CompTIA Linux+

Work with the Flow Control Constructs

Introduction

Welcome to the **Work with the Flow Control Constructs** Practice Lab. In this module you will be provided with the instructions and devices needed to develop your hands-on skills.

Flow Control Constructs
Loop
Linux System

Learning Outcomes

In this module, you will complete the following exercise:

- Exercise 1 - Work with the Flow Control Constructs

After completing this lab, you will be able to:

- Use the if…else statement
- Use the while loop
- Use the until loop
- Use the for loop

Exam Objectives

The following exam objectives are covered in this lab:

- **LPI:** 105.2 Customize or write simple scripts
- **CompTIA:** 5.1 Given a scenario, deploy and execute basic BASH scripts.

Note: *Our main focus is to cover the practical, hands-on aspects of the exam objectives. We recommend referring to course material or a search engine to research theoretical topics in more detail.*

Lab Duration

It will take approximately **1 hour** to complete this lab.

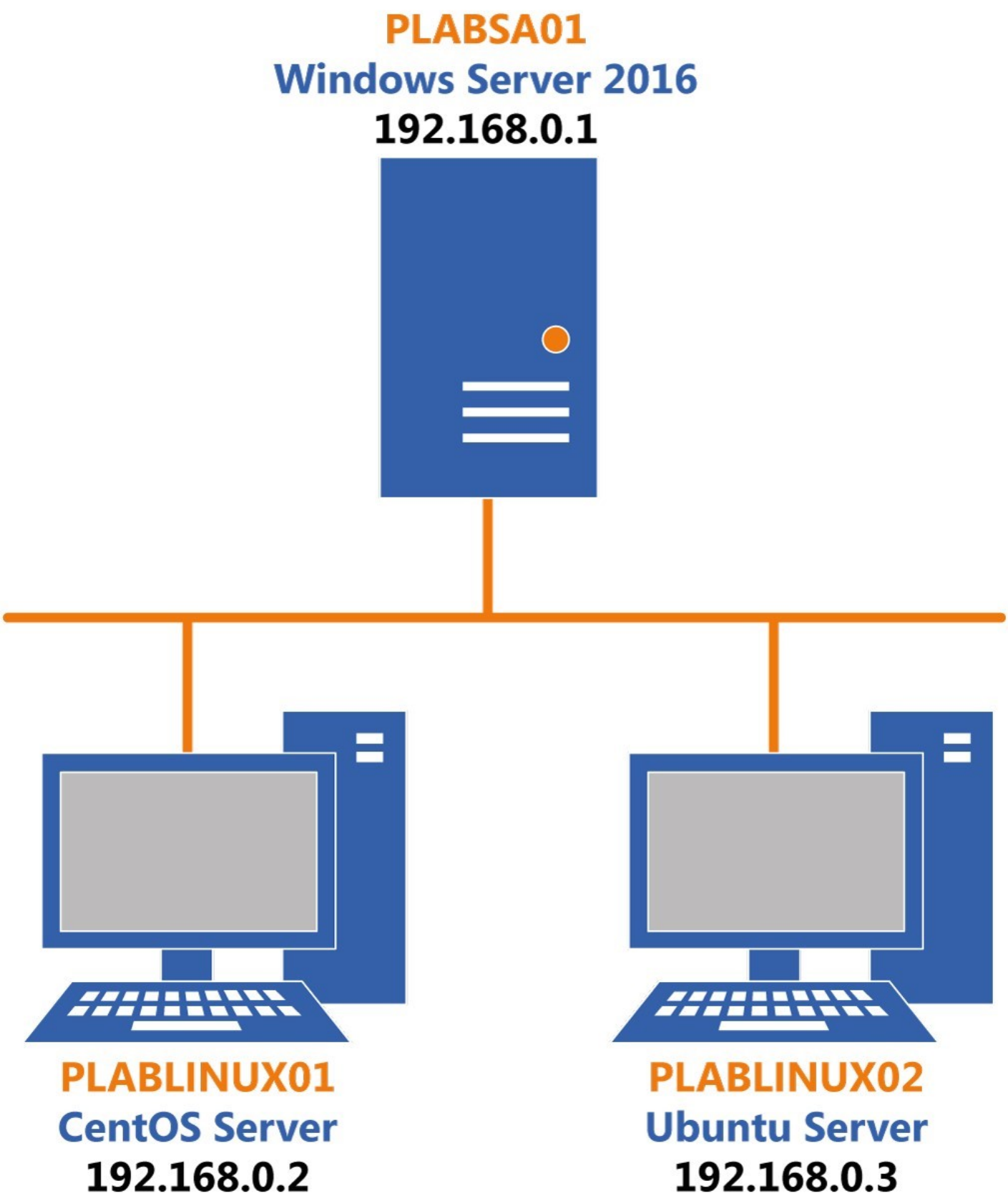
Help and Support

For more information on using Practice Labs, please see our **Help and Support** page. You can also raise a technical support ticket from this page.

Click Next to view the Lab topology used in this module.

Lab Topology

During your session, you will have access to the following lab configuration.



Depending on the exercises you may or may not use all of the devices, but they are shown here in the layout to get an overall understanding of the topology of the lab.

- **PLABSA01** (Windows Server 2016)
- **PLABLINUX01** (CentOS Server)
- **PLABLINUX02** (Ubuntu Server)

Click Next to proceed to the first exercise.

Exercise 1 - Work with the Flow Control Constructs

A server can hold more than one hard drive. Due to an unforeseen error, there are changes that the server may crash, which can also impact the hard drive by making it unusable. The server administrator must regularly check the filesystem health, which is installed on the hard drive. CentOS provides several commands for this purpose.

In this exercise, you will learn to view the hard drive details.

Learning Outcomes

After completing this exercise, you will be able to:

- Log into a Linux System
- Use if...else statements

Your Devices

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

- **PLABLINUX01** (CentOS Server)

Task 1 - Use if...else statements

The if...else statements are mainly used for making a decision and therefore, are known as decision-making statements. When you have to make a choice from two given choices, the if...else statements work well. However, there may be a scenario in

which you will have to work with multiple conditions to choose from, then how to use the case...esac statements.

In this task, you will learn to use if...else statements. To use the if...else statements, perform the following steps:

Step 1

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

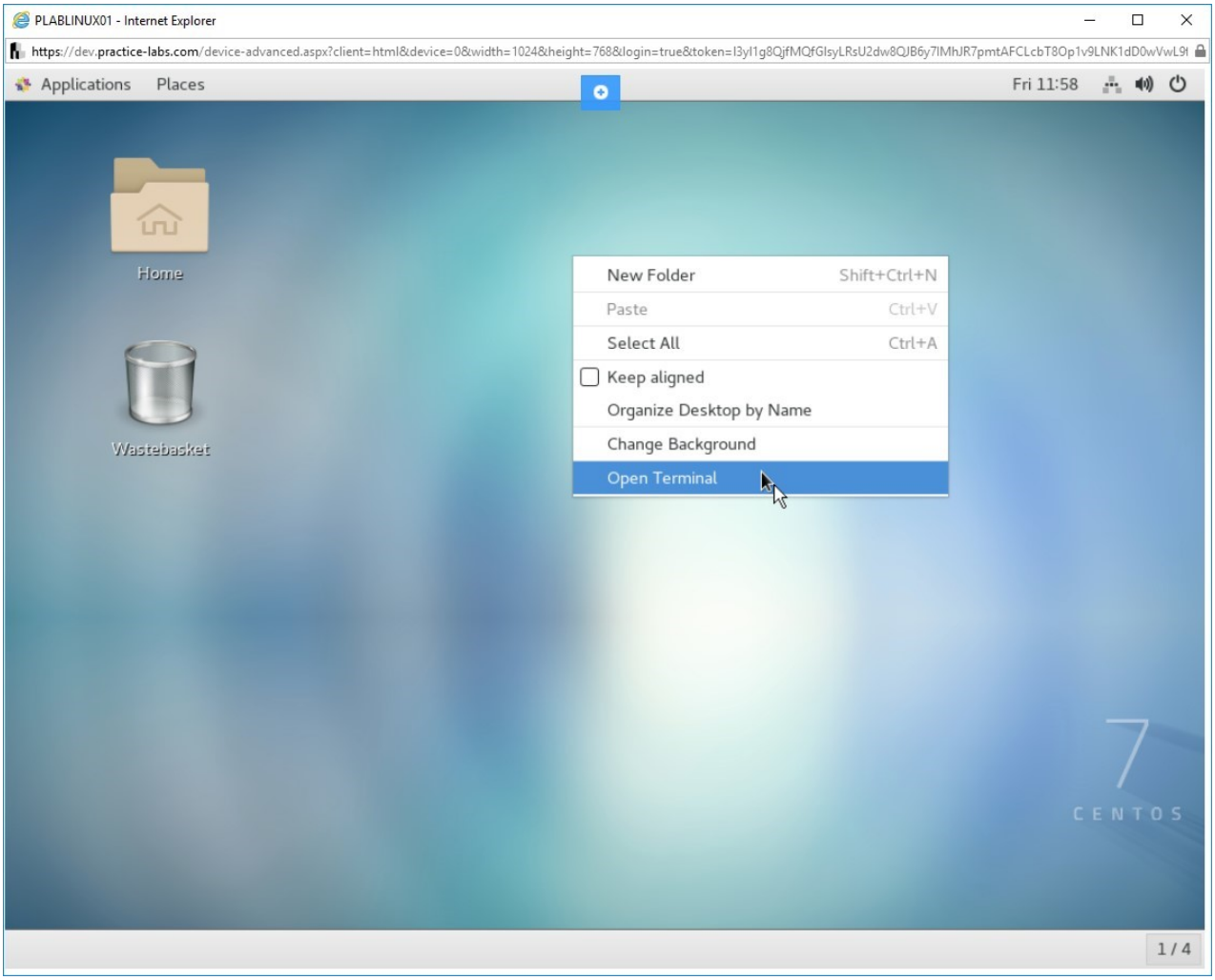


Figure 1.1 Screenshot of PLABLINUX01: Selecting the Open Terminal option from the context menu.

Step 2

In shell scripting, you can use three different types of if...else statements. They are the following:

- if ... fi statement
- if ... else ... fi statement

- if … elif … else … fi statement

Let's first create a new shell script with the following command:

```
gedit if_fi.sh
```

Press **Enter**.

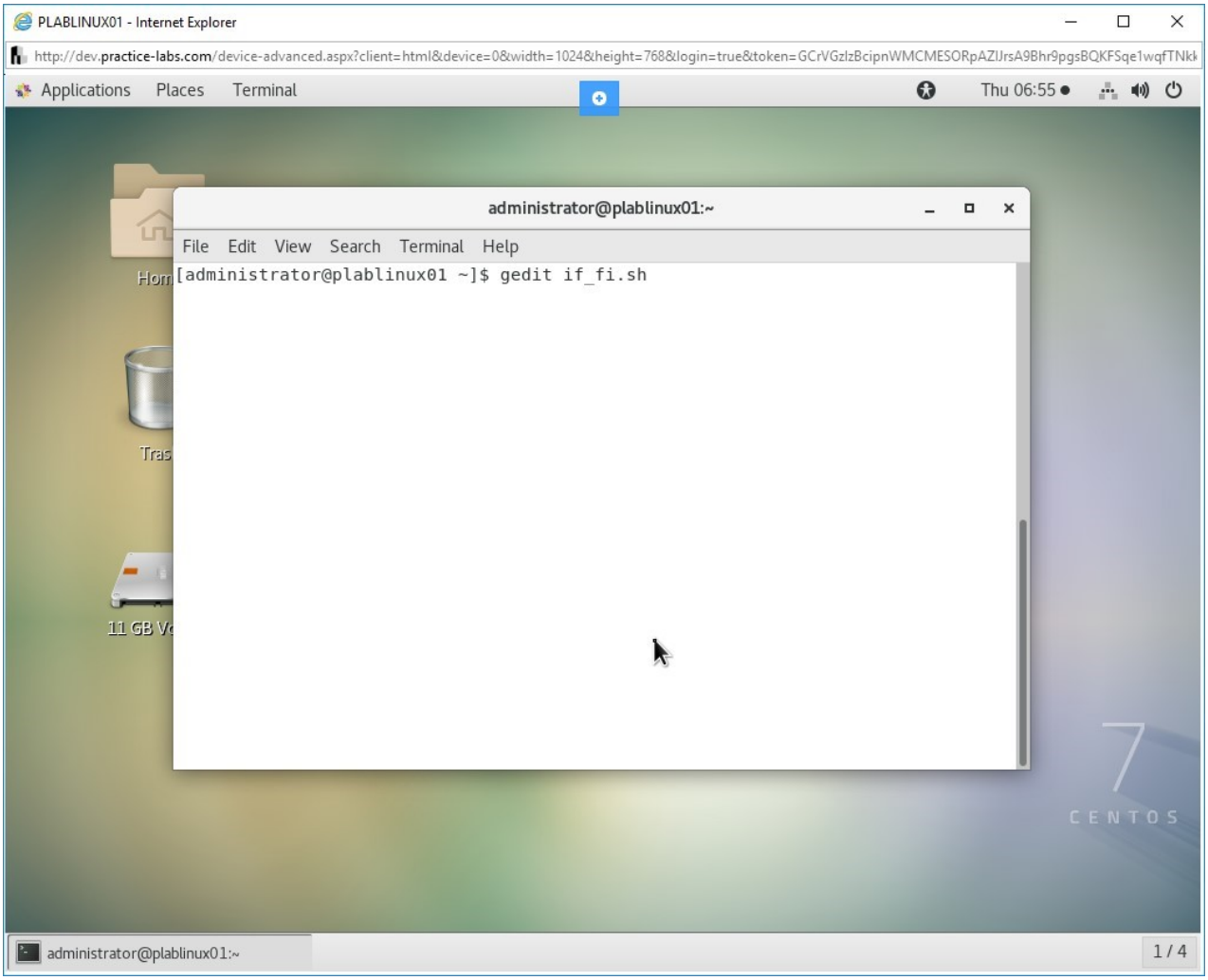


Figure 1.2 Screenshot of PLABLINUX01: Opening a new file in gedit.

