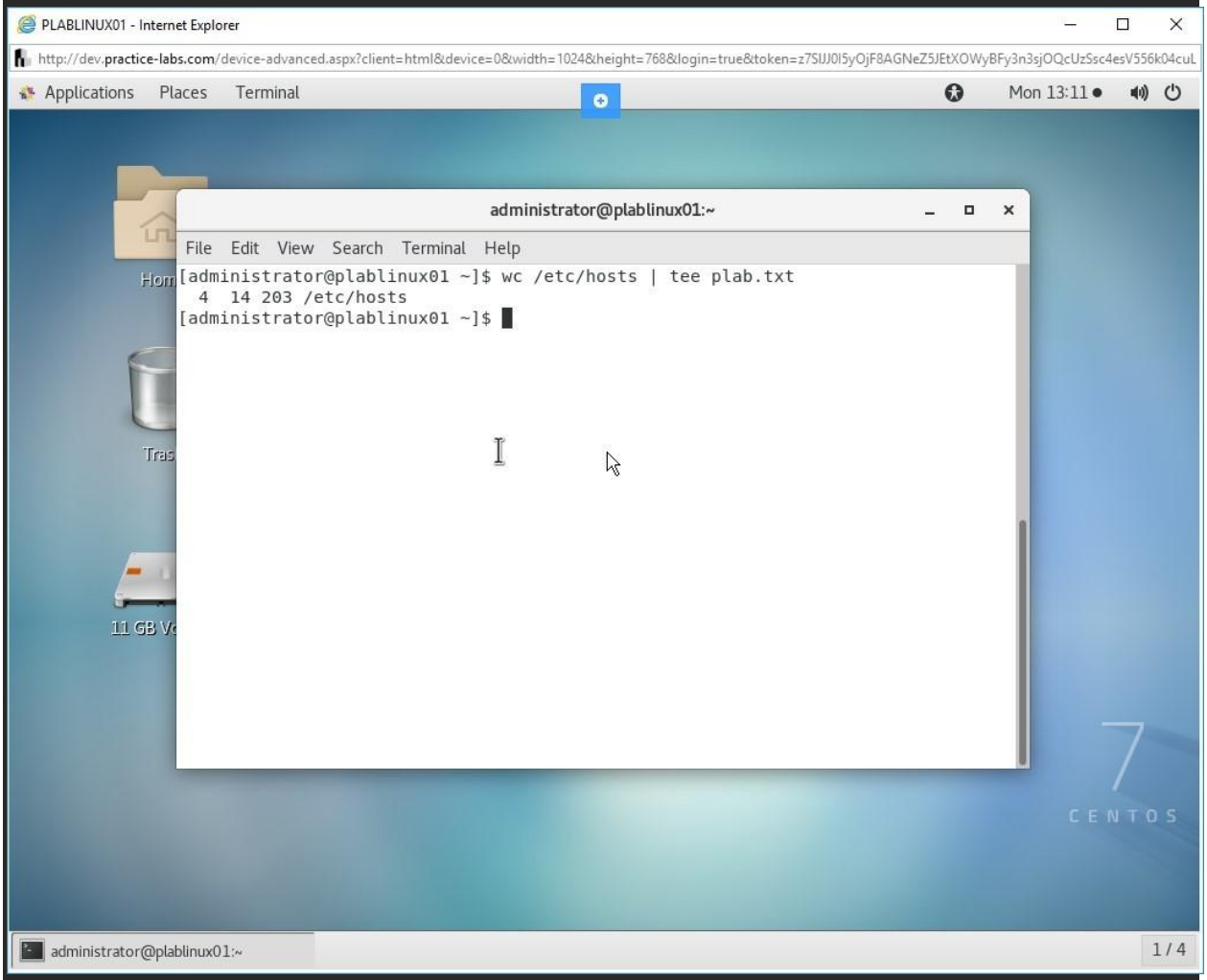

Figure 1.15 Screenshot of PLABLINUX01: Displaying the output on the screen and creating the plab.txt file.

Step 2

Let's verify the plab.txt file. Type the following command:

```
cat plab.txt
```

Press Enter. Notice that the number of characters, lines, and words are displayed. The output is what you saw in the previous command.

