

**Cloud Computing Lab**

**Submitted To:**

**Sir Shoaib & Sir Waqas**

**Submitted By:**

**Hamail Fatima**

**Roll No:**

**2023-BSE-023**

**Section:**

**5(A)**

**Lab #04**

**Task 1 – Verify VM resources in VMware**

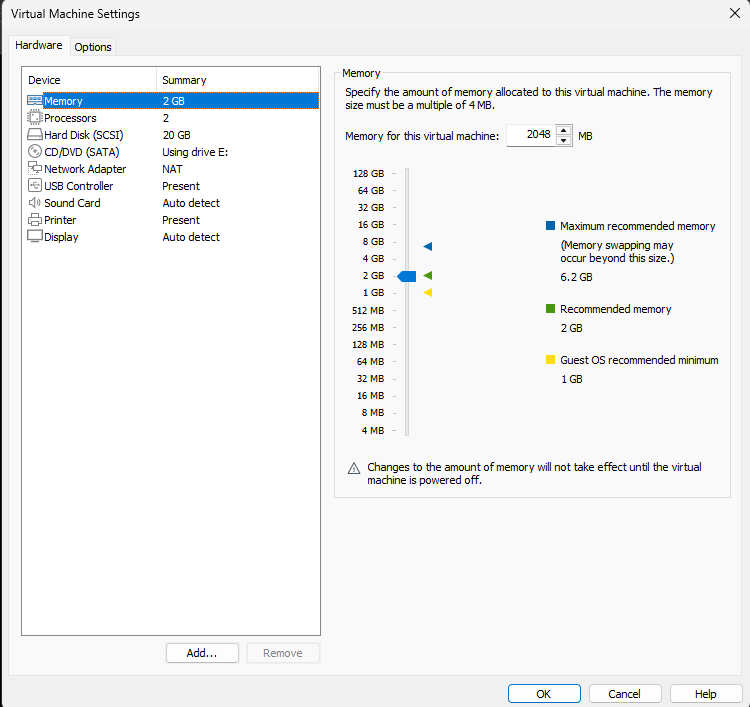
Confirm the VM resources that were allocated in Lab 1.

Steps

1. Open VMware Workstation and locate the Ubuntu Server VM you used in Lab 1.
2. Inspect VM settings and note the following (no commands required for GUI): VM name, RAM, CPU, disk, and network adapter type.
3. Take a screenshot of the VM settings window showing RAM, CPU, disk and networking. Save screenshot as: vm\_settings.png

📸 Screenshot Required:

* vm\_settings.png

****

**Task 2 – Start VM and log in (use your preferred host terminal method only)**

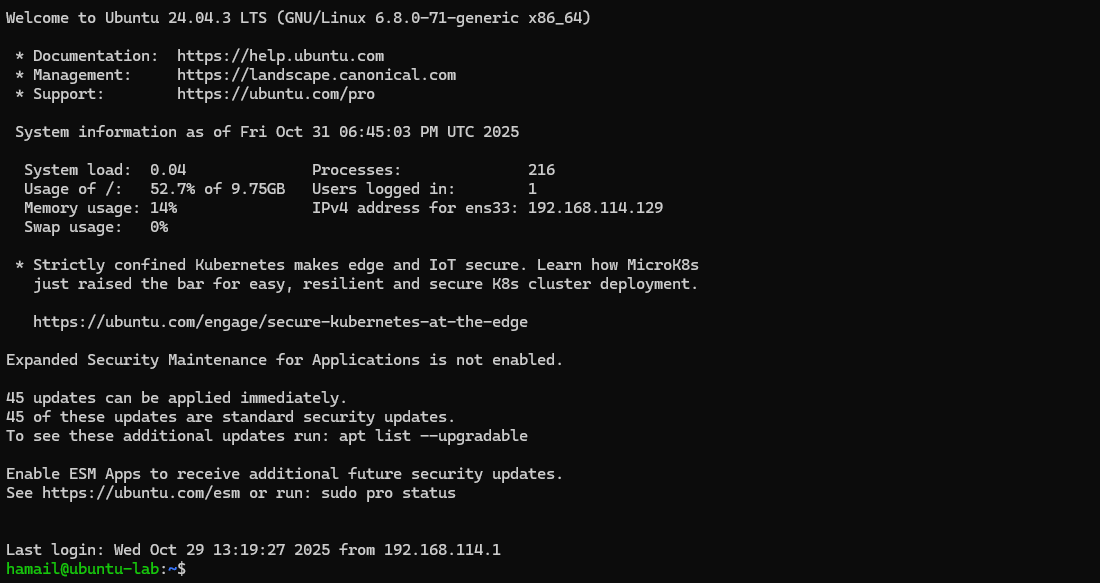
Use a single preferred host-terminal method to connect to the VM. Do not switch between methods during the task.

Steps

1. Start (or resume) the VM in VMware Workstation on your host.
2. From your host, open your preferred terminal (for example: Windows Command Prompt, PowerShell, macOS Terminal, or Linux Terminal) and connect to the VM using SSH. Example:

ssh student@<vm-ip-address>

* After connecting, save a screenshot of your host terminal showing the SSH login prompt/results as: vm\_login.png



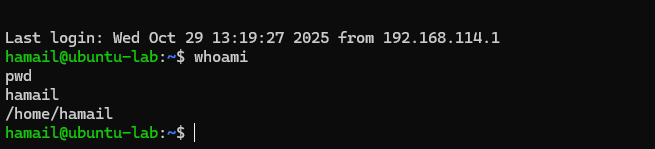
1. After logging in, run both commands and capture them together in a single screenshot:

whoami

pwd

* Save a single screenshot that clearly shows both outputs as: whoami\_pwd.png

Screenshot Required:



* vm\_login.png (optional if whoami\_pwd.png provided)
* whoami\_pwd.png (required; must show both whoami and pwd outputs)

**Task 3 – Filesystem exploration — root tree and dotfiles**

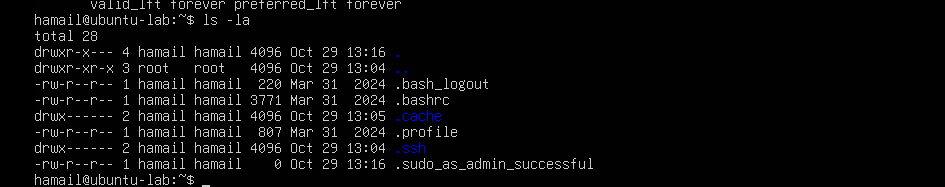
Explore Linux filesystem layout and hidden files. Capture outputs using screenshots only (do not create text files). For the short explanation required in this task, write the paragraph in an editor and capture it as a screenshot (answers\_md.png) — do not supply the .md file.