Step 3

Notice a new file is created with the name **if_fi.sh**. Type the following:

```
#!/bin/sh
x=10
z=20
```

```
if [ $x == $z ]
then
   echo "x is same as z"
elif [ $x -gt $z ]
then
   echo "x is more than z"
elif [ $x -lt $z ]
then
   echo "x is less than z"
else
   echo "None of them match"
fi
```

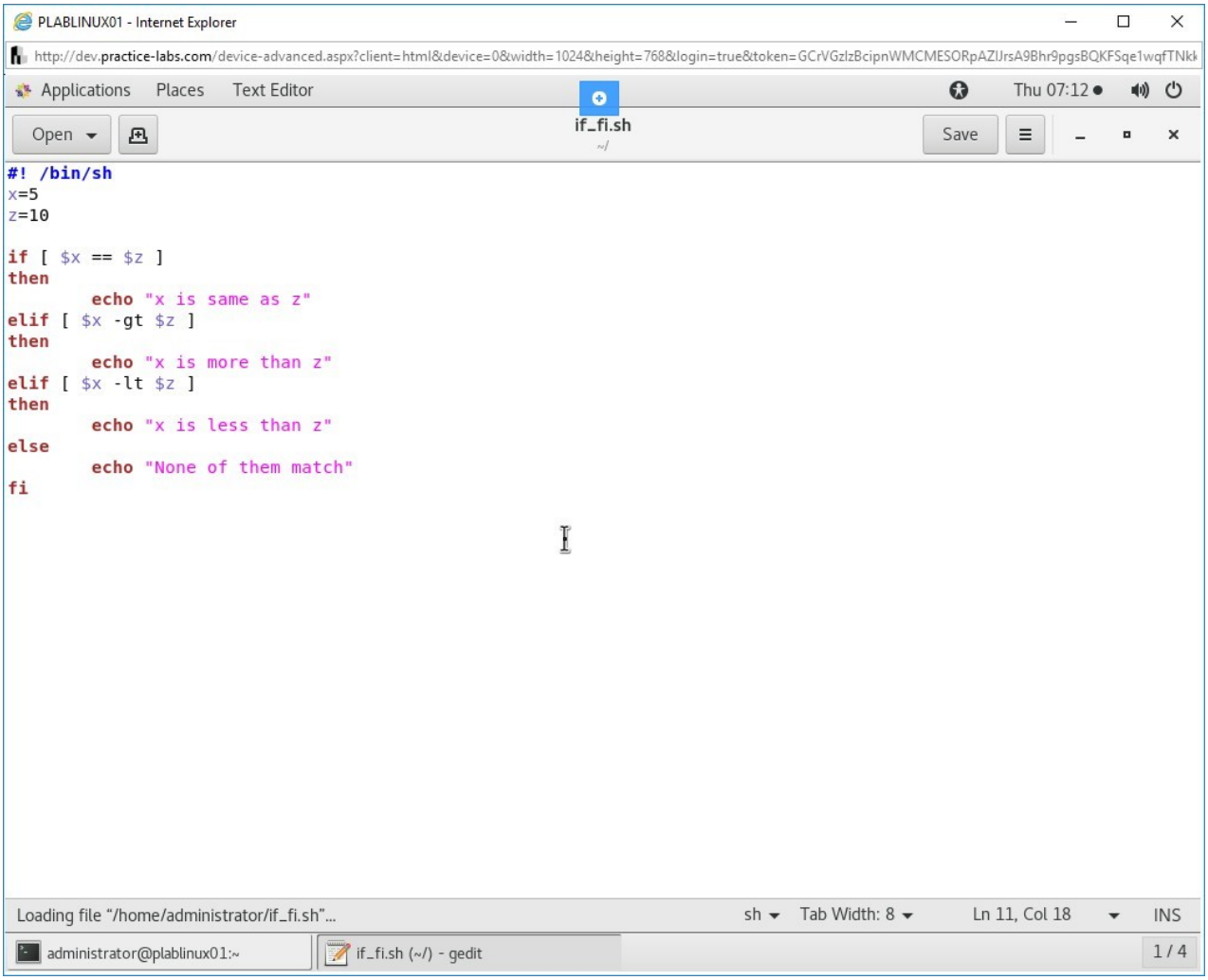


Figure 1.3 Screenshot of PLABLINUX01: Entering the content in the file.

Step 4

Click **Save** to save the file. Then, close the file.

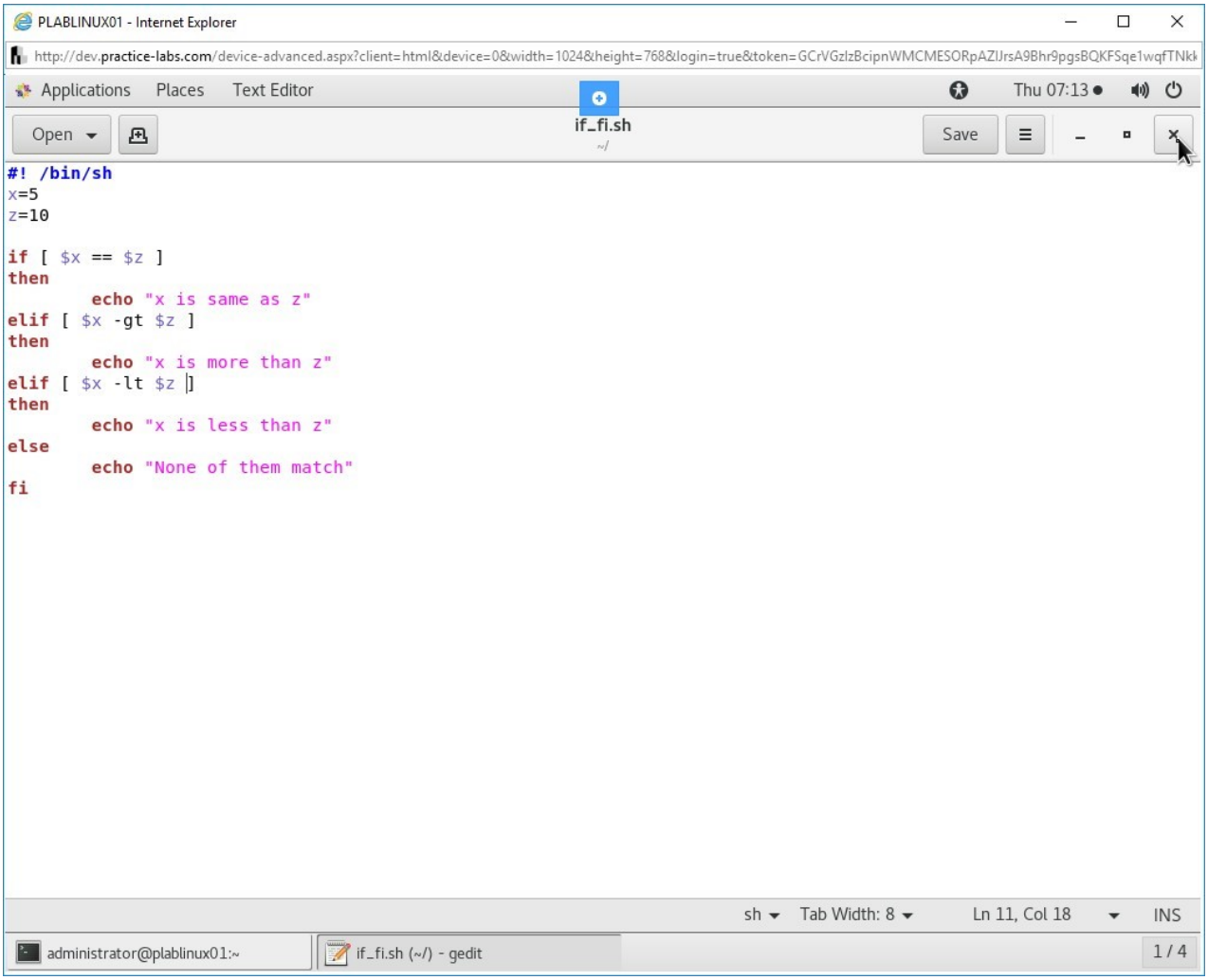


Figure 1.4 Screenshot of PLABLINUX01: Saving and closing the file.

Step 5

Clear the screen by entering the following command:

```
clear
```

You need to make the shell script executable. Type the following command:

```
sudo chmod +x if_fi.sh
```

Press **Enter**.

When prompted, type the following password:

`Passw0rd`

Press **Enter**.

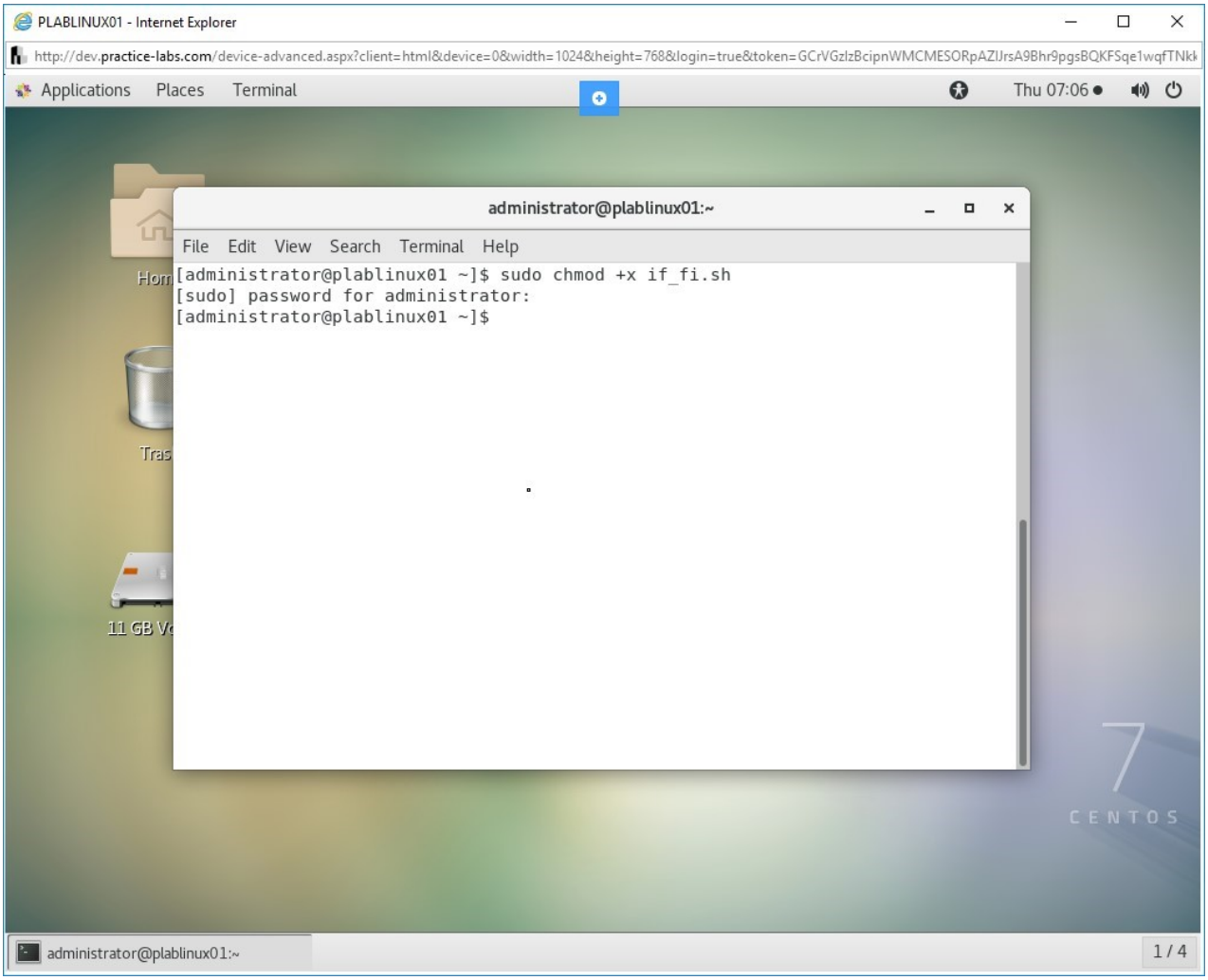


Figure 1.5 Screenshot of PLABLINUX01: Making the file executable.

Step 6

Clear the screen by entering the following command:

```
clear
```

You will now execute the shell script. Type the following command:

```
./if_fi.sh
```

Press **Enter**. Note that the last condition is met in which x is less than z.