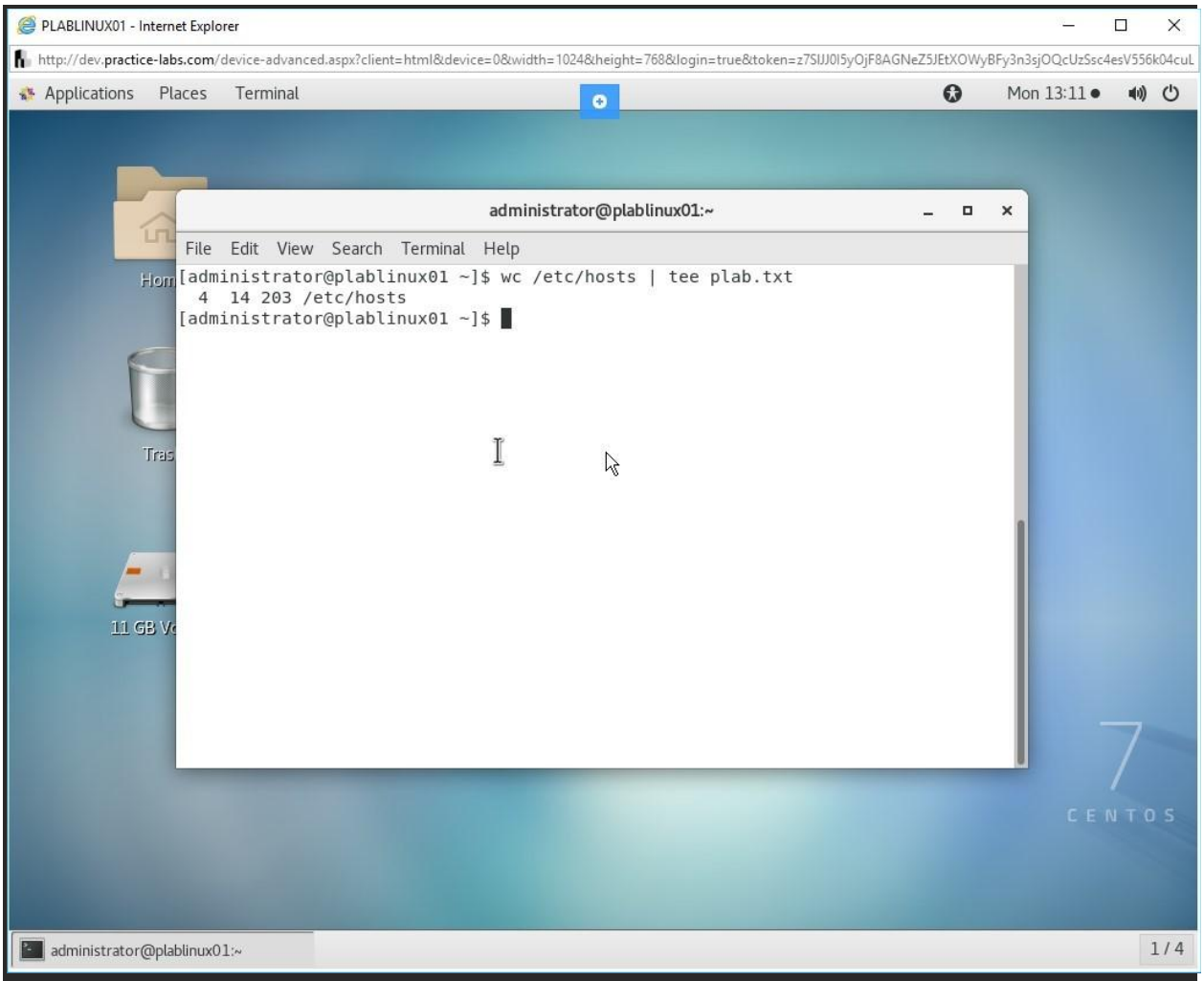

Figure 1.16 Screenshot of PLABLINUX01: Verifying the plab.txt file.

Step 3

Clear the screen by entering the following command:

```
clear
```

You can also redirect the output to multiple files at once. Type the following command:

```
ls -l | tee plab1 plab2 plab3
```

Press Enter.

