

# TryHackMe Lab Report

**Name:** Pelumi Johnson

**Date:** 01/04/2026

**Course / Platform:** TryHackMe

**Room:** Linux Fundamentals Part 1

**Operating System:** Ubuntu 20.04 LTS

## Objective

The objective of this lab was to build foundational Linux command-line skills by interacting directly with a live Linux operating system. The lab focused on understanding Linux basics, navigating the filesystem, managing files, searching logs, and controlling command execution using operators.

All tasks were completed using a real Ubuntu Linux machine deployed in-browser through TryHackMe.

## Tools & Environment Used

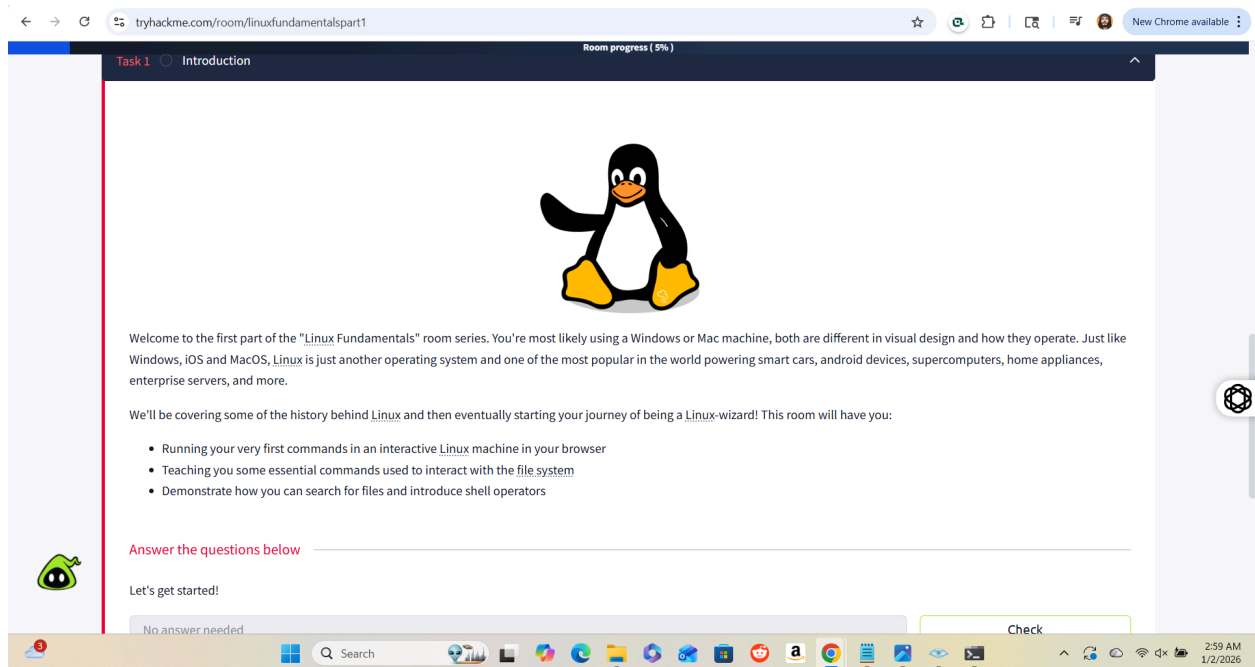
- Ubuntu 20.04 LTS (TryHackMe In-Browser Machine)
- Bash Shell
- Linux Command Line Utilities (echo, whoami, ls, cd, pwd, cat, grep)
- Shell Operators (>, >>, &&, &)

## Step 1: Introduction to Linux

Linux was introduced as an open-source operating system widely used across servers, cloud environments, embedded systems, websites, point-of-sale systems, and critical infrastructure. Unlike Windows and macOS, Linux is commonly interacted with through the command line, especially in professional and security-focused environments.

I learned that Linux is not a single operating system, but an umbrella term for many distributions (distros) such as Ubuntu and Debian. For this lab, Ubuntu was used.