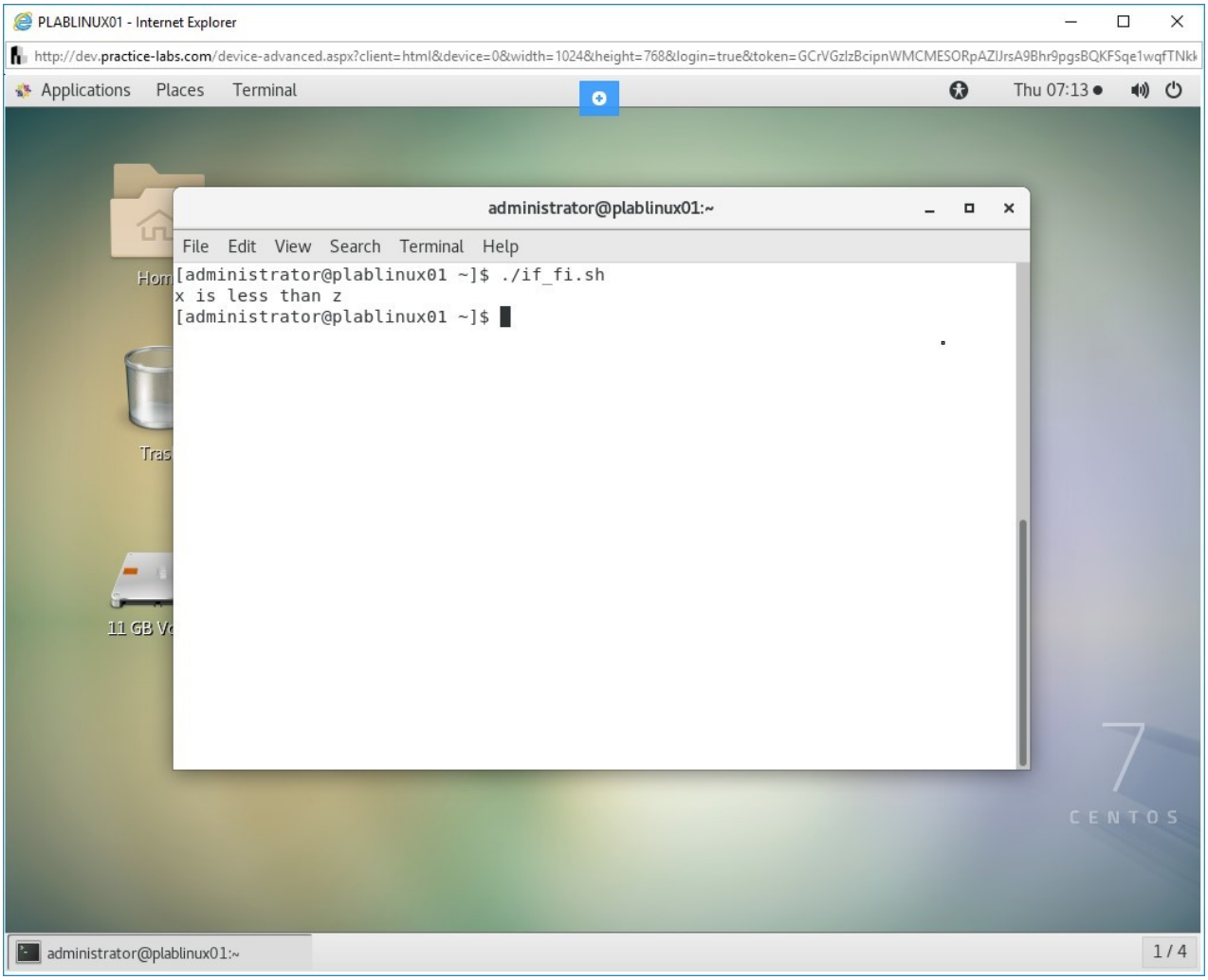


Figure 1.6 Screenshot of PLABLINUX01: Executing the shell script.

Step 7

In a scenario, where you have multiple conditions, it is best to use case...esac statements. Create a new file with gedit and name is case.sh

Type the following:

```
#!/bin/sh
CAR="Honda"
case "$CAR" in
   "Ferrari") echo "Ferrari is quite expansive."
   ;;
   "Jaguar") echo "I like Jaguar."
   ;;
   "Honda") echo "Honda is the car of the year."
```

```
    ;;
esac
```

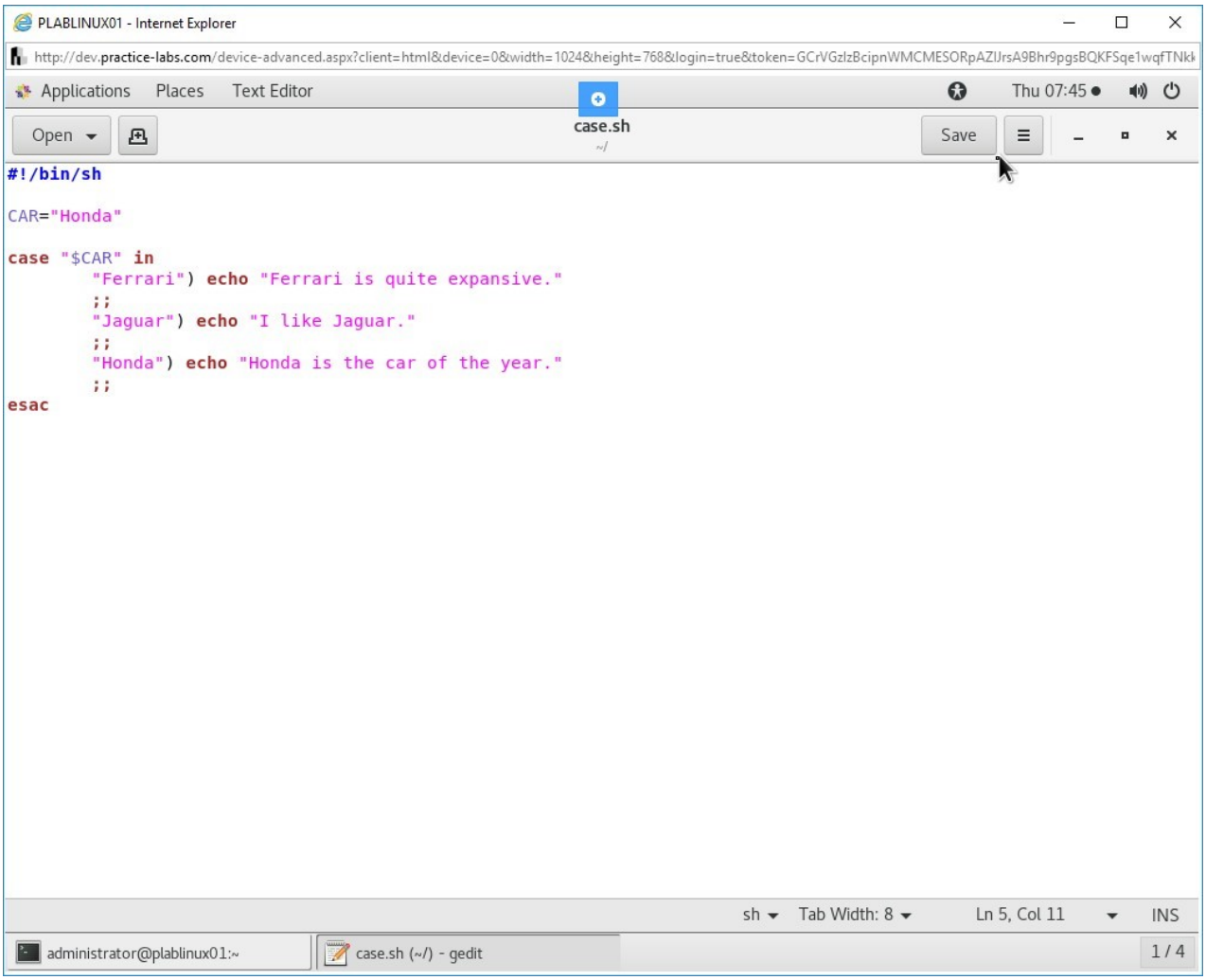


Figure 1.7 Screenshot of PLABLINUX01: Entering the content in a new file.

Step 8

Click **Save** to save the file. Then, close the file.

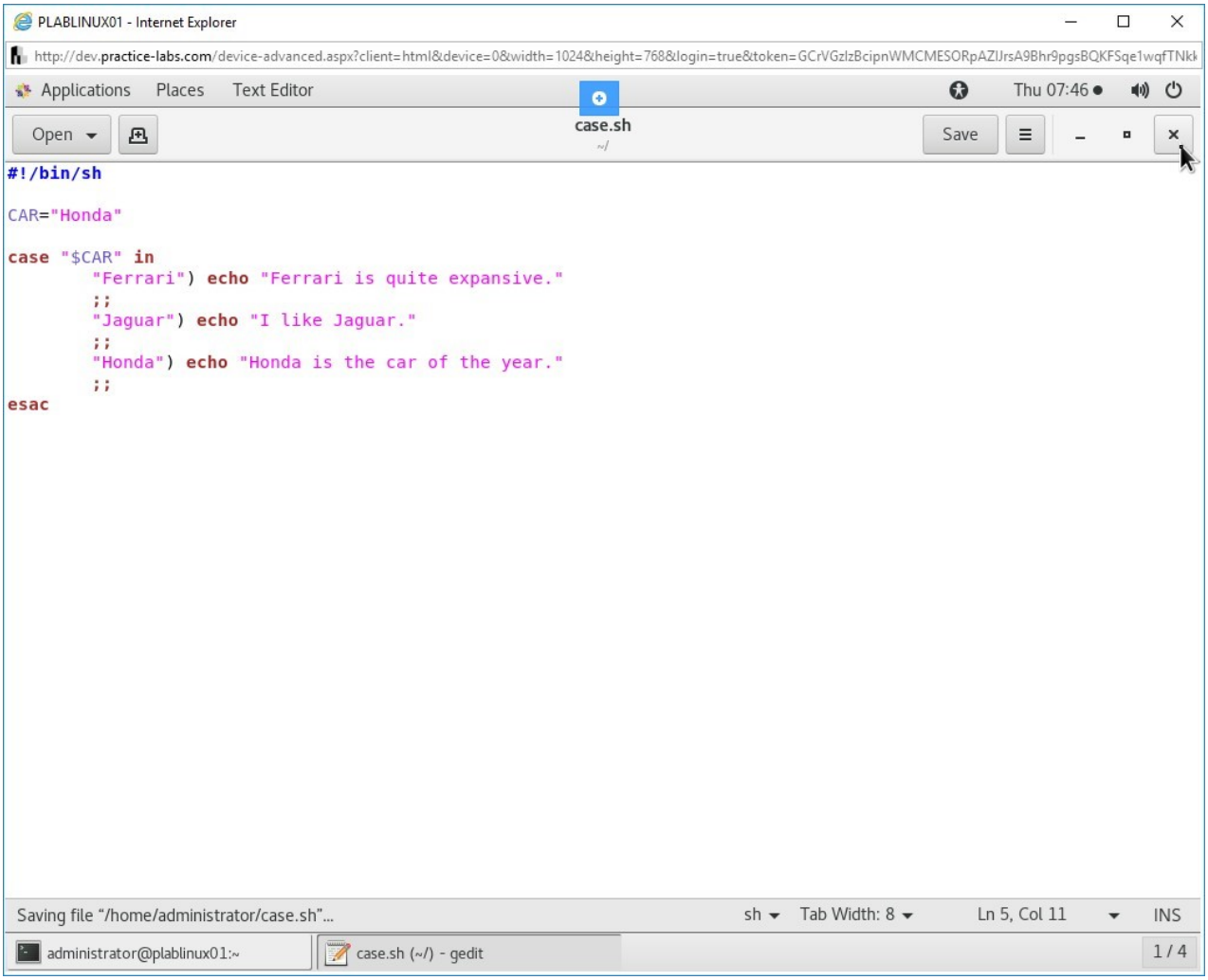


Figure 1.8 Screenshot of PLABLINUX01: Saving and closing the file.

Step 9

Clear the screen by entering the following command:

```
clear
```

You need to make the shell script executable. Type the following command:

```
sudo chmod +x case.sh
```

Press **Enter**.

When prompted, type the following password:

`Passw0rd`

Press **Enter**.

