# ABDULLAH SOHAIL 70147165 SEC ( I )

# Assignment: Mastering Linux for Cloud Computing (Ubuntu Focus)

Course: Cloud Computing  
Total Marks: 100  
Submission Format: PDF or DOCX with Screenshots and Source Code

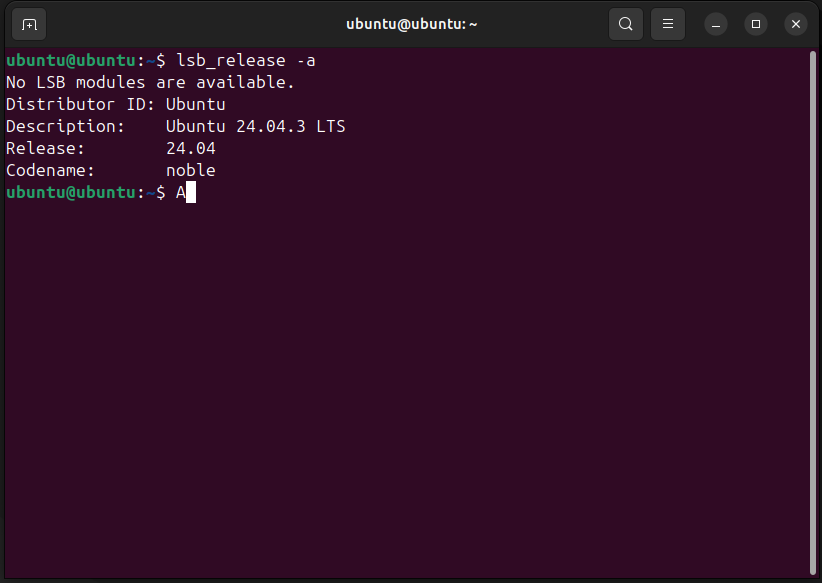
## Objective

To build practical expertise in using Linux (Ubuntu) as a foundation for Cloud Computing environments — from beginner-level system navigation to advanced administration, automation, and cloud integration.

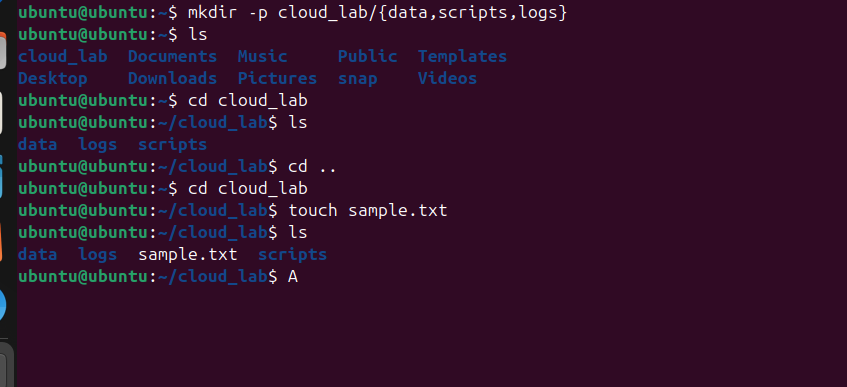
## Part A: Linux Fundamentals (20 Marks)

Goal: Get comfortable with Linux basics and system navigation.

1. Install Ubuntu (Desktop or Server) on your system or virtual machine.  
    - Mention version used (e.g., Ubuntu 22.04 LTS).  
    - Take a screenshot after successful login.