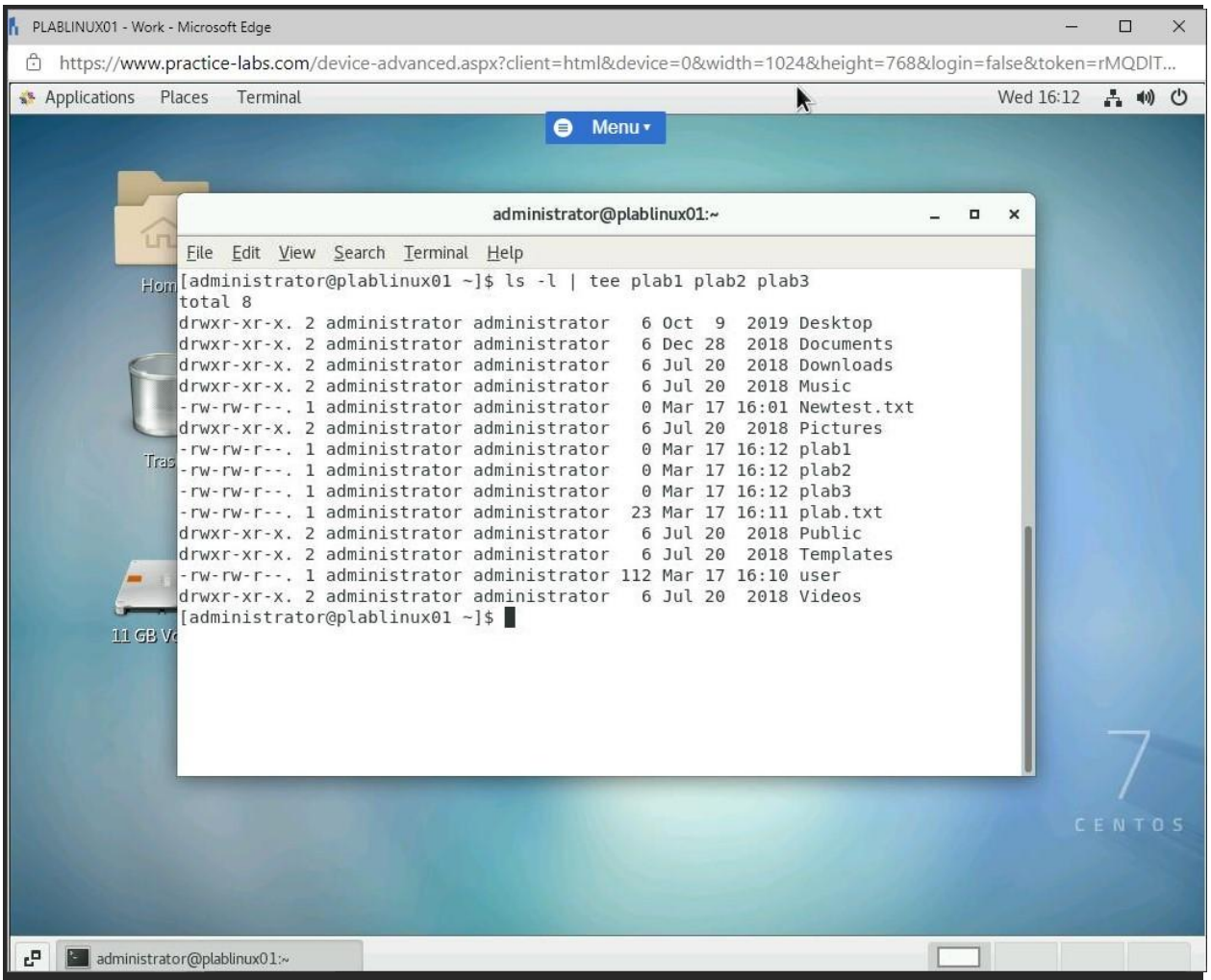

Figure 1.17 Screenshot of PLABLINUX01: Redirecting the output to multiple files at once.

Step 4

Clear the screen by entering the following command:

```
clear
```

Let's verify if plab1, plab2, and plab3 files are created. Type the following command:

```
ls -l
```

Press Enter.