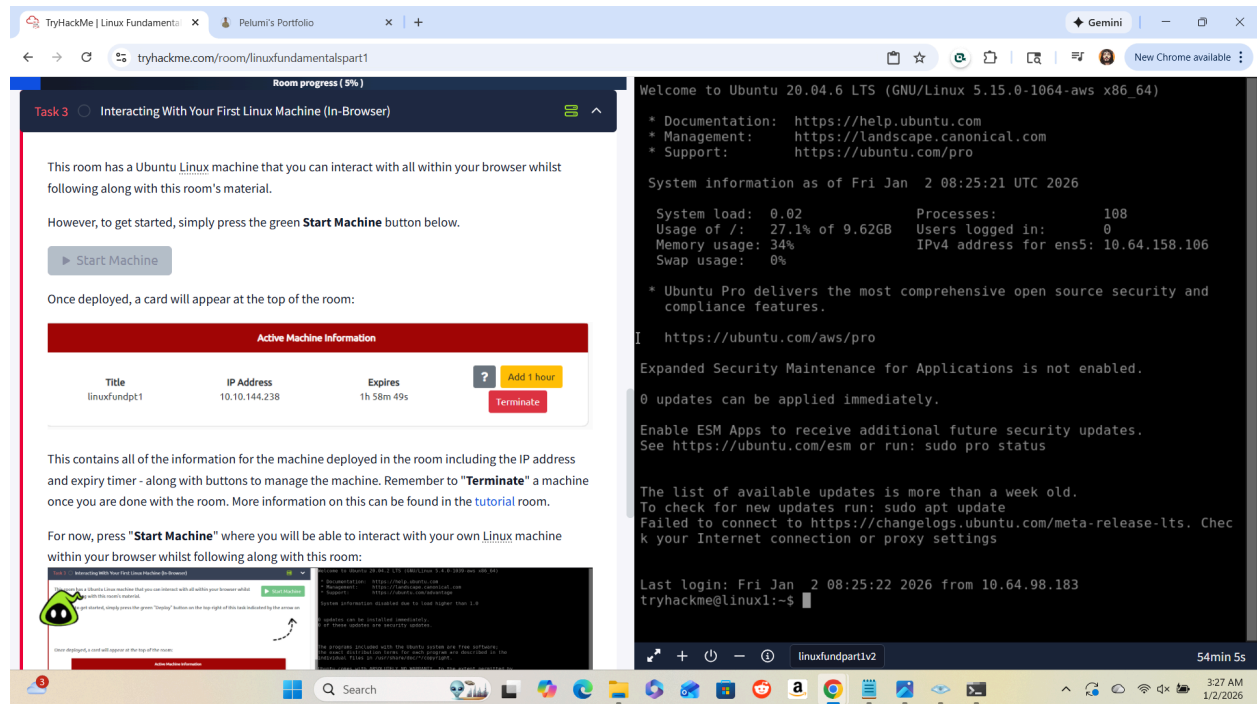
I also researched the history of Linux and confirmed that the first Linux operating system was released in 1991.



## Step 2: Deploying the Linux Machine

I deployed the Ubuntu Linux machine provided by TryHackMe directly in the browser. Once deployed, I was given access to a live terminal session.

This demonstrated how Linux servers are commonly accessed remotely and reinforced the importance of terminal-based interaction in real-world environments.



## Step 3: Basic Command Interaction

I practiced issuing basic commands to understand how the Linux terminal works.