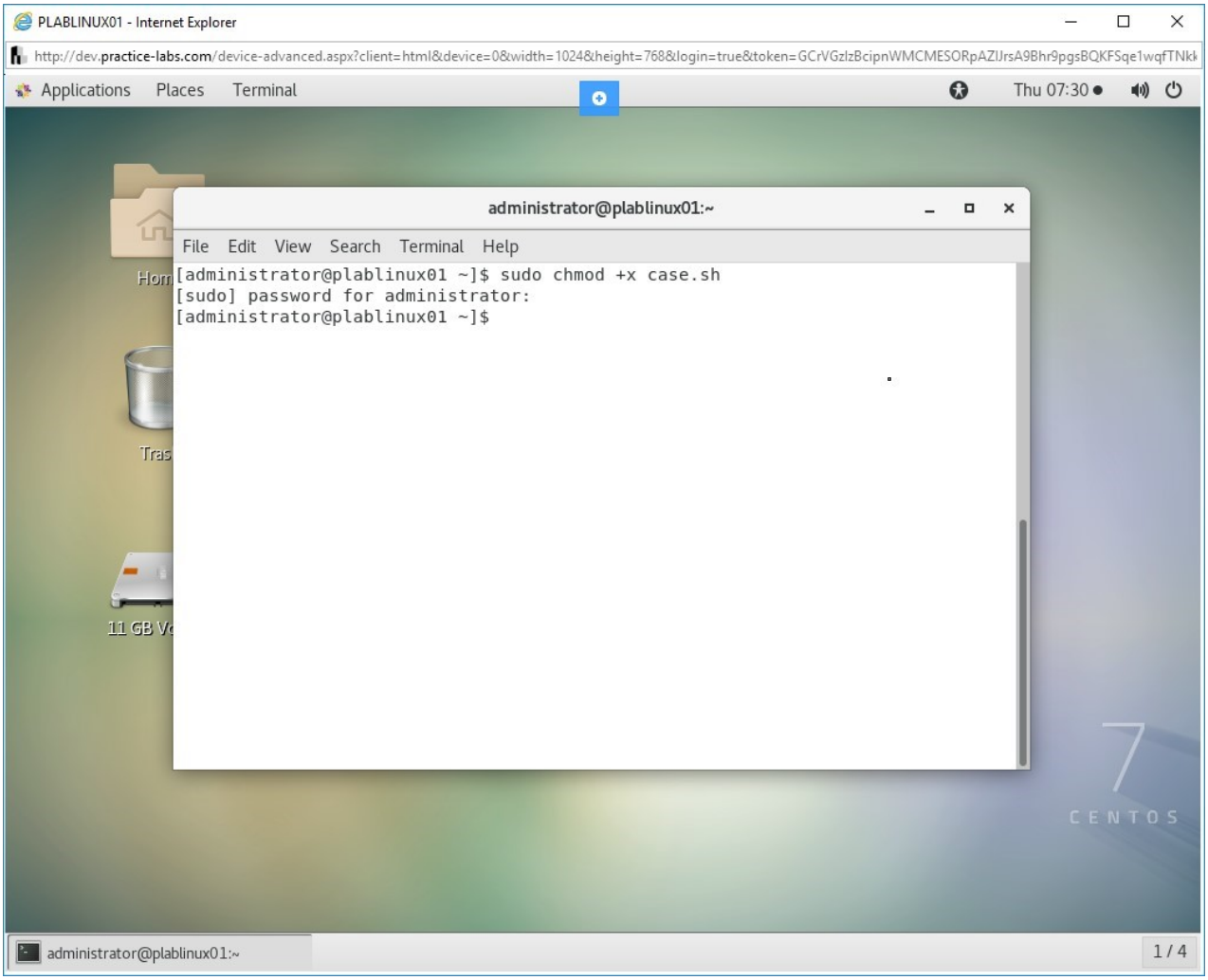


Figure 1.9 Screenshot of PLABLINUX01: Making the file executable.

Step 10

Clear the screen by entering the following command:

```
clear
```

You will now execute the shell script. Type the following command:

```
./case.sh
```

Press **Enter**. Notice the answer. After evaluating two conditions, the third condition meets the criteria and is printed as the answer.

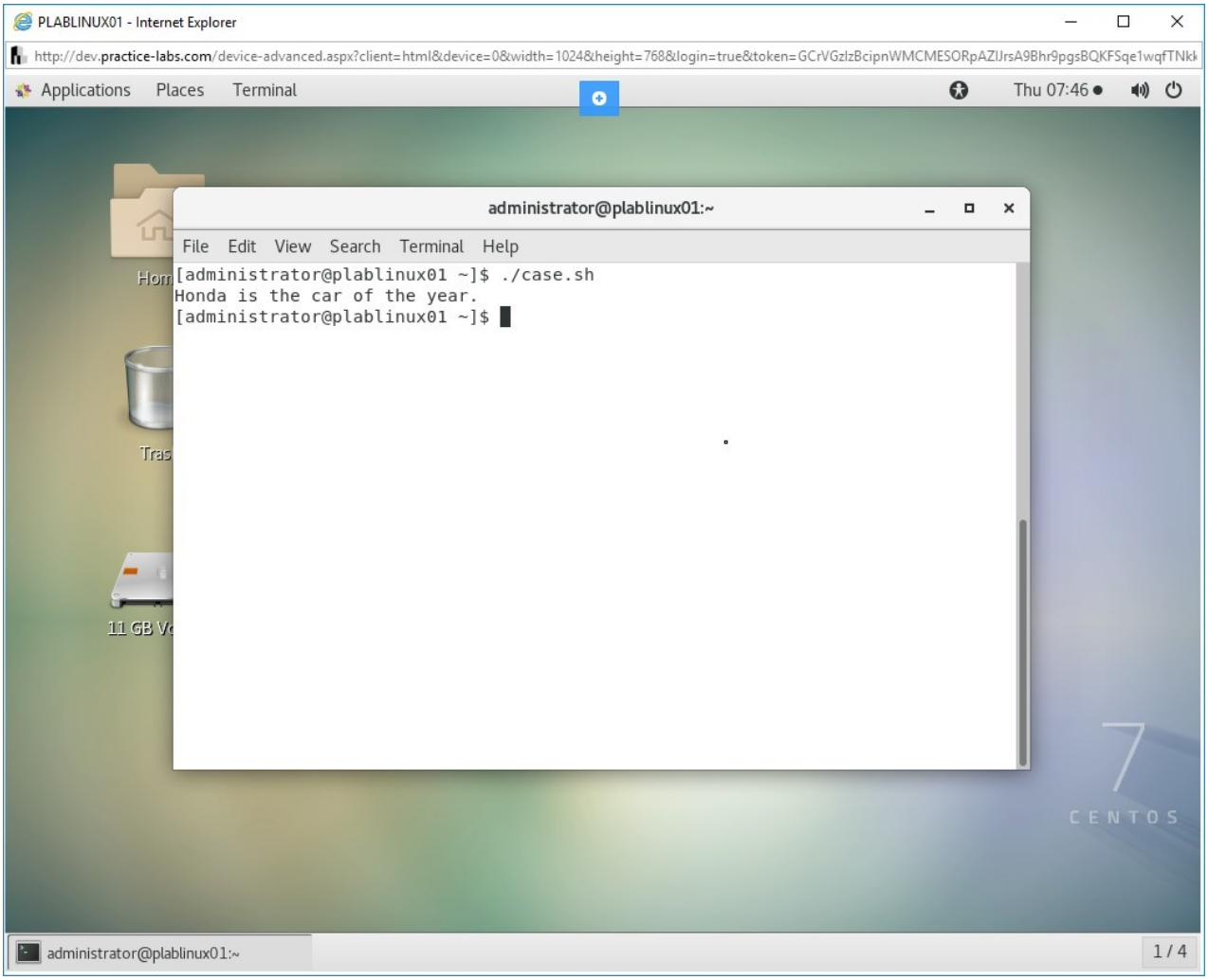


Figure 1.10 Screenshot of PLABLINUX01: Executing the shell script.

Task 2 - Use While Loops

A while loop runs until a statement becomes true. It can nest other loops. There is no limit to nesting loops. The number is limited to your requirement.

In this task, you will learn to use while loop. To use the loop, perform the following steps:

Step 1

Clear the screen by entering the following command:

```
clear
```

Let's first create a new shell script with the following command:

```
gedit while.sh
```

Press **Enter**.

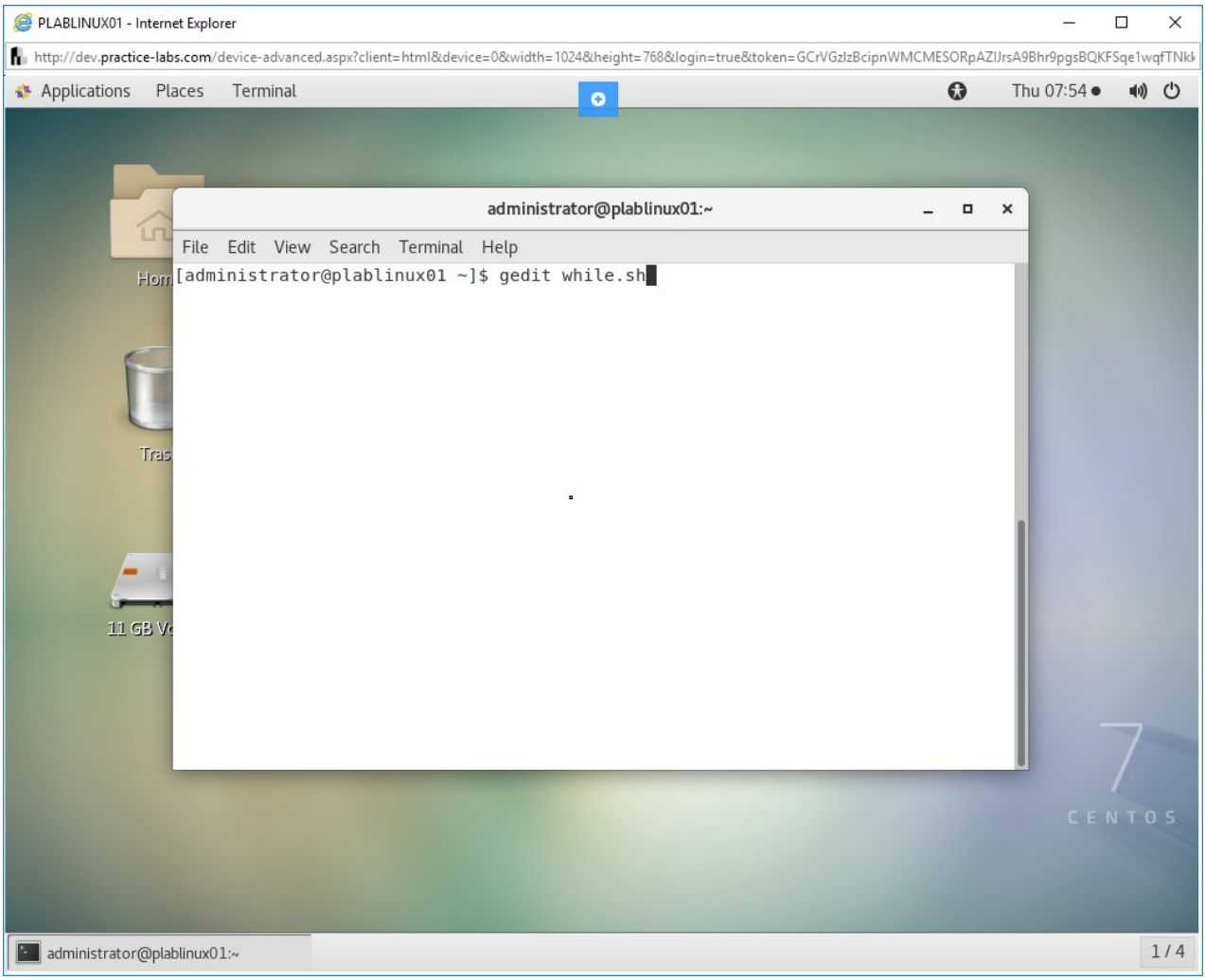


Figure 1.11 Screenshot of PLABLINUX01: Opening a new file in gedit.

Step 2

Notice a new file is created with the name **while.sh**. Type the following:

```
#!/bin/bash
a=0
while [ $a -le 10 ]
do
  echo "Welcome $a times."
```

```
    a=$(( a+1 ))
done
```

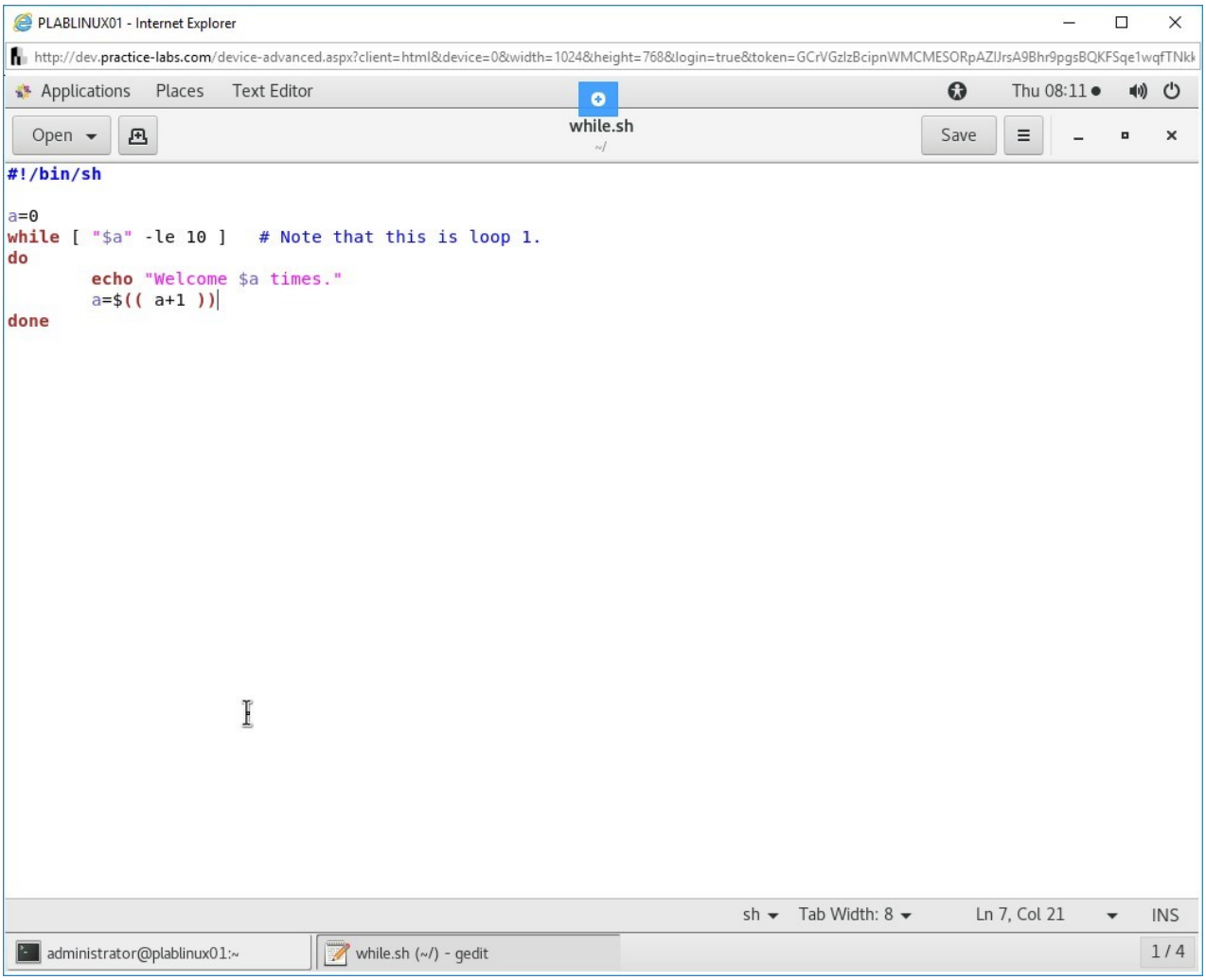


Figure 1.12 Screenshot of PLABLINUX01: Entering the content in the file.