Notice that there are four files with the names containing plab. The first file is plab.txt, which you had created in previous commands.

The three new files, plab1, plab2, and plab3, are now created.

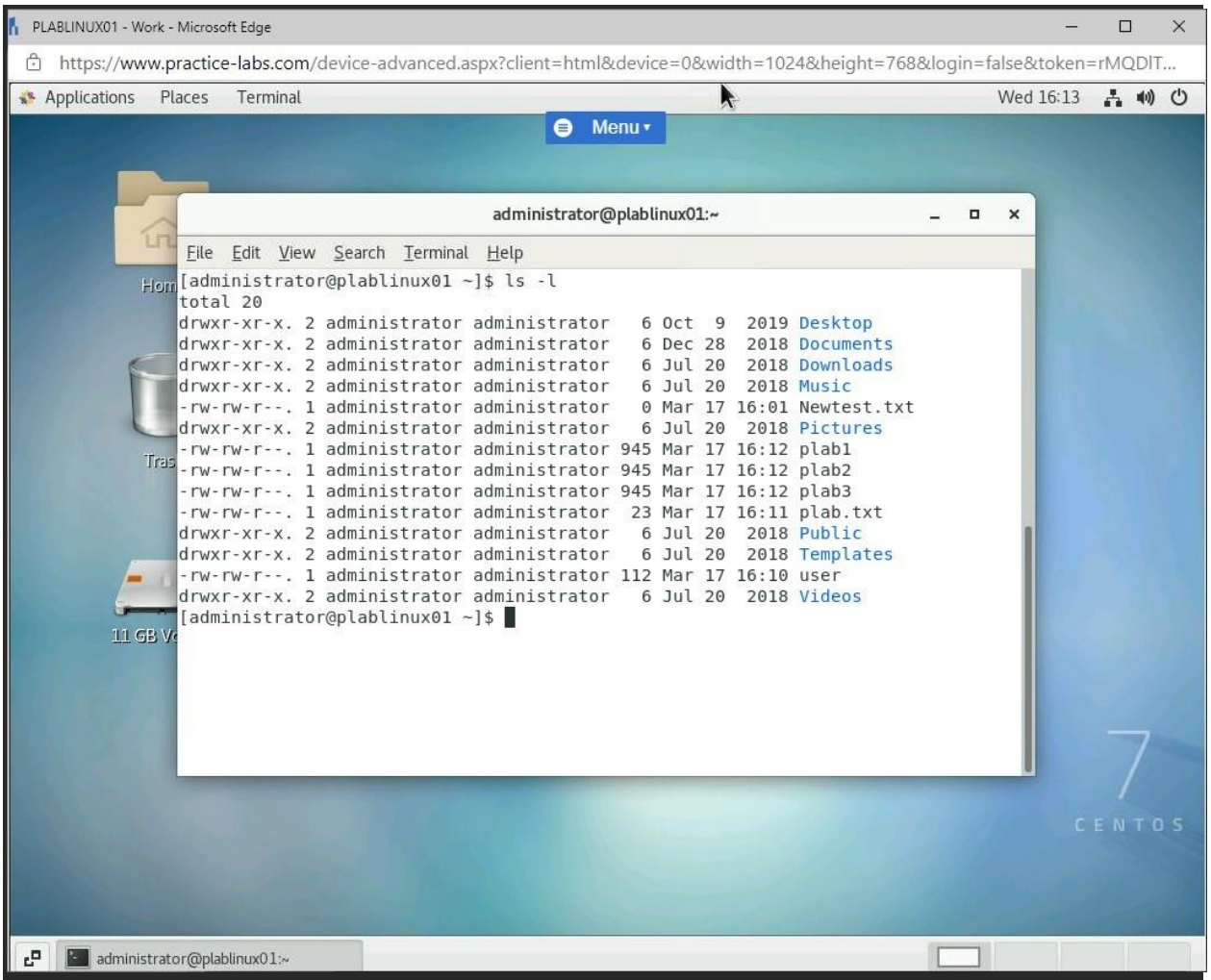

Figure 1.18 Screenshot of PLABLINUX01: Verifying if plab1, plab2, and plab3 files are created.