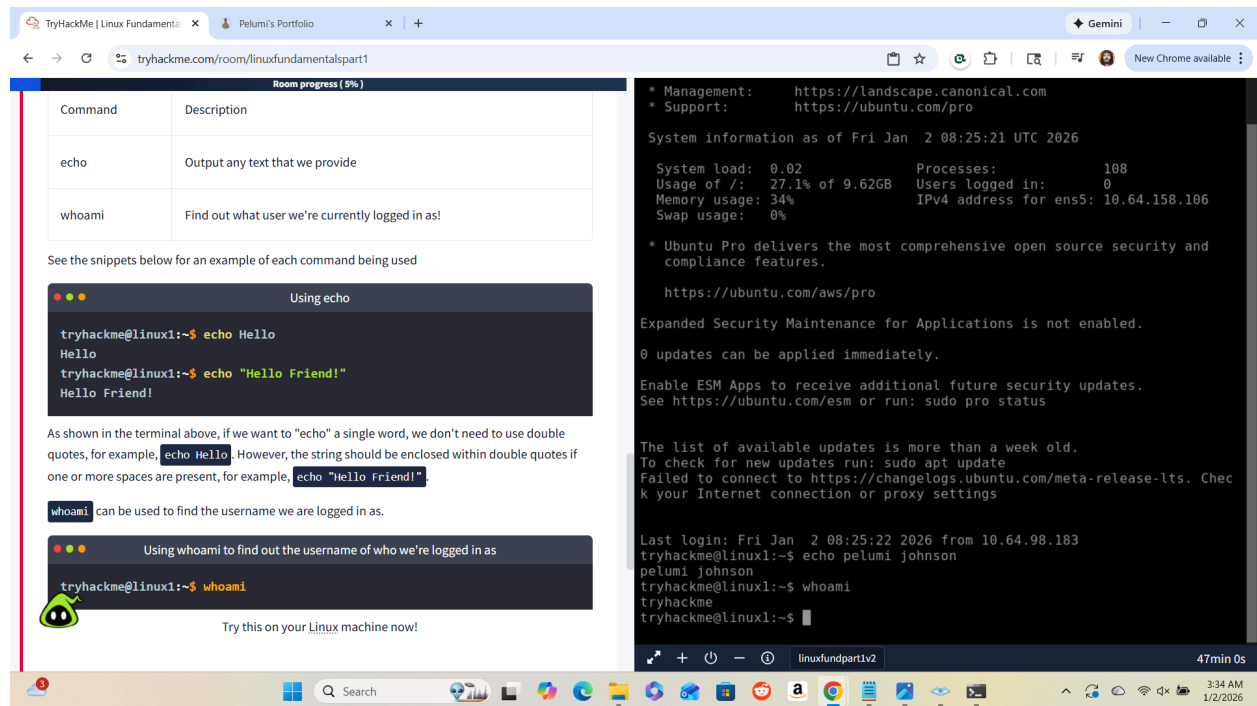
### Commands used:

- Echo
- whoami

Using the echo command, I printed text to the terminal to confirm command execution. Using the whoami command, I identified the active user account.

The output confirmed that I was logged in as the user:  
tryhackme

This step reinforced the importance of understanding user context when operating within Linux systems.



## Step 4: Navigating the Linux Filesystem

I learned how Linux organizes files and directories and how to navigate the filesystem using terminal commands.

### Commands used:

- Pwd
- Ls
- Cd
- cd ..

Using pwd, I displayed the full path of my current working directory.

Using ls, I listed files and folders within directories.

Using cd, I moved into specific directories.

Using cd .., I returned to the parent directory.

This helped me understand Linux directory structure and paths such as /home/tryhackme.

The screenshot shows a web browser with two tabs: 'TryHackMe | Linux Fundamentals' and 'Pelumi's Portfolio'. The active tab is 'tryhackme.com/room/linuxfundamentalspart1'. The page content includes a 'Room progress (30%)' indicator, a paragraph about navigating the Linux machine, and a section titled 'Using "pwd" to list the full path of the current directory'. Below this is a terminal window showing the command 'pwd' being executed, resulting in the path '/home/ubuntu/Documents'. A list of instructions follows, explaining the use of 'pwd' and 'cd'. At the bottom, there is a section 'Answer the questions below' and a small green alien icon.

Room progress (30%)

You'll notice as you progress through navigating your Linux machine, the name of the directory that you are currently working in will be listed in your terminal.

It's easy to lose track of where we are on the filesystem exactly, which is why I want to introduce "pwd". This stands for **print working directory**.