**Steps (run inside VM terminal)**

1. List root directory contents:

ls -la /

* Save screenshot as: ls\_root.png



1. Inspect these directories (run each command and screenshot the output):

ls -la /bin

* Save screenshot as ls\_bin.png.



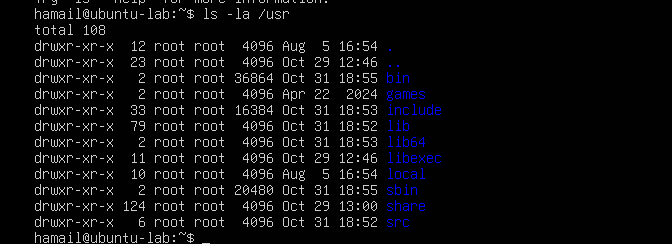
ls -la /sbin

* Save screenshot as ls\_sbin.png



ls -la /usr

* Save screenshot as ls\_usr.png.



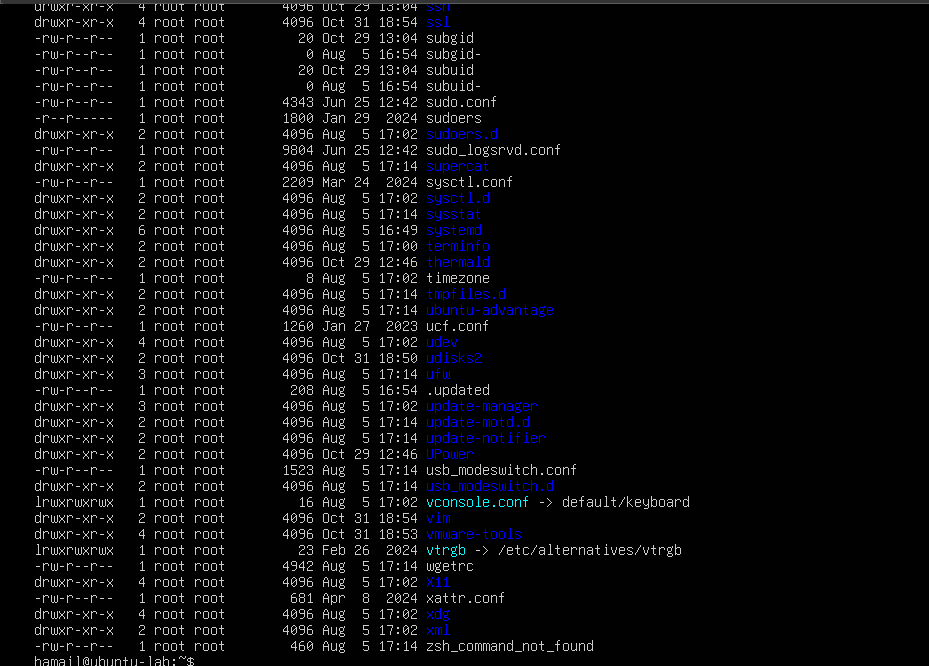
ls -la /opt

* Save screenshot as ls\_opt.png.



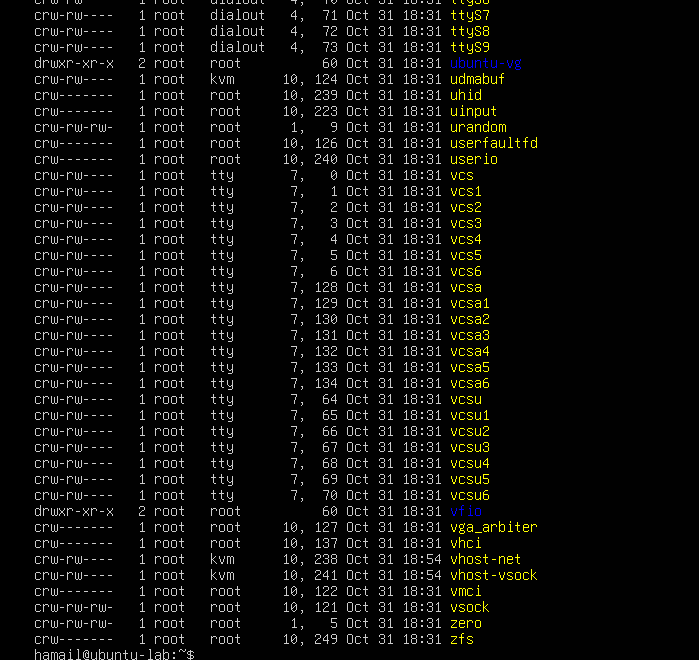
ls -la /etc

* Save screenshot as ls\_etc.png.



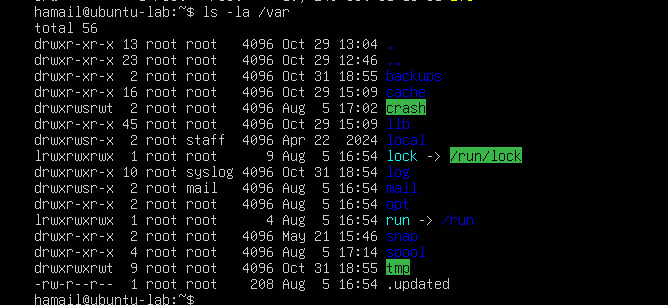
ls -la /dev

* Save screenshot as ls\_dev.png.