Step 3

Click **Save** to save the file. Then, close the file.

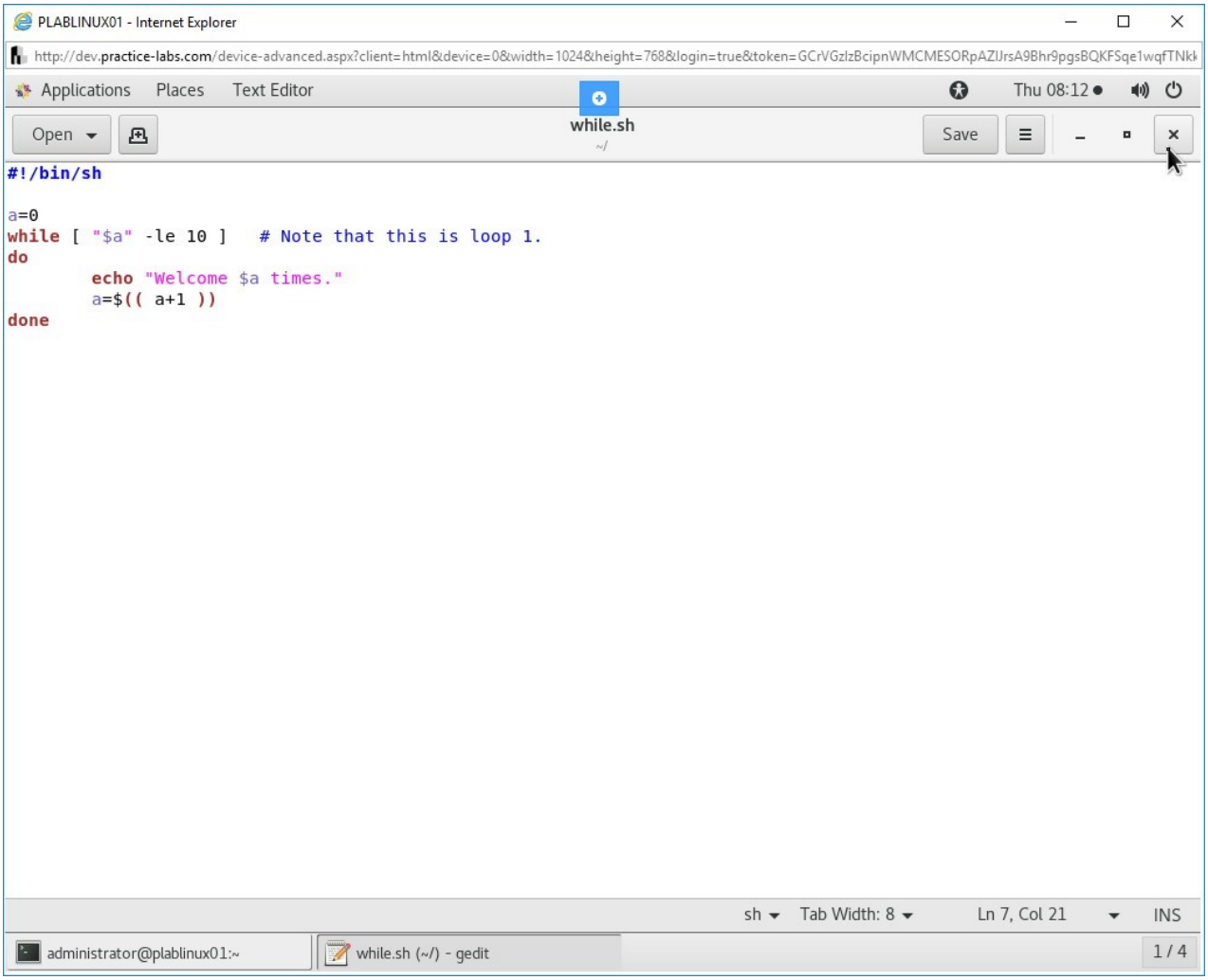


Figure 1.13 Screenshot of PLABLINUX01: Saving and closing the file.

Step 4

Clear the screen by entering the following command:

```
clear
```

You need to make the shell script executable. Type the following command:

```
sudo chmod +x while.sh
```

Press **Enter**.

When prompted, type the following password:

`Passw0rd`

Press **Enter**.

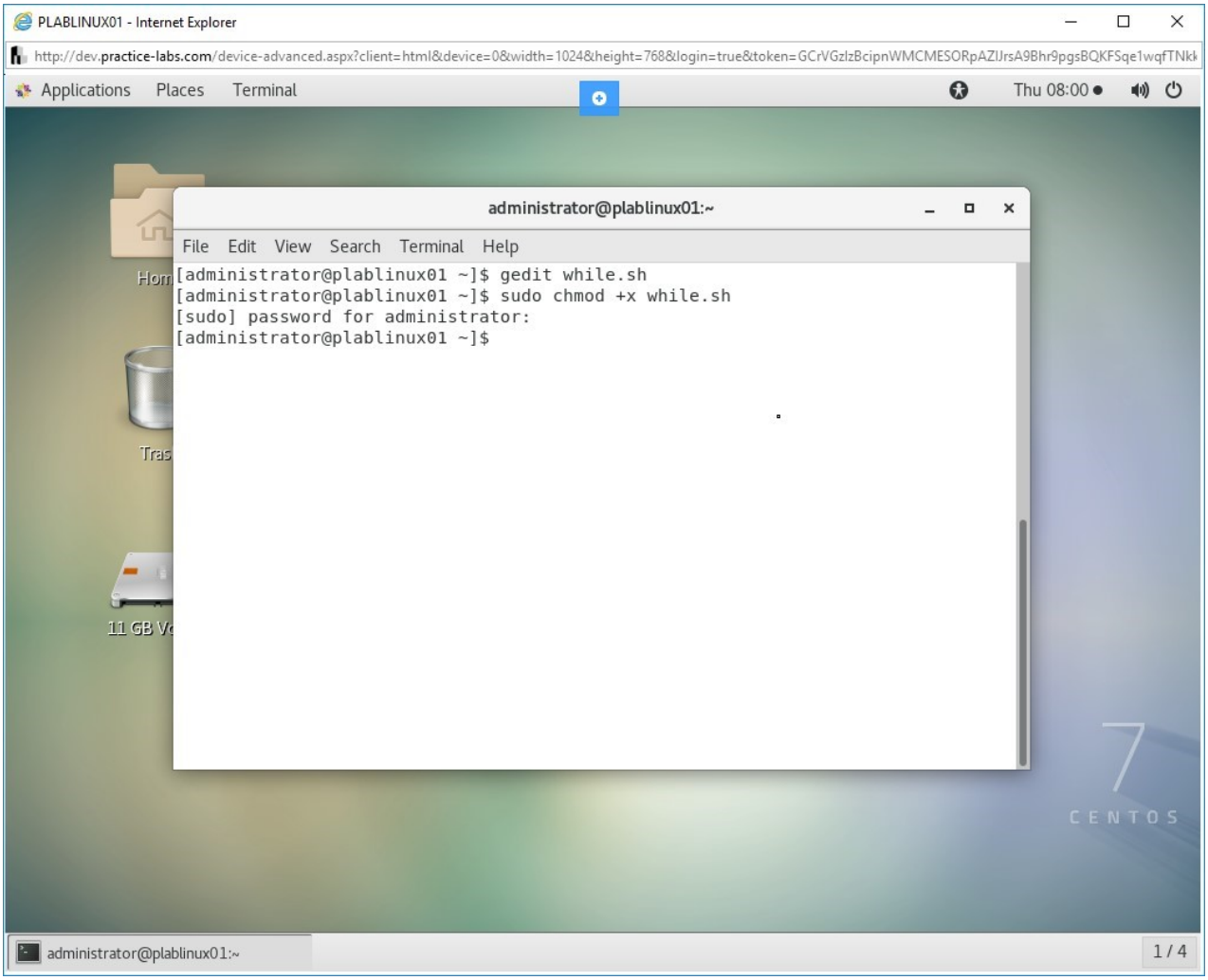


Figure 1.14 Screenshot of PLABLINUX01: Making the file executable.

Step 5

Clear the screen by entering the following command:

```
clear
```

You will now execute the shell script. Type the following command:

```
./while.sh
```

Press **Enter**. Note that the last condition is met in which x is less than z.

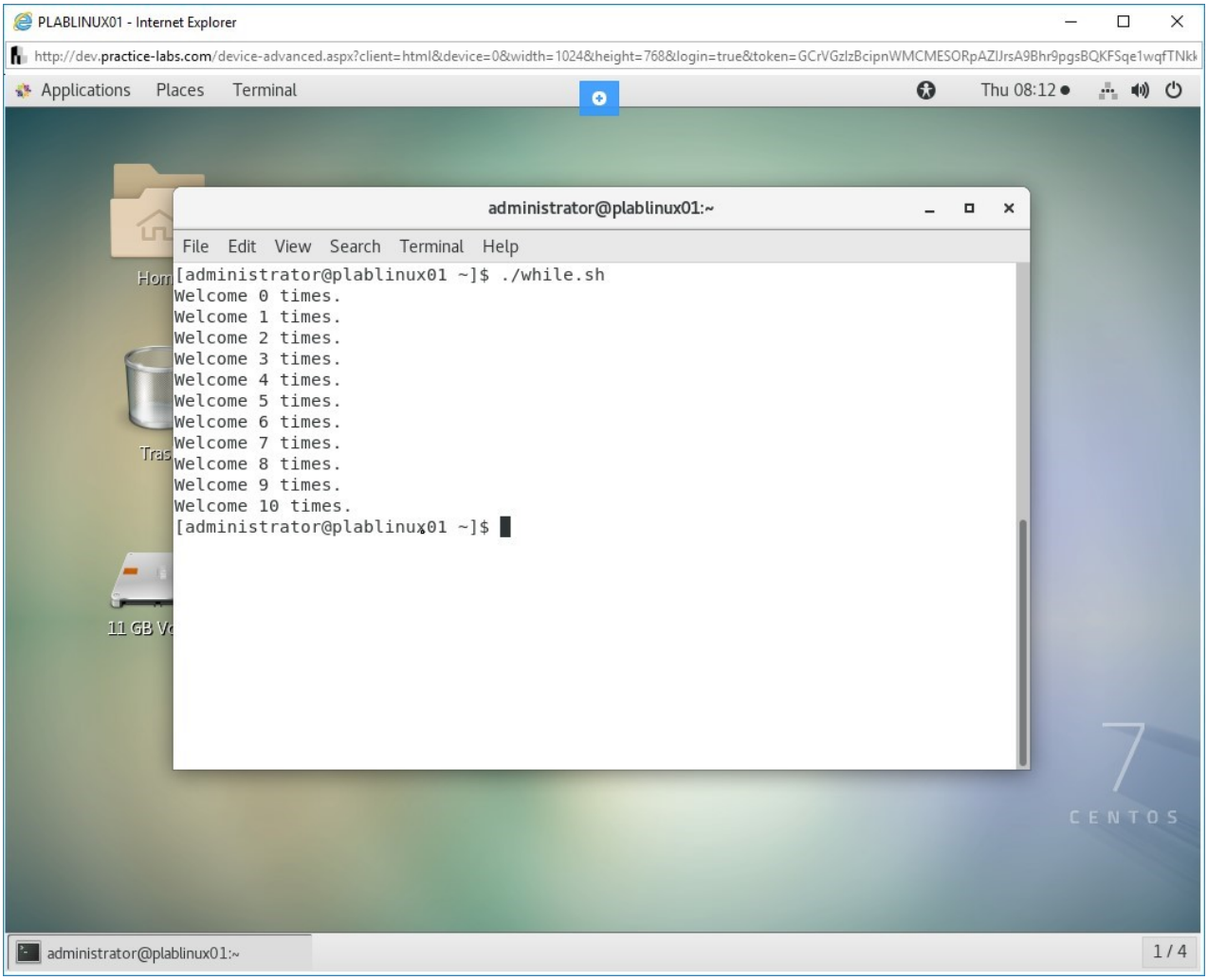


Figure 1.15 Screenshot of PLABLINUX01: Executing the shell script.

Task 3 - Use Until Loops

An until loop runs until a statement becomes true.

In this task, you will learn to use until loop. To use the until loop, perform the following steps:

Step 1

Clear the screen by entering the following command:

```
clear
```

Let’s first create a new shell script with the following command:

```
gedit until.sh
```

Press **Enter**.