Using the example machine from before, we are currently in the "Documents" folder — but where is this exactly on the Linux machine's filesystem? We can find this out using this "pwd" command like within the screenshot below:

```
tryhackme@linux1:~/Documents$ pwd
/home/ubuntu/Documents
tryhackme@linux1:~/Documents$
```

Using "pwd" to list the full path of the current directory

Let's break this down:

1. We already know we're in "Documents" thanks to our terminal, but at this point in time, we have no idea where "Documents" is stored so that we can get back to it easily in the future.
2. I have used the "pwd" (print working directory) command to find the full file path of this "Documents" folder.
3. We're helpfully told by Linux that this "Documents" directory is stored at "/home/ubuntu/Documents" on the machine — great to know!
4. Now in the future, if we find ourselves in a different location, we can just use `cd /home/ubuntu/Documents` to change our working directory to this "Documents" directory.

Answer the questions below

```
/home/tryhackme
tryhackme@linux1:~$ ls
Pelumi.txt  access.log  folder1  folder2  folder3  folder4  pelumi.txt
tryhackme@linux1:~$ cd ..
tryhackme@linux1:/home$ ls
tryhackme  ubuntu
tryhackme@linux1:/home$ ls tryhackme
Pelumi.txt  access.log  folder1  folder2  folder3  folder4  pelumi.txt
tryhackme@linux1:/home$ cd tryhackme
tryhackme@linux1:~$ ls
Pelumi.txt  access.log  folder1  folder2  folder3  folder4  pelumi.txt
tryhackme@linux1:~$ cat pelumi.txt
Pelumi
tryhackme@linux1:~$ cd ..
tryhackme@linux1:/home$ ls
tryhackme  ubuntu
tryhackme@linux1:/home$ cd ubuntu
tryhackme@linux1:/home/ubuntu$ ls
tryhackme@linux1:/home/ubuntu$ ls
tryhackme@linux1:/home/ubuntu$ cd ..
tryhackme@linux1:/home$ cd tryhackme
tryhackme@linux1:~$ pwd
/home/tryhackme
tryhackme@linux1:~$ cd /home/ubuntu
tryhackme@linux1:/home/ubuntu$ cd /home/tryhackme
tryhackme@linux1:~$ ls
Pelumi.txt  access.log  folder1  folder2  folder3  folder4  pelumi.txt
tryhackme@linux1:~$ cd folder4
tryhackme@linux1:~/folder4$ ls
note.txt
tryhackme@linux1:~/folder4$ cat note.txt
Hello World!
tryhackme@linux1:~/folder4$ pwd
/home/tryhackme/folder4
tryhackme@linux1:~/folder4$
```

## Step 5: Creating and Viewing Files

I practiced creating files and viewing their contents directly from the command line.

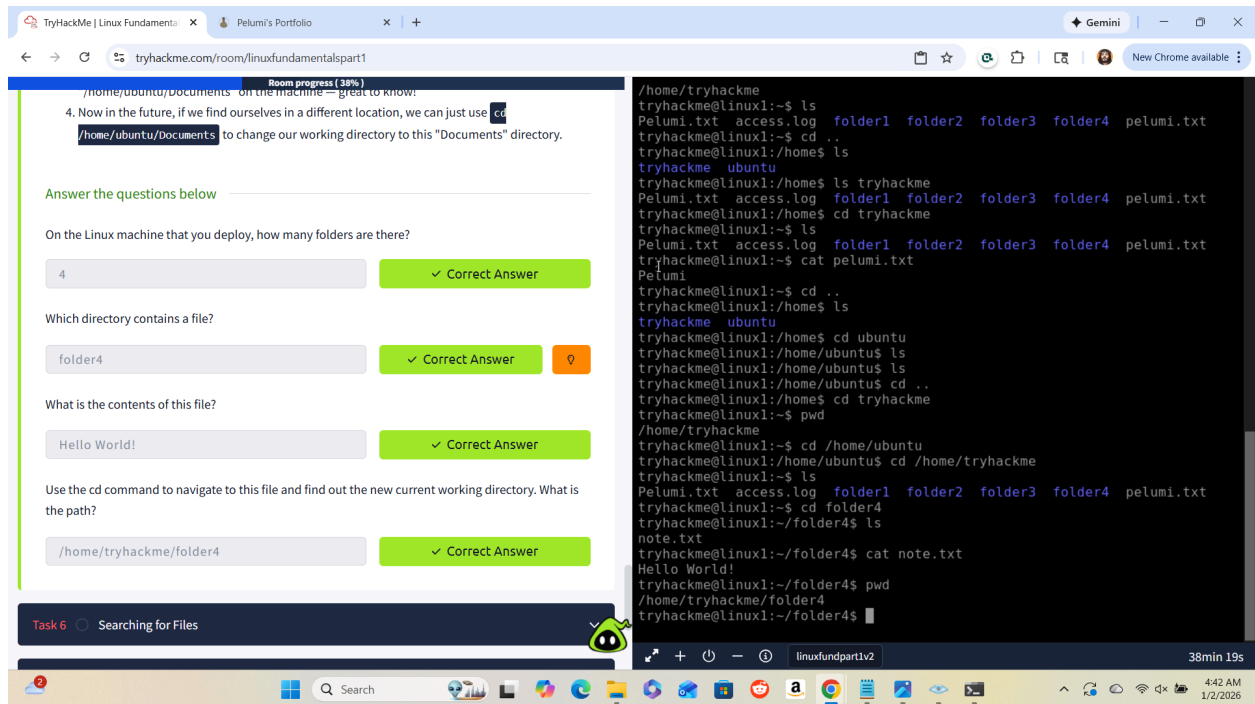
### Commands used:

- `echo "text" > filename`
- `cat filename`

Using output redirection (>), I created files and wrote text into them.

Using cat, I displayed the contents of files directly in the terminal.

This demonstrated how Linux treats nearly everything as a file and how command output can be redirected into files.



## Step 6: Searching Files and Logs with grep

I learned how to search through files efficiently using the grep command, which is essential when working with log files.

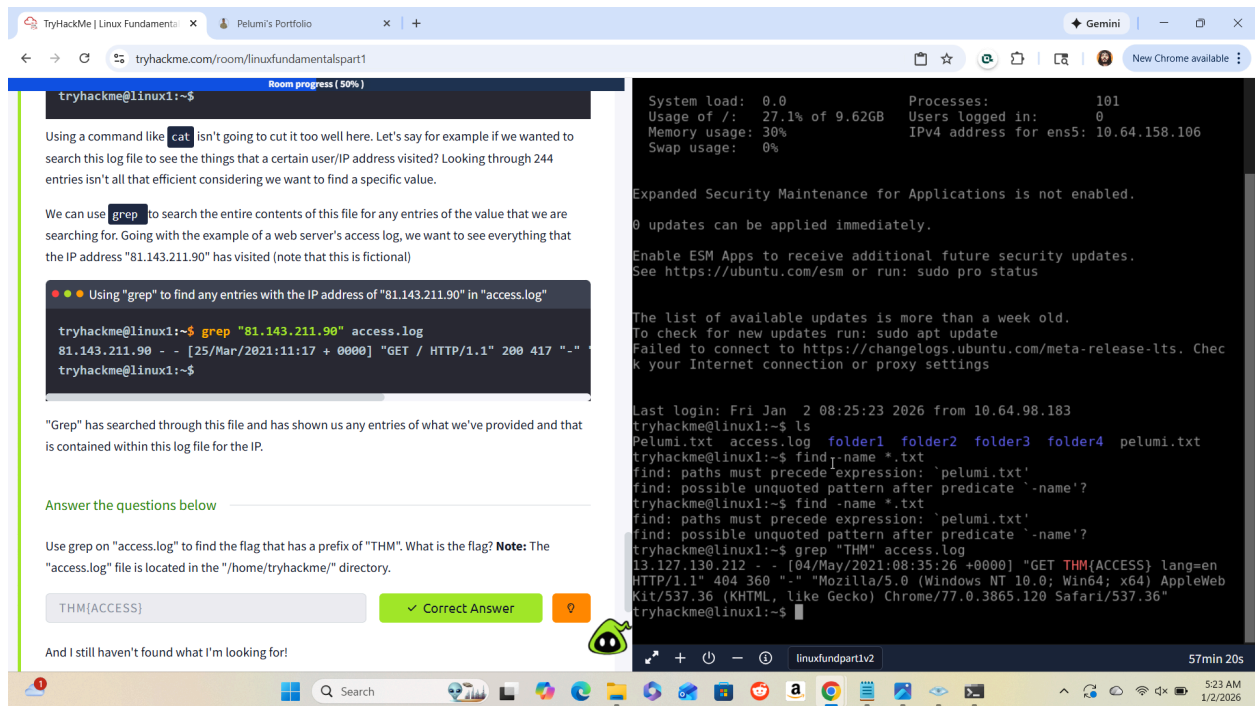
### Command used:

- grep "THM" access.log

Using grep, I searched a web server access log to find entries containing a specific pattern. This allowed me to successfully locate the embedded flag:

```
THM{ACCESS}
```

This step highlighted the importance of log analysis in cybersecurity monitoring and incident response.



## Step 7: Output Redirection and Command Operators

I learned how Linux handles command output and how that output can be redirected or controlled using shell operators.

### Operators learned:

- > Overwrites file contents
- >> Appends to file contents
- ; connect multiple commands
- && Runs the next command only if the previous command succeeds
- & Runs a command in the background

### Commands used:

- echo password123 > passwords
- echo tryhackme >> passwords

Using the > operator, I replaced the contents of a file.

Using the >> operator, I appended new content without overwriting existing data.

The screenshot shows a web browser with two tabs: 'TryHackMe | Linux Fundamentals' and 'Pelumi's Portfolio'. The active tab is 'tryhackme.com/room/linuxfundamentalspart1'. The page content includes a 'Room progress (61%)' bar and a text explanation of the output redirection operator '>'. It shows how to redirect the output of the 'echo' command to a file named 'welcome'. Two terminal snippets are provided: one using 'echo hey > welcome' and another using 'cat welcome' to verify the file's contents. A note states that if the file already exists, its contents will be overwritten. Below this is a section titled 'Operator ">>"' which explains that this operator appends content to a file instead of overwriting it. To the right of the browser is a terminal window titled 'Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1064-aws x86\_64)'. It displays system information, update status, and the results of the commands 'echo hey > welcome' and 'cat welcome', which outputs 'hey'. The terminal window has a timer at the bottom showing '1h 17min 46s'.

The screenshot shows the same web browser with the 'tryhackme.com/room/linuxfundamentalspart1' page. The 'Room progress' bar is now at '83%'. The page displays a quiz section titled 'Answer the questions below'. It contains three questions about Linux operators: 1) 'If we wanted to run a command in the background, what operator would we want to use?' with the answer '&'. 2) 'If I wanted to replace the contents of a file named "passwords" with the word "password123", what would my command be?' with the answer 'echo password123 > passwords'. 3) 'Now if I wanted to add "tryhackme" to this file named "passwords" but also keep "passwords123", what would my command be?' with the answer 'echo tryhackme >> passwords'. Each question has a 'Correct Answer' button. Below the quiz are sections for 'Task 8: Conclusions & Summaries' and 'Task 9: Linux Fundamentals Part 2'. At the bottom, there is a question 'How likely are you to recommend this room to others?'. To the right, the terminal window is the same as in the first screenshot, showing the same commands and output, with a timer at the bottom showing '1h 4min 34s'.

## Step 8: Command Chaining with the Semicolon (;)

I learned how to connect multiple commands using the semicolon operator.



**Example:**

commandA ; commandB

Command A runs first, followed immediately by command B, regardless of whether command A succeeds or fails. This is useful when running independent commands sequentially.

## Step 9: Conditional Command Execution with &&

I learned how to chain commands conditionally using the && operator.

**Example:**

Command A && command B {sudo apt update && sudo apt upgrade -y}

This allows for running multiple commands on a go. With this operator, command B only runs if command A completes successfully. This ensures controlled execution and is commonly used in installations, updates, and automation scripts.

## Step 10: Background Execution with &

I learned how to run commands in the background while continuing to use the terminal.

**Example:**

command &

This allows a command to run asynchronously, meaning I could continue executing other commands while a process was running in the background. This demonstrated Linux's multitasking capabilities.