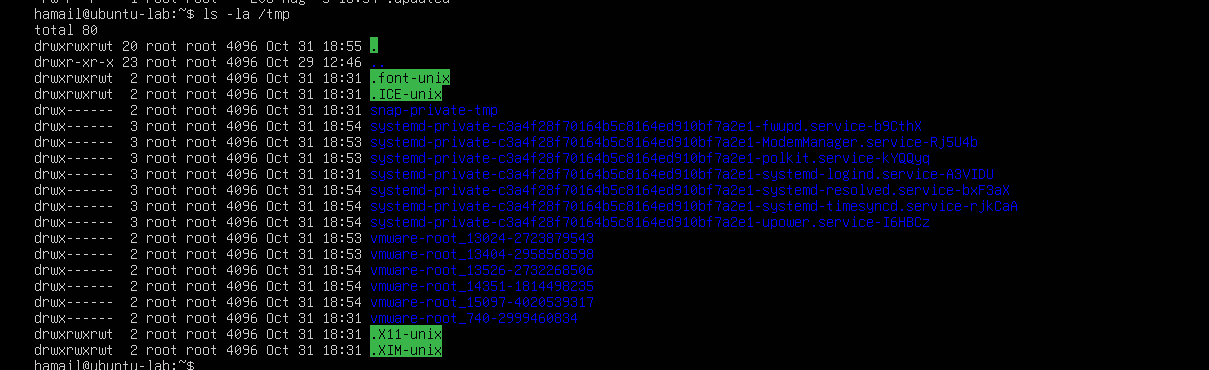
ls -la /var

* Save screenshot as ls\_var.png.



ls -la /tmp

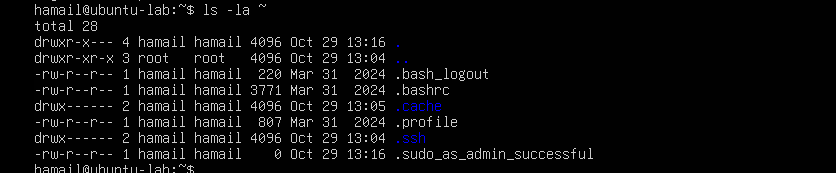
* Save screenshot as ls\_tmp.png.



1. List your home directory and show hidden (dot) files:

ls -la ~

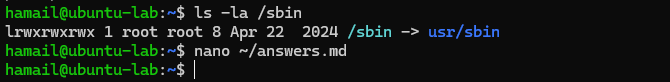
* Save screenshot as: home\_ls.png



1. Write a short paragraph (3–5 sentences) that explains the difference between /bin, /usr/bin and /usr/local/bin. Open your editor:

nano ~/answers.md

* Type the paragraph in the editor, save and exit.
* After saving, open the editor display (or show the file) and capture a screenshot of the paragraph. Save that screenshot as: answers\_md.png



**Task 4 – Essential CLI tasks — navigation and file operations**

Practice basic commands used daily on Linux systems. Use screenshots only for evidence.

**Steps (inside VM terminal)**

1. Create a workspace and navigate:

mkdir -p ~/lab4/workspace/python\_project

* Save screenshot as mkdir\_workspace.png.

cd ~/lab4/workspace/python\_project



* Save screenshot as cd\_workspace.png.



pwd

* Save screenshot as pwd\_workspace.png.



1. Create files using an editor (open each editor session and save a screenshot showing content):

nano README.md

* Inside nano add: Lab 4 README and save.
* Save screenshot of the editor after saving as nano\_readme.png.

nano main.py



* Inside nano add: print("hello lab4") and save.
* Save screenshot as nano\_main.png.



nano .env

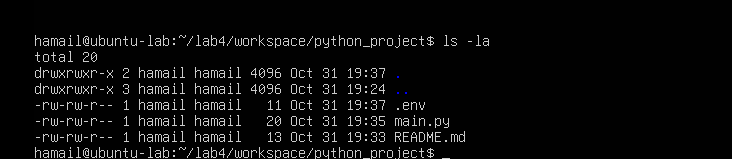
* Inside nano add: ENV=lab4 and save.
* Save screenshot as nano\_env.png.



1. List files and capture:

ls -la

* Save screenshot as workspace\_ls.png.



1. Copy, move and remove:

cp README.md README.copy.md

* After running, save screenshot as cp\_readme.png.



mv README.copy.md README.dev.md

* After running, save screenshot as mv\_readme.png.

rm README.dev.md



* After running, save screenshot as rm\_readme.png.



mkdir -p ~/lab4/workspace/java\_app

* Save screenshot as mkdir\_java\_app.png.



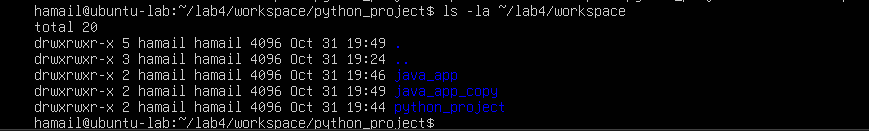
cp -r ~/lab4/workspace/python\_project ~/lab4/workspace/java\_app\_copy

* After running, save screenshot as cp\_recursive.png.



ls -la ~/lab4/workspace

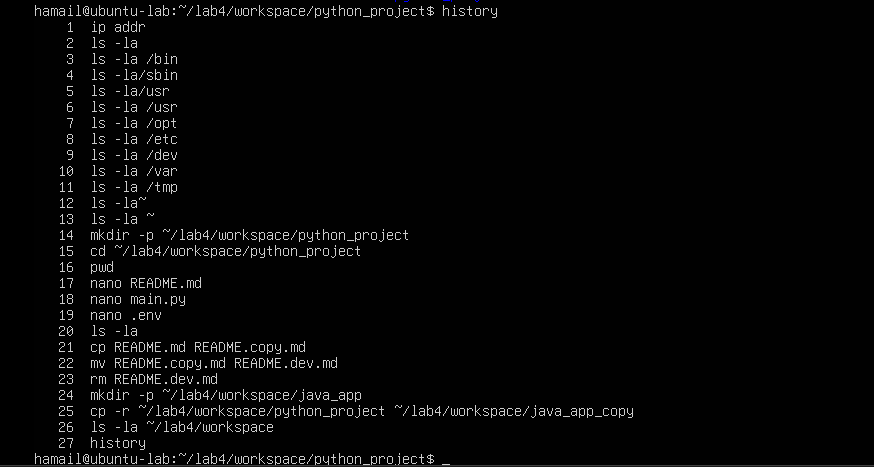
* Save screenshot as copy\_verify.png.