Step 5

Clear the screen by entering the following command:

```
clear
```

Let's verify the contents of one of the three files. Type the following command:

```
cat plab3
```

Press Enter.

Notice that it contains the same output as the output shown in Step 3.

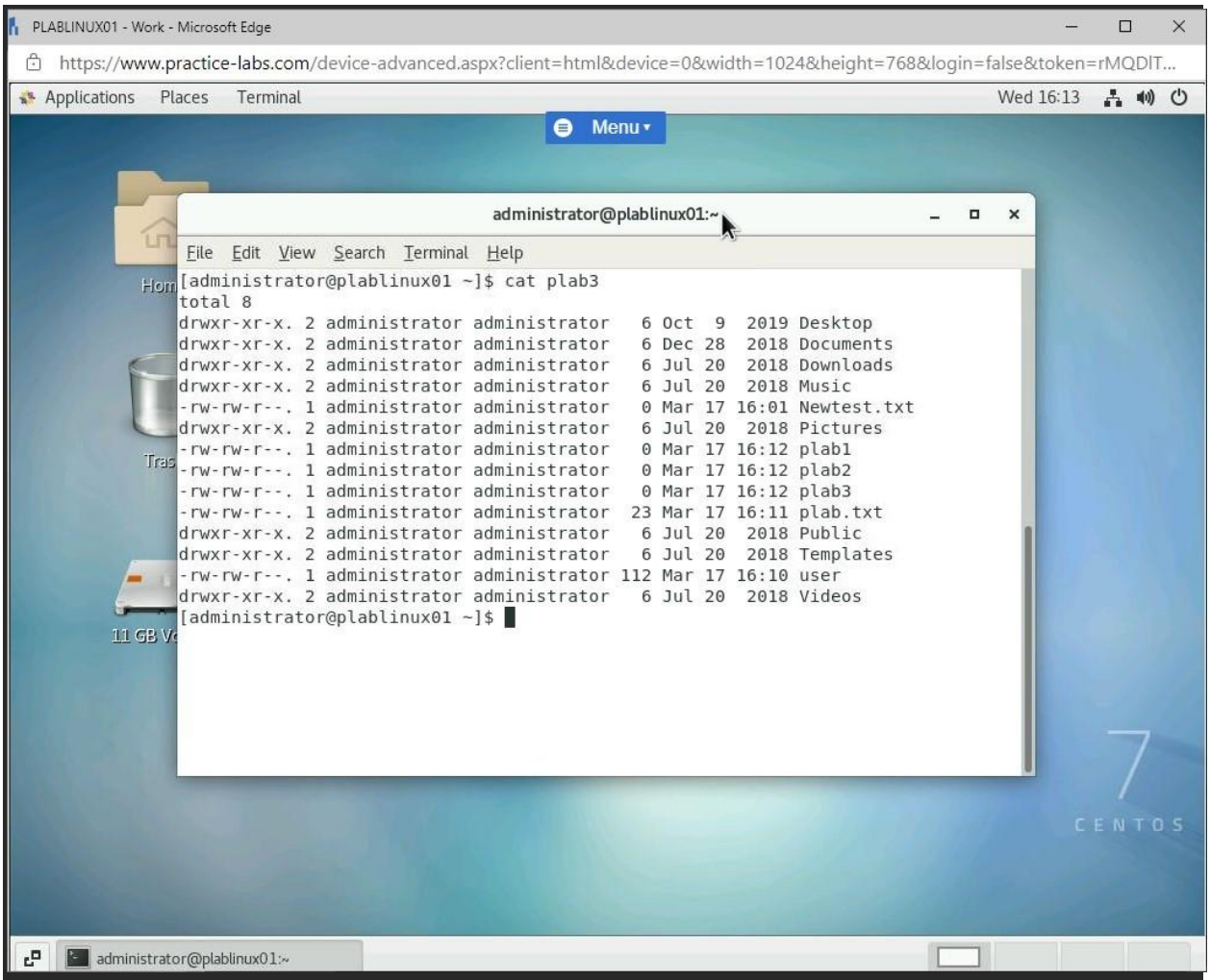

Figure 1.19 Screenshot of PLABLINUX01: Verifying the contents of the plab3 file.