## Step 11: Practical Reinforcement

I applied all learned commands directly on the live Linux machine, reinforcing:

- User awareness

- Directory navigation
- File creation and modification
- Log searching
- Output redirection
- Command chaining and background execution

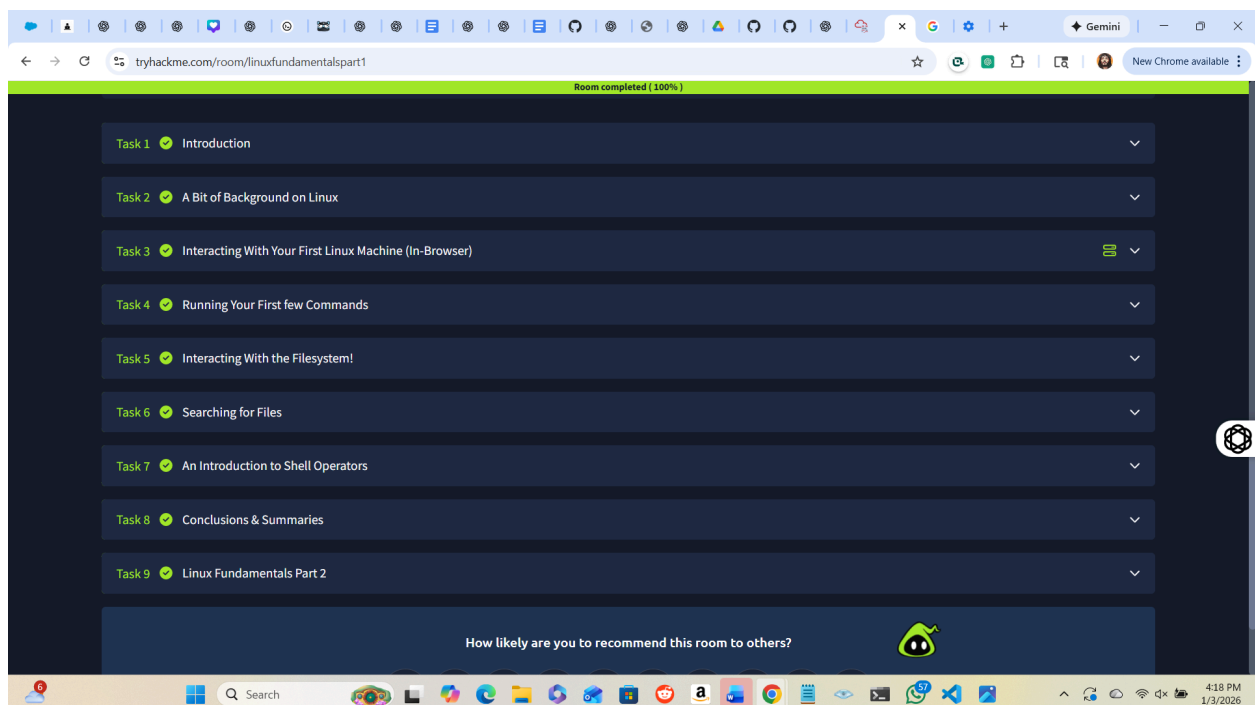
Each task was validated through TryHackMe's built-in checks.

## Completion

I successfully completed all tasks in the TryHackMe Linux Fundamentals Part 1 room. The completion screen confirmed that all exercises were validated and points were awarded.

## Key Takeaways

- Linux is precise and unforgiving, but extremely powerful when used intentionally
- The command line becomes approachable once its structure is understood
- Logs contain valuable information, and tools like grep help uncover it
- Small command-line skills form the foundation for advanced cybersecurity operations




TryHackMe | Linux Fundamentals

Pelumi's Portfolio

Gemini

tryhackme.com/room/linuxfundamentalspart1

New Chrome available



### Congratulations on completing Linux Fundamentals Part 1!!! 🎉


Points earned  
88

Completed tasks  
9

Room type  
Walkthrough

Difficulty  
info


Streak  
1

 87,514 users are actively learning this week

Leave Feedback

Continue

Search



6:18 AM  
1/2/2026