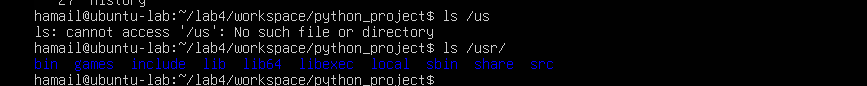
1. Use command history and tab completion:

history

* Save screenshot as history.png.



* Demonstrate tab completion (type partial name and press Tab) and capture that action as tab\_completion.png.



**Task 5 – System info, resources & processes**

Collect system information and observe processes. Use screenshots only.

**Steps (inside VM terminal)**

1. Kernel and OS:

uname -a

* Save screenshot as uname.png.



1. CPU (ensure model name visible):

cat /proc/cpuinfo

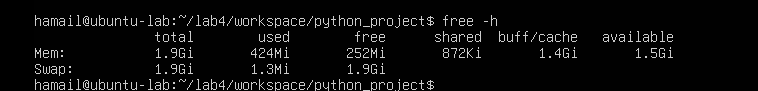
* Save screenshot as cpuinfo.png.



1. Memory:

free -h

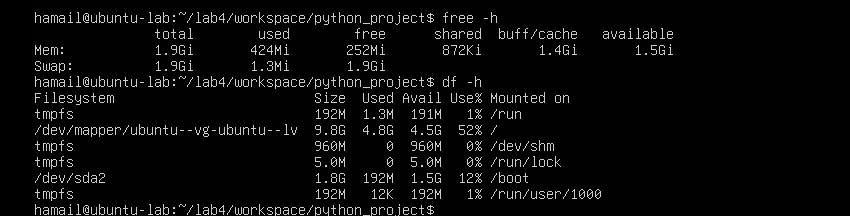
* Save screenshot as meminfo.png.



4.Disk:

df -h

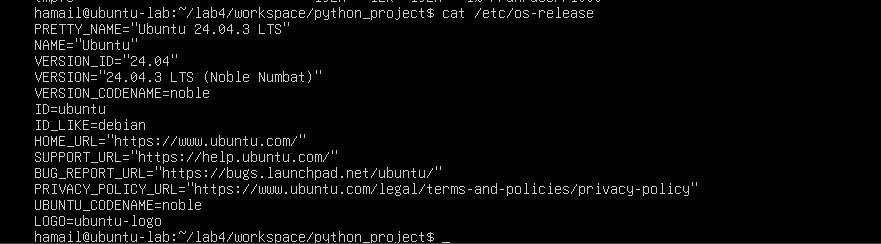
* Save screenshot as diskinfo.png.



1. View OS release information:

cat /etc/os-release

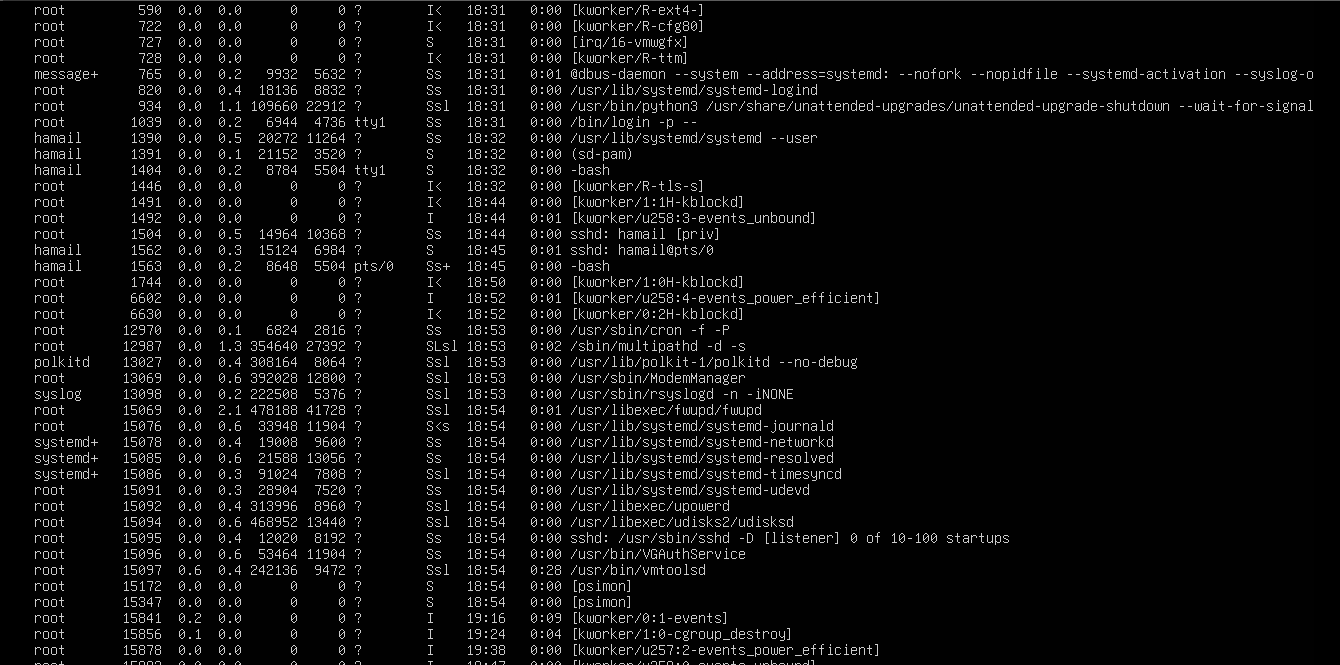
* Save screenshot as os-release.png.