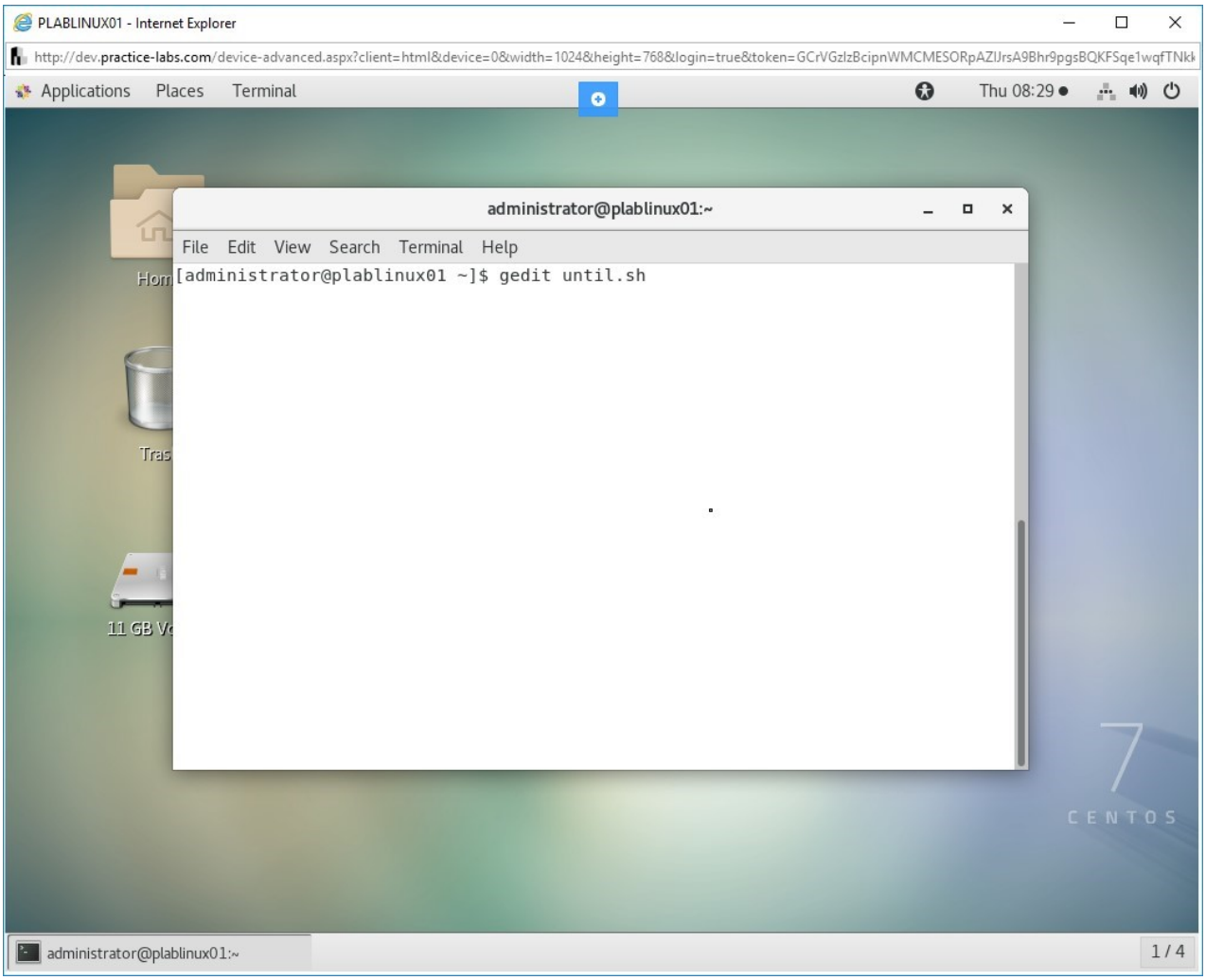


Figure 1.16 Screenshot of PLABLINUX01: Creating a new file in gedit.

Step 2

Notice a new file is created with the name **until.sh**. Type the following:

```
#!/bin/sh
a=1
until [ $a -gt 6 ]
do
  echo $a
```

```
    a=$(( a+1 ))
done
```

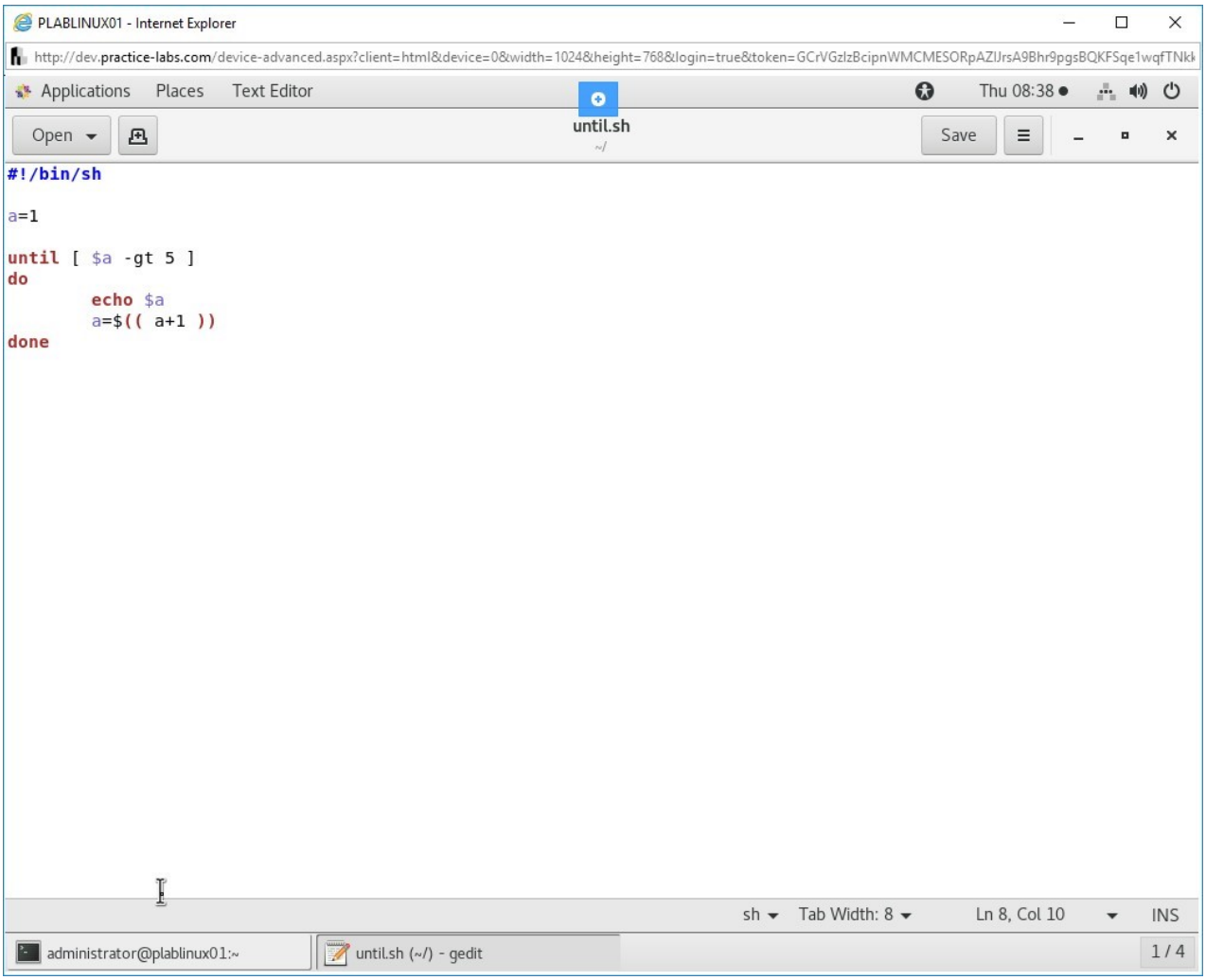


Figure 1.17 Screenshot of PLABLINUX01: Entering the content in the file.

Step 3

Click **Save** to save the file. Then, close the file.

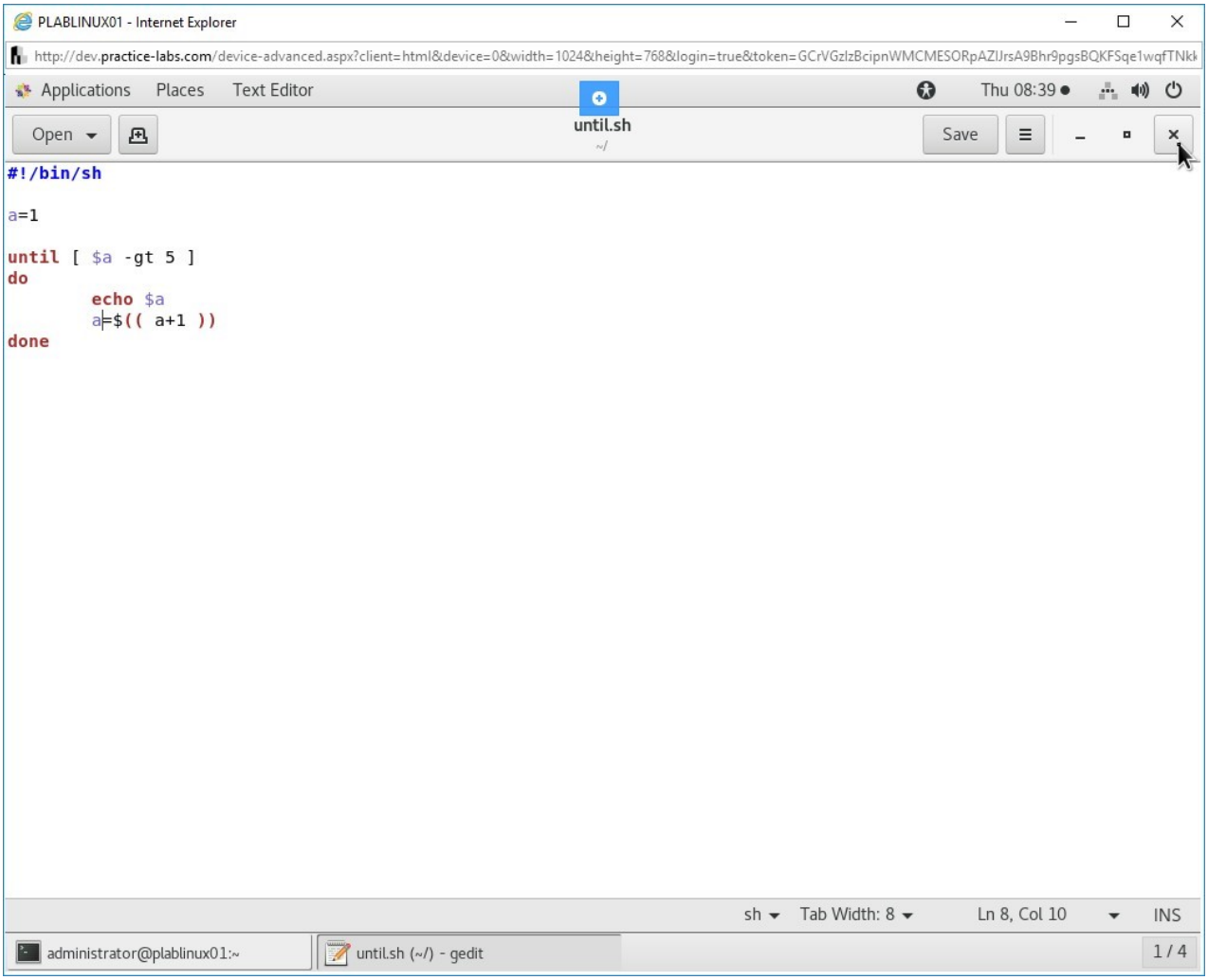


Figure 1.18 Screenshot of PLABLINUX01: Saving and closing the file.

Step 4

Clear the screen by entering the following command:

```
clear
```

You need to make the shell script executable. Type the following command:

```
sudo chmod +x until.sh
```

Press **Enter**.

When prompted, type the following password:

`Passw0rd`

Press **Enter**.