1. Processes (show top lines of ps output):

ps aux

* Save screenshot as processes.png.



**Task 6 – Users and account verification (no sudo group change)**

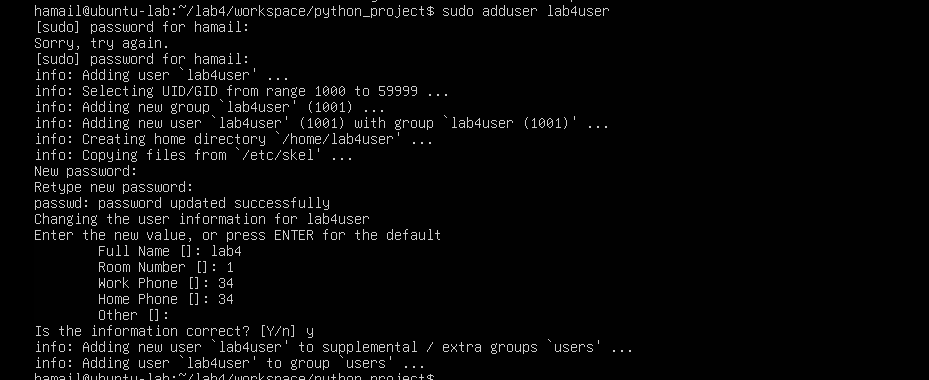
Create a non‑root user and verify the account exists. This task does NOT add the created user to the sudo group.

**Steps (inside VM terminal)**

1. Create a new user named lab4user:

sudo adduser lab4user

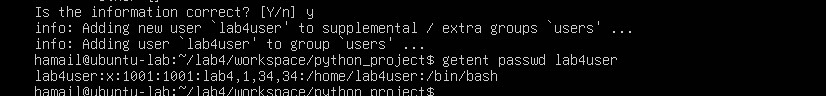
* During prompts, capture the terminal and save screenshot as adduser\_lab4user.png.



1. Verify the user entry:

getent passwd lab4user

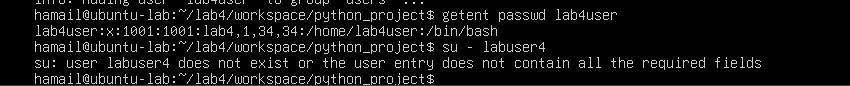
* Save screenshot as lab4user\_passwd.png.



1. Switch to the new user to verify login:

su - lab4user

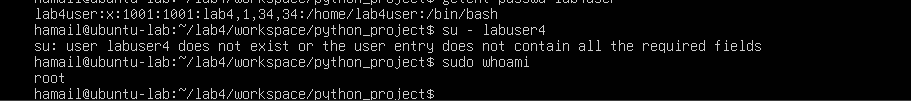
* Save screenshot as su\_lab4user.png.



1. From the new user you may attempt a sudo command to show that sudo is not available for this account (expected failure), e.g.:

sudo whoami

* Save screenshot as sudo\_whoami.png.



1. Return to the original user:

exit

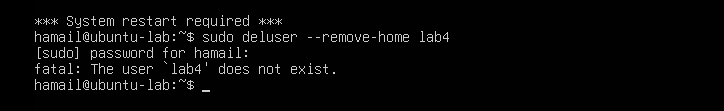
* Save screenshot as exit\_back.png.



1. (Optional) Remove the test user when finished:

sudo deluser --remove-home lab4user

* If run, save screenshot as deluser.png.



**Bonus Task 7 – Create a small demo script using an editor and run it**

This task is optional — complete it for extra practice or extra credit. It is not required for passing the core lab tasks.

**Steps (inside VM)**

1. Open an editor to create the script:

nano ~/lab4/workspace/run-demo.sh

* Type the following lines into the editor (manually or paste), save and exit:

#!/bin/bash

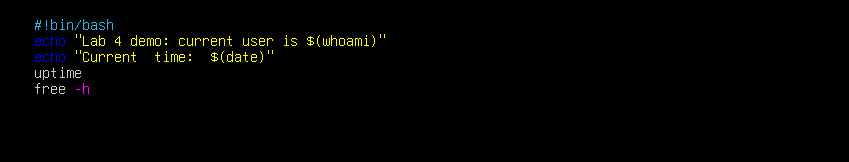
echo "Lab 4 demo: current user is $(whoami)"

echo "Current time: $(date)"

uptime

free -h

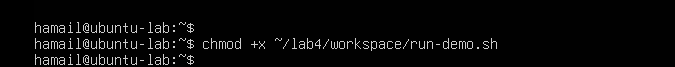
* Save screenshot of the editor after saving the file as nano\_run\_demo.png.



1. Make the script executable:

chmod +x ~/lab4/workspace/run-demo.sh

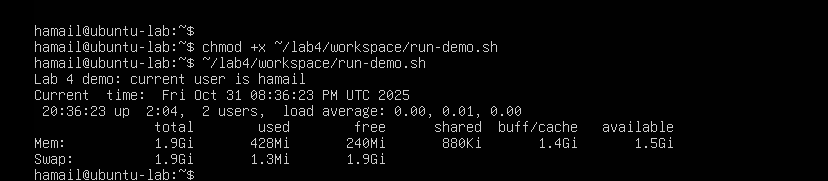
* Save screenshot as chmod\_run\_demo.png.



1. Run the script as your regular user:

~/lab4/workspace/run-demo.sh

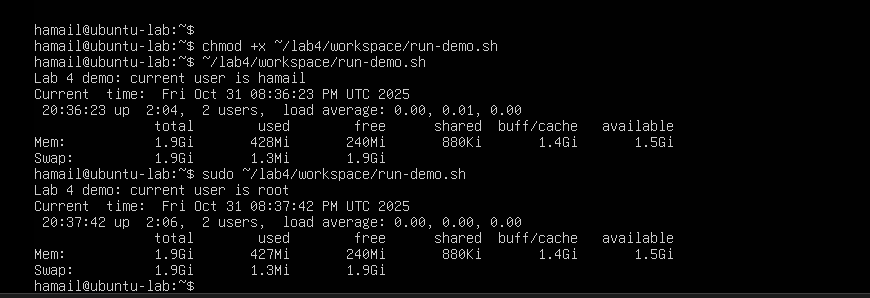
* Save screenshot of the script output as run\_demo\_output.png.



1. Optionally run it with sudo:

sudo ~/lab4/workspace/run-demo.sh

* Save screenshot as run\_demo\_output\_sudo.png.



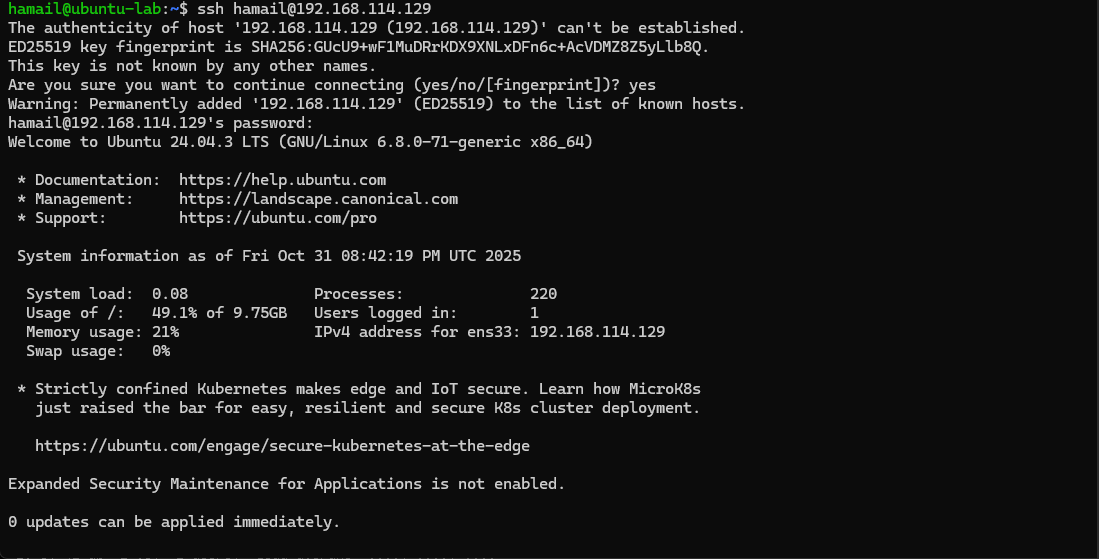
**Exam Evaluation Questions**

**1. Remote Access Verification (Cyber Login Check)**

**Scenario:**  
You are part of a SOC (Security Operations Center) investigating unauthorized access to a Linux server hosted on VMware. Prove you can securely connect and verify your identity.

**Steps:**

1. Connect to the Ubuntu VM remotely from your host terminal.
   * Screenshot as Q1\_remote\_connection.png



2.Verify your current user and home directory path.

* + Screenshot as Q1\_user\_verification.png



1. Confirm you are connected to the correct host machine.
   * Screenshot as Q1\_host\_confirmation.png

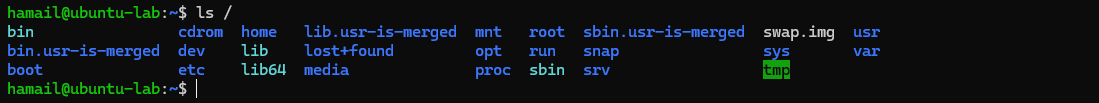


**2. Filesystem Inspection for Forensic Evidence**

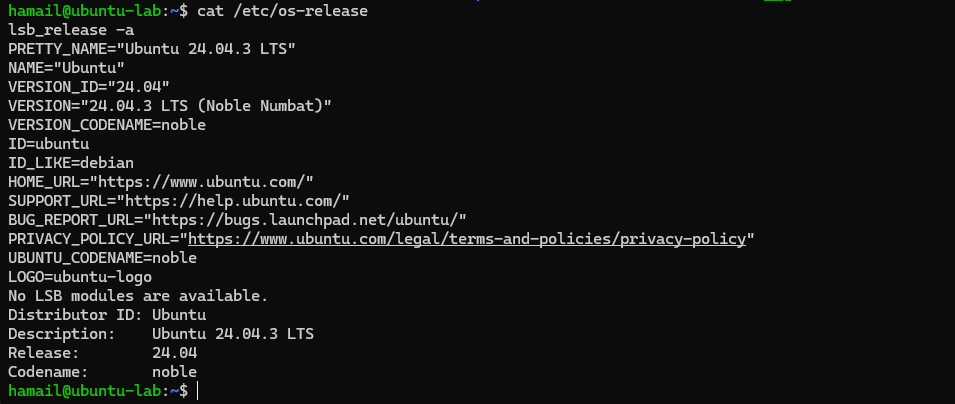
**Scenario:**  
The incident response team suspects malicious files in system directories. You must explore the filesystem to locate and document the system’s structure.