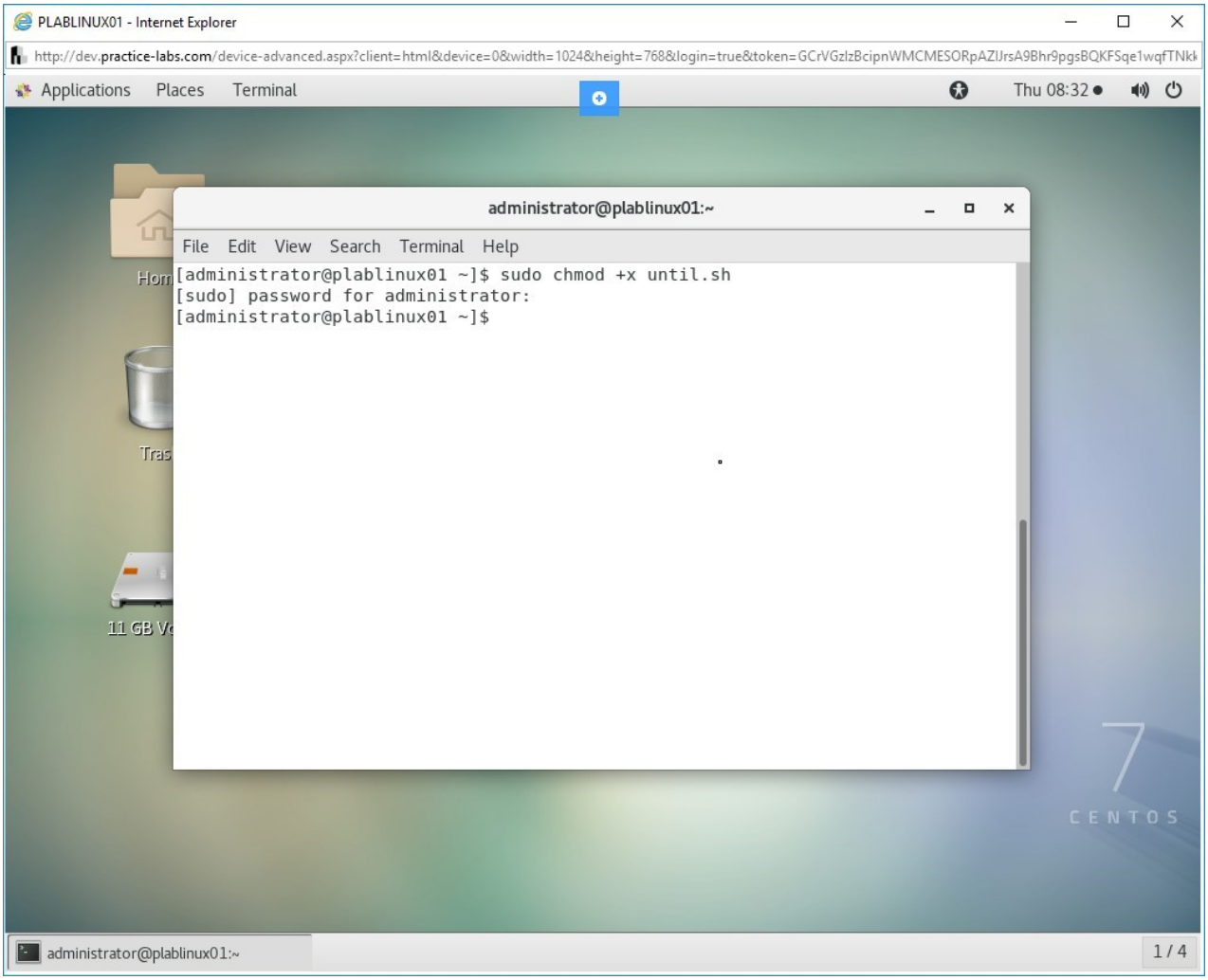


Figure 1.19 Screenshot of PLABLINUX01: Making the file executable.

Step 5

Clear the screen by entering the following command:

```
clear
```

You will now execute the shell script. Type the following command:

```
./until.sh
```

Press **Enter**.

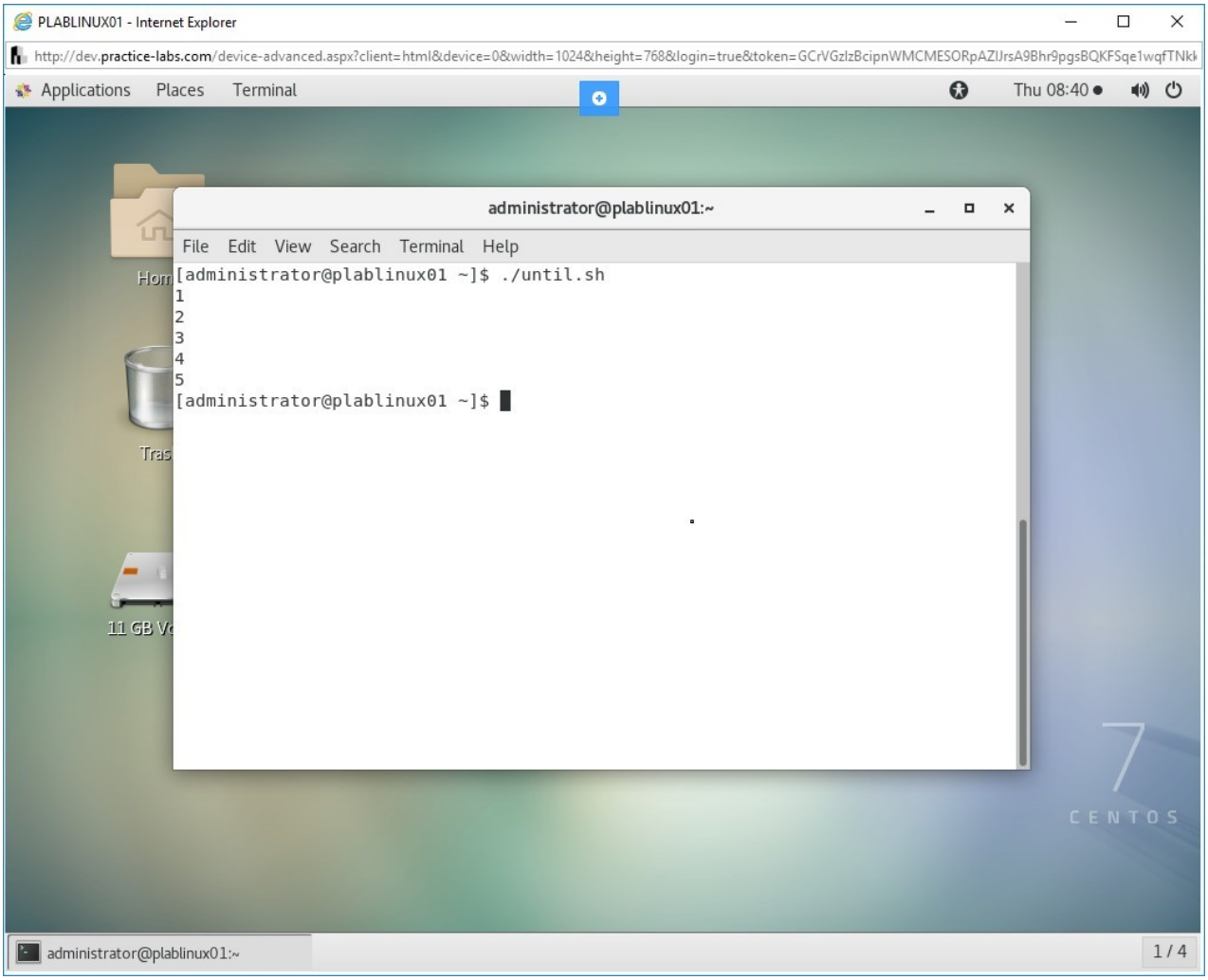


Figure 1.20 Screenshot of PLABLINUX01: Executing the shell script.

Task 4 - Use for Loops

The for loop runs through a list of values in a list until the time list values are exhausted.

In this task, you will learn to use for loop. To use the for loop, perform the following steps:

Step 1

Clear the screen by entering the following command:

```
clear
```

Let's first create a new shell script with the following command:

```
gedit for.sh
```

Press **Enter**.

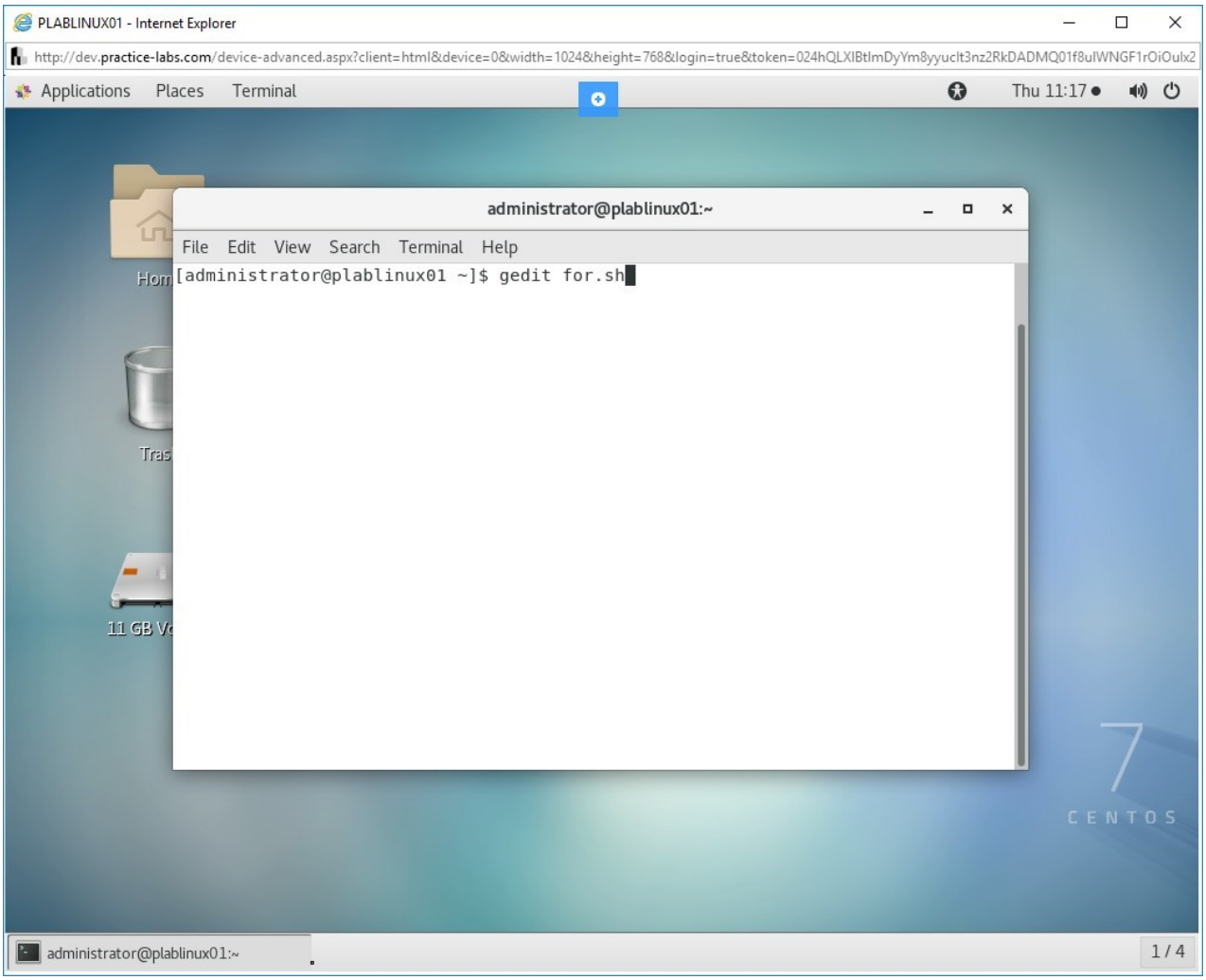


Figure 1.21 Screenshot of PLABLINUX01: Creating a new file in gedit.

Step 2

Notice a new file is created with the name **for.sh**. Type the following:

```
#!/bin/bash
echo "Files in the /etc directory are:"
for f in $(ls /etc/*)
do
```

```
    echo $f
done
```

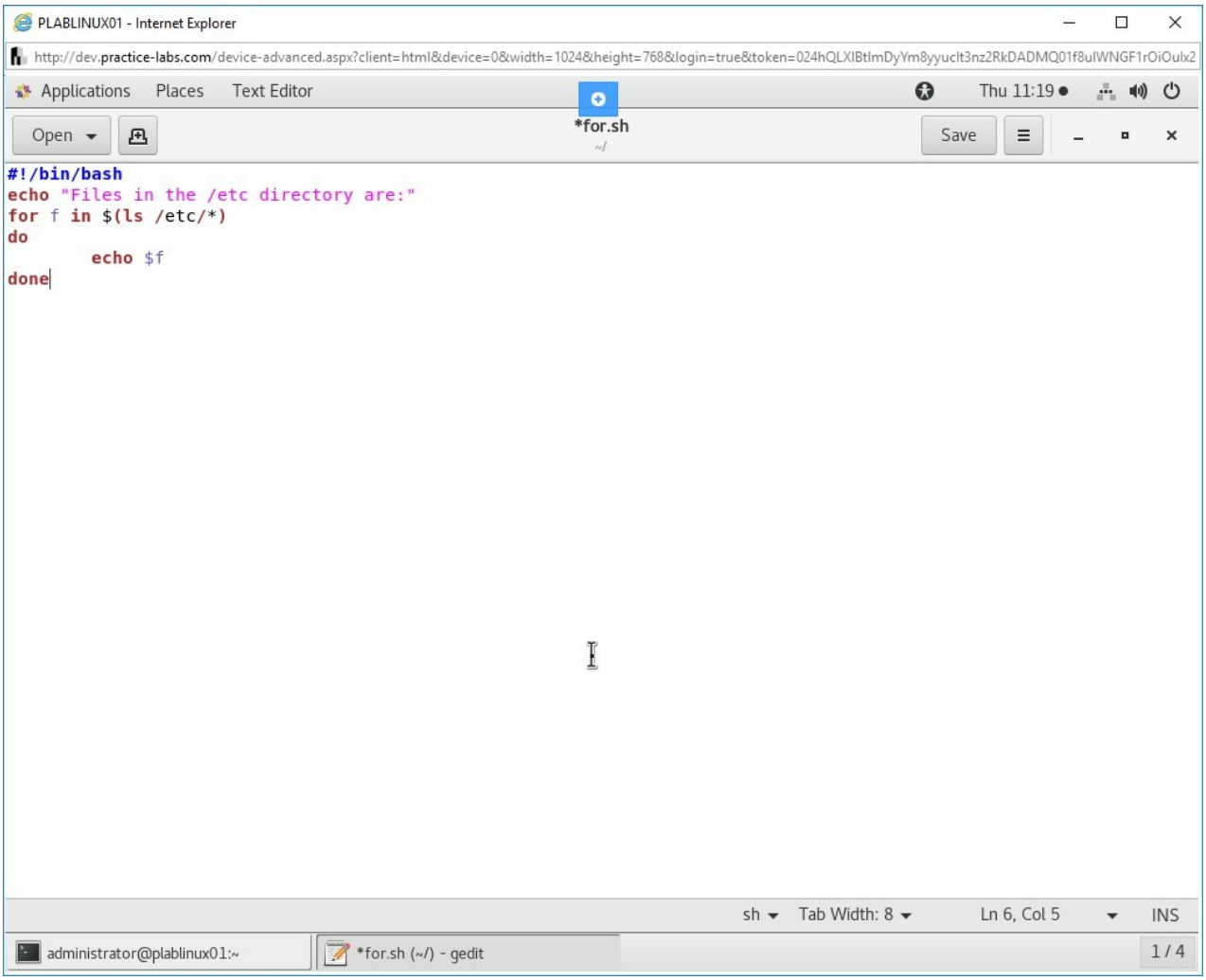


Figure 1.22 Screenshot of PLABLINUX01: Entering the content in the file.