**Steps:**

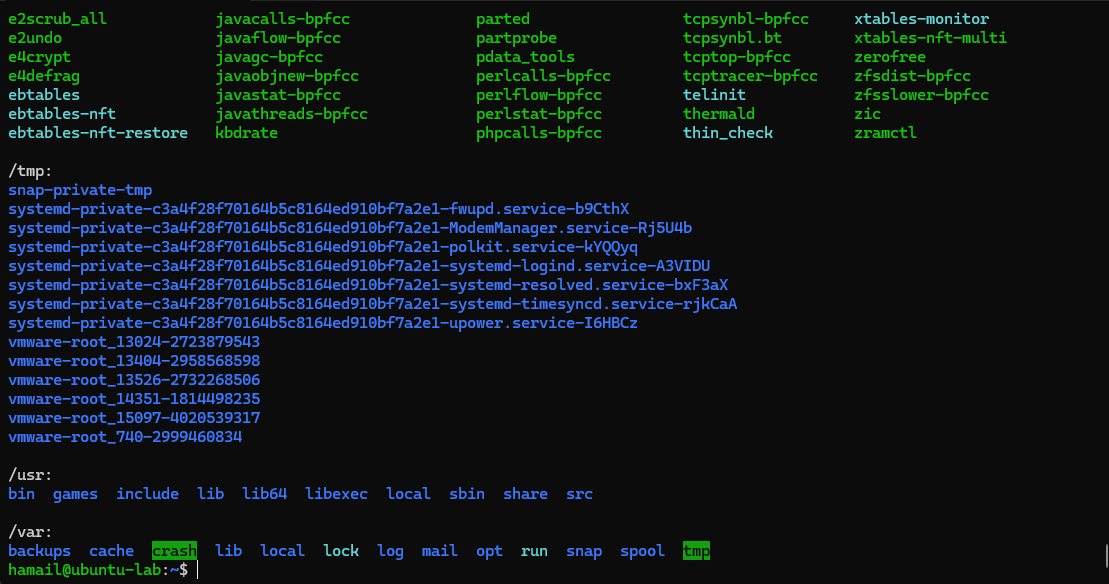
1. Display the contents of the root directory.
   * Screenshot as Q2\_root\_listing.png



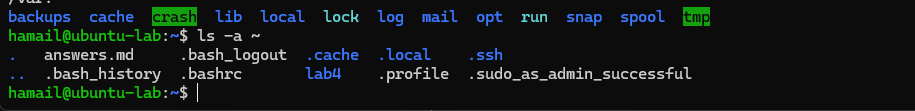
1. Display the OS version and release information.
   * Screenshot as Q2\_os\_version.png



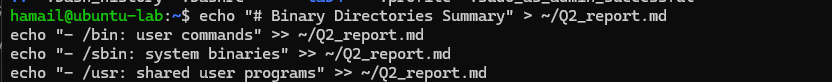
1. Explore and record directory listings for /bin, /sbin, /usr, /opt, /etc, /dev, /var, and /tmp.
   * Screenshot as Q2\_directory\_evidence.png



1. Display all hidden files in your home directory.
   * Screenshot as Q2\_hidden\_files.png



1. Create a markdown file summarizing your findings on key binary directories.
   * Screenshot as Q2\_report\_file.png



**3. Evidence Handling & File Operations**

**Scenario:**  
You are creating a sandbox environment to safely analyze and handle suspicious files collected from a compromised system.

**Steps:**

1. Create a structured folder hierarchy under your home directory for analysis.
   * Screenshot as Q3\_workspace\_created.png



1. Create three text files, including one hidden file, in your workspace.
   * Screenshot as Q3\_files\_created.png



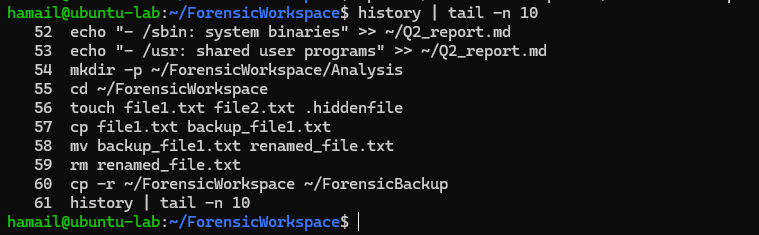
1. Create a backup copy of one file, rename it, and then delete it after verification.
   * Screenshot as Q3\_backup\_handling.png



1. Copy the entire workspace as an evidence backup folder.
   * Screenshot as Q3\_workspace\_backup.png



1. Display your command history to document all actions performed.
   * Screenshot as Q3\_command\_history.png



1. Demonstrate Linux auto-completion by typing a partial command or filename.
   * Screenshot as Q3\_autocomplete.png



**4. System Profiling and Process Monitoring**

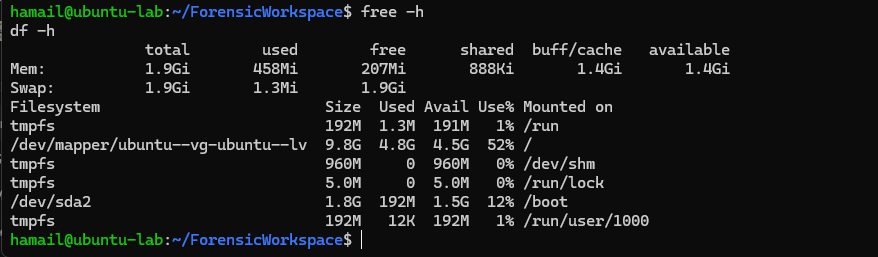
**Scenario:**  
You are investigating a potential malware infection that is consuming excessive resources on the Linux VM.

**Steps:**

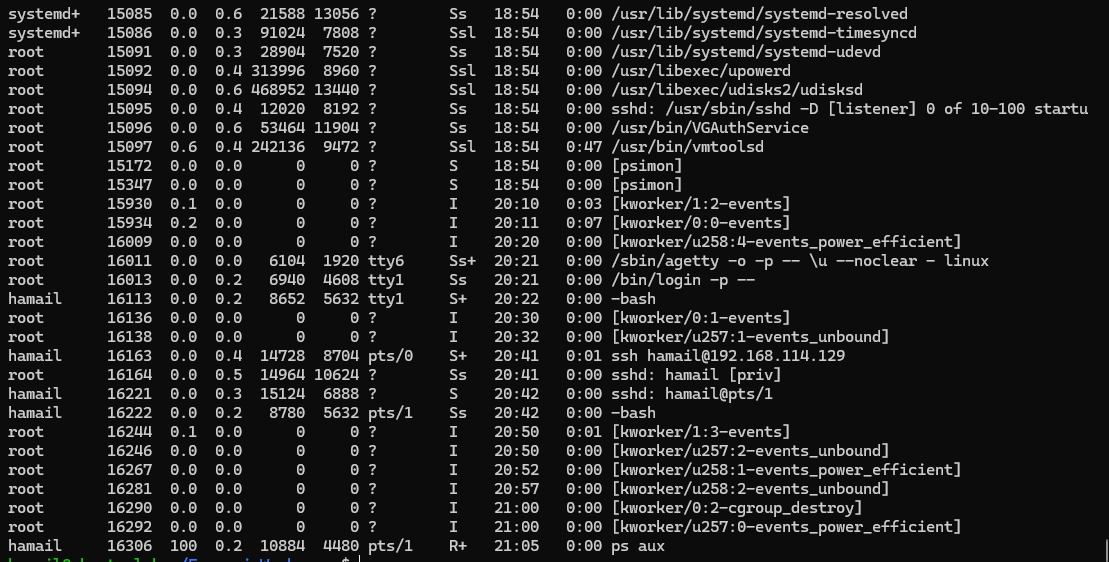
1. Display the system’s OS and kernel version for the investigation report.
   * Screenshot as Q4\_system\_info.png



1. Display CPU, memory, and disk usage information.
   * Screenshot as Q4\_resource\_info.png



1. Display all active running processes to identify suspicious activity.
   * Screenshot as Q4\_process\_list.png

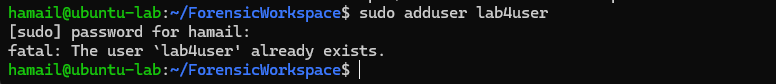


**5. User Account Audit & Privilege Escalation Simulation**

**Scenario:**  
You are performing a **user activity audit** on a compromised Linux server.  
The SOC suspects a newly created account (lab4user) may have been used for unauthorized access.  
Your task is to simulate the account creation, perform privilege tests, and analyze authentication logs for forensic evidence.

**Steps:**

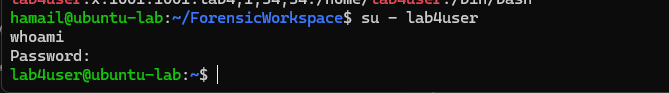
1. Create a new test user named lab4user.
   * Screenshot as Q5\_user\_created.png



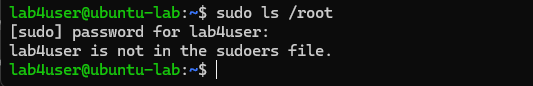
1. Verify that the new user record exists in the system’s user database.
   * Screenshot as Q5\_user\_verified.png



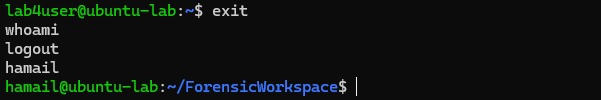
1. Log in as lab4user and confirm successful login.
   * Screenshot as Q5\_user\_login.png



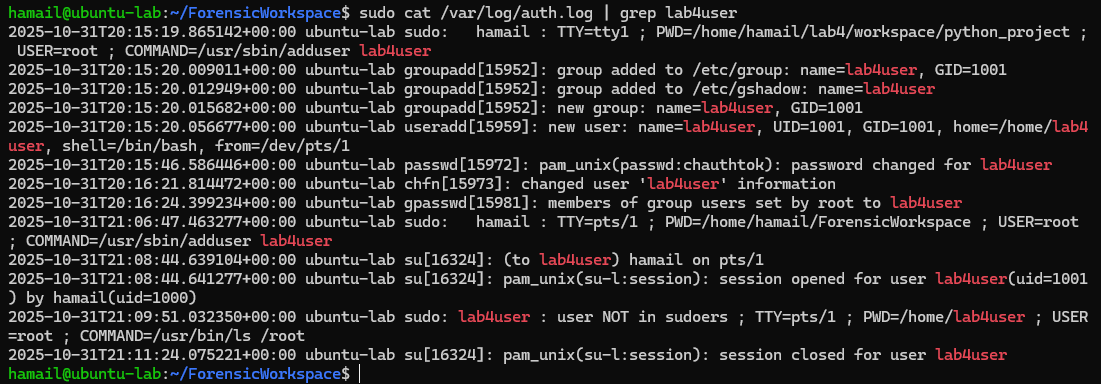
1. Attempt to run an administrative command as lab4user (expect permission denied).
   * Screenshot as Q5\_permission\_denied.png



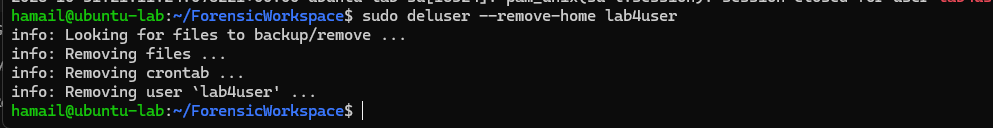
1. Switch back to your main analyst account.
   * Screenshot as Q5\_switch\_back.png



1. Inspect the system authentication logs located at /var/log/auth.log to determine whether the lab4user account attempted any logins (successful or failed).
   * Screenshot as Q5\_authlog\_analysis.png



1. (Optional) Remove the lab4user account after the audit and verify deletion.
   * Screenshot as Q5\_user\_removed.png



**Reminder:**  
Take a screenshot after every step and name it as shown above. Include all screenshots in your submission for full credit.

**Summary**

In this lab you:

* Verified VM configuration in VMware Workstation.
* Logged into the existing Ubuntu Server VM using the Windows host terminal and explored the filesystem.
* Practiced core Linux CLI commands for navigation and file manipulation (evidence via screenshots).
* Collected system and process information and created a non‑root user account without granting sudo.
* (Optional) Wrote and executed a simple demo script using an editor.

**Submission**

* Upload a **Word file and PDF or .md file** containing:
  + Step outputs or terminal screenshots for each task
  + Your answers to the hands-on practical exam questions
  + workspace/ — folder containing files you created in the workspace:
    - README.md, main.py, .env (required)
    - run-demo.sh (optional / bonus)
  + screenshots/ — folder containing all required screenshots listed above (preferred), including answers\_md.png