Step 3

Click **Save** to save the file. Then, close the file.

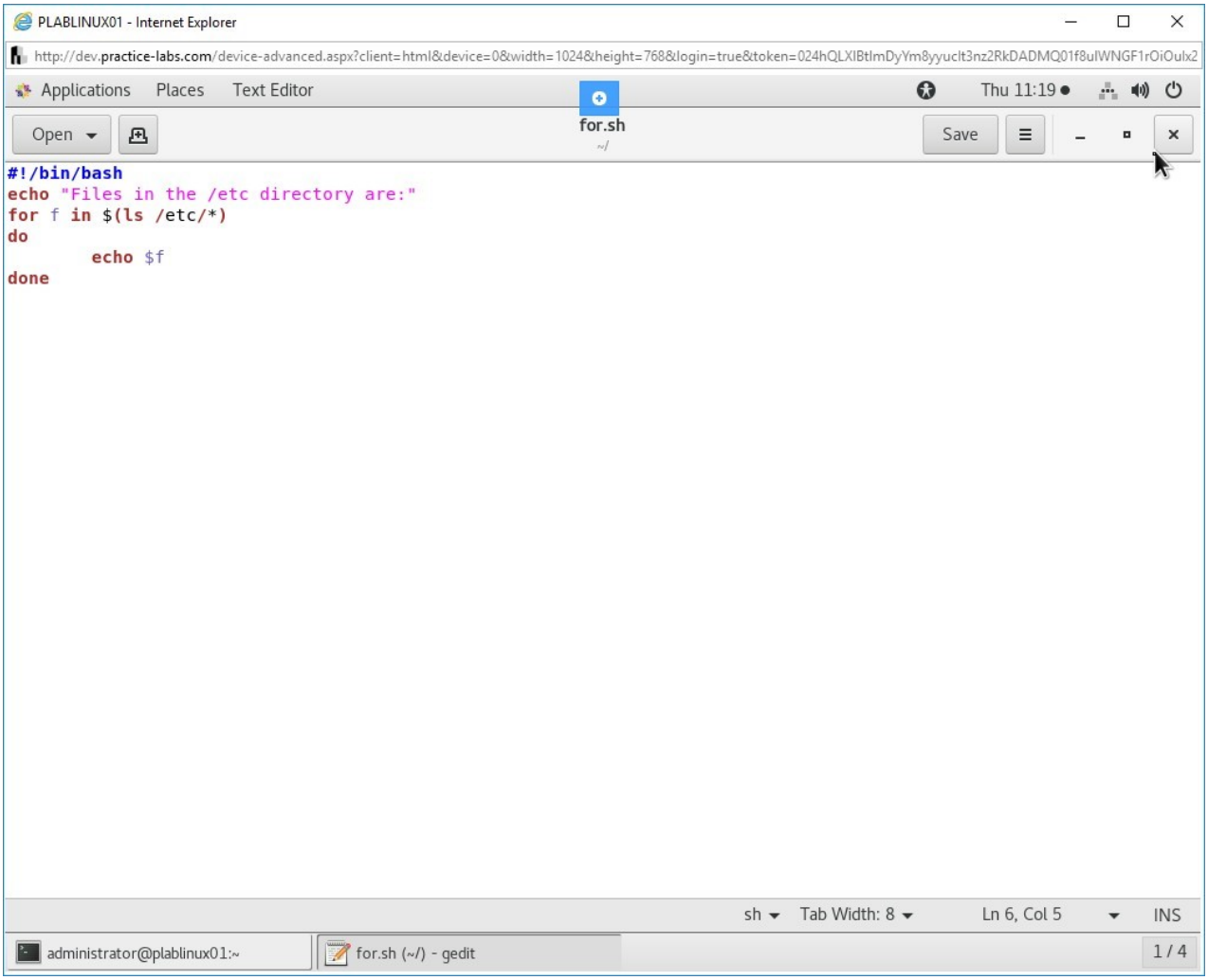


Figure 1.23 Screenshot of PLABLINUX01: Saving and closing the file.

Step 4

Clear the screen by entering the following command:

```
clear
```

You need to make the shell script executable. Type the following command:

```
sudo chmod +x for.sh
```

Press **Enter**.

When prompted, type the following password:

`Passw0rd`

Press **Enter**.

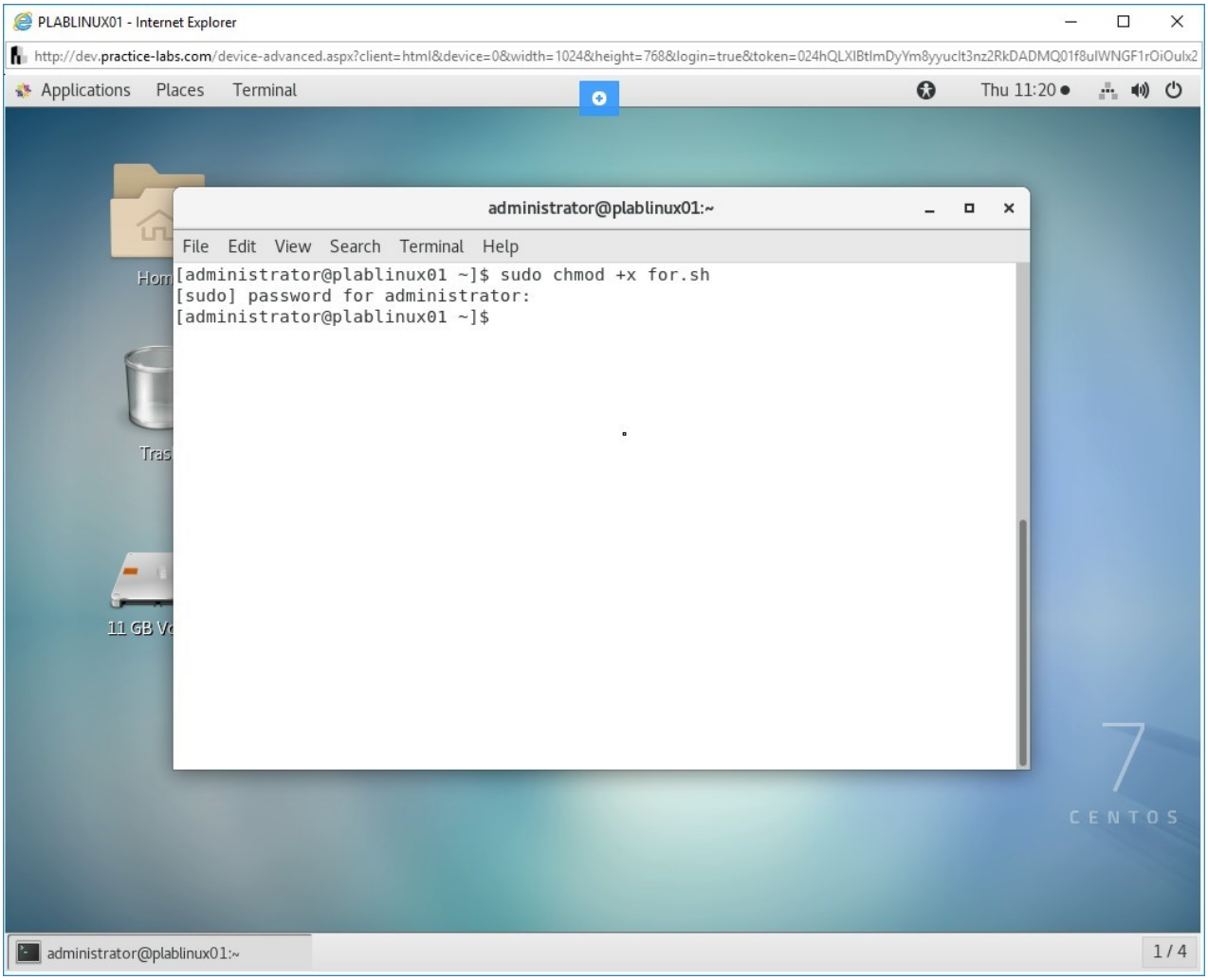


Figure 1.24 Screenshot of PLABLINUX01: Making the file executable.

Step 5

Clear the screen by entering the following command:

```
clear
```

You will now execute the shell script. Type the following command:

```
./for.sh
```

Press **Enter**.

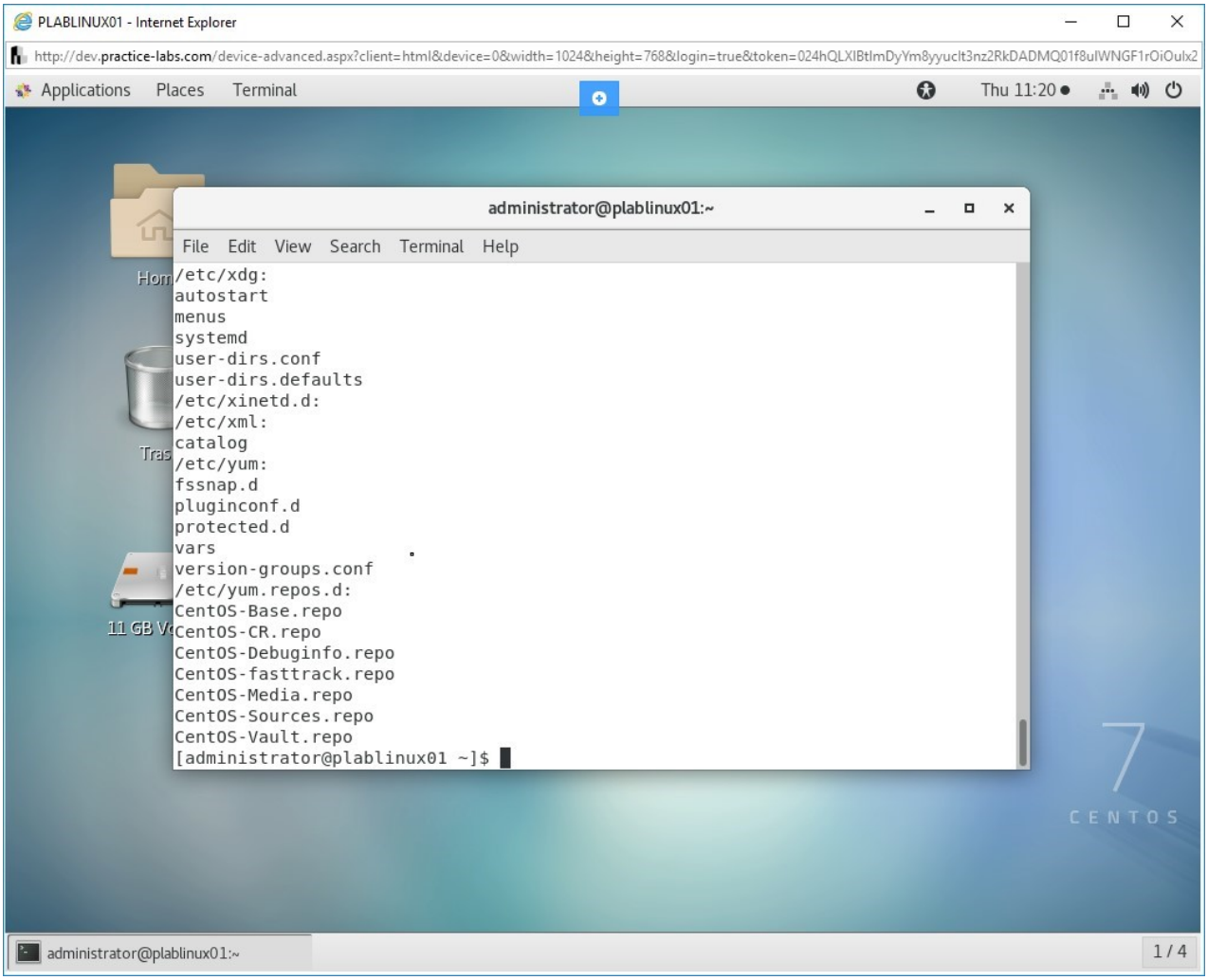


Figure 1.25 Screenshot of PLABLINUX01: Executing the shell script.

Keep all devices in their current state and proceed to the next exercise.

Review

Well done, you have completed the **Work with the Flow Control Constructs** Practice Lab.

Summary

You completed the following exercise:

- Exercise 1 - Work with the flow control constructs

You should now be able to:

- Use the if…else statement
- Use the while loop
- Use the until loop
- Use the for loop

Feedback

Shutdown all virtual machines used in this lab. Alternatively, you can log out of the lab platform.