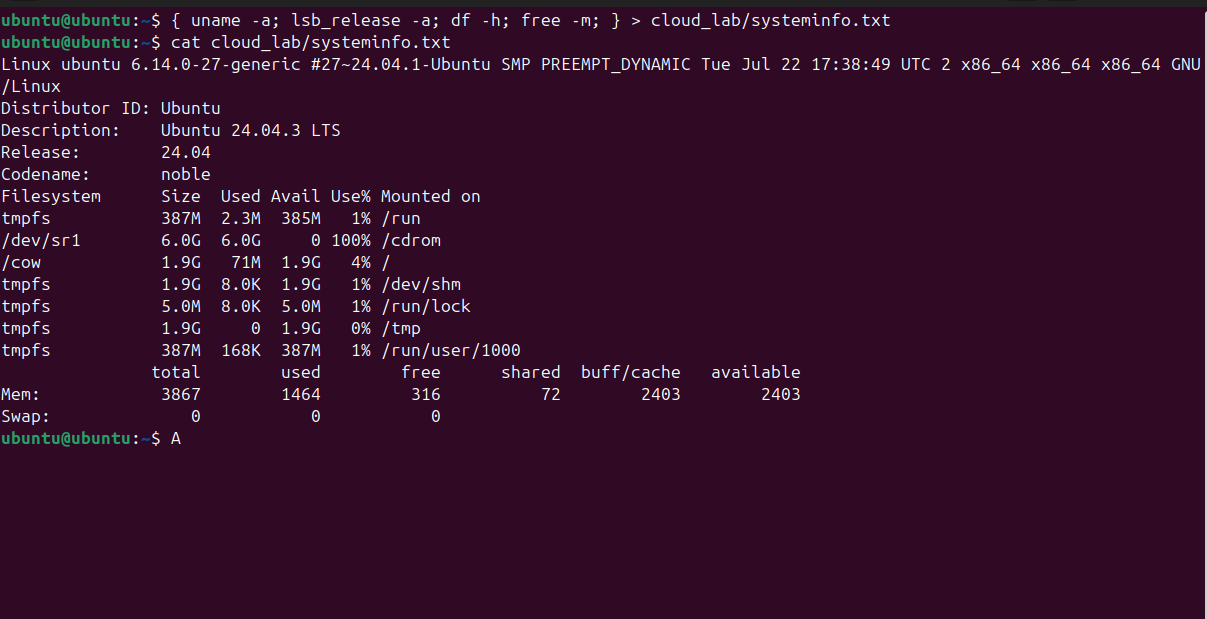


2. Explore the Linux File System using commands: pwd, ls, cd, mkdir, rmdir, cp, mv, rm, cat, less.  
 - Create a directory named 'cloud\_lab' and inside it make three subfolders: data, scripts, and logs.

  
3. Display system information using: uname -a, hostnamectl, lsb\_release -a, df -h, free -m.

A screenshot of a computer

AI-generated content may be incorrect.  
4. Create a new user named 'cloud\_user' and grant them sudo privileges.



5. Practice file permissions with chmod, chown, and chgrp.

A screenshot of a computer program

AI-generated content may be incorrect.

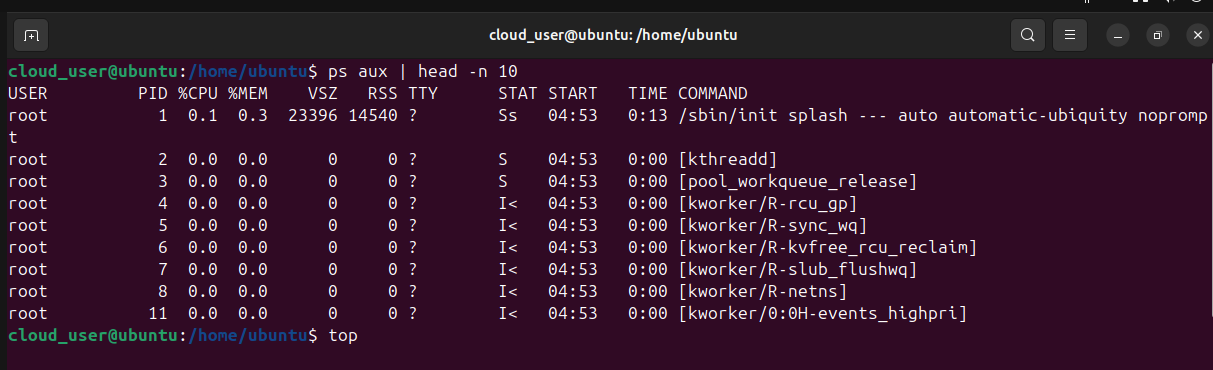
A computer screen shot of a program

AI-generated content may be incorrect.

## Part B: Intermediate Linux Administration (25 Marks)

Goal: Learn to manage system processes, packages, and network.

1. Manage processes and services using ps, top, htop, and systemctl.



A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

2. Explore package management with apt (install/remove curl, wget, vim, git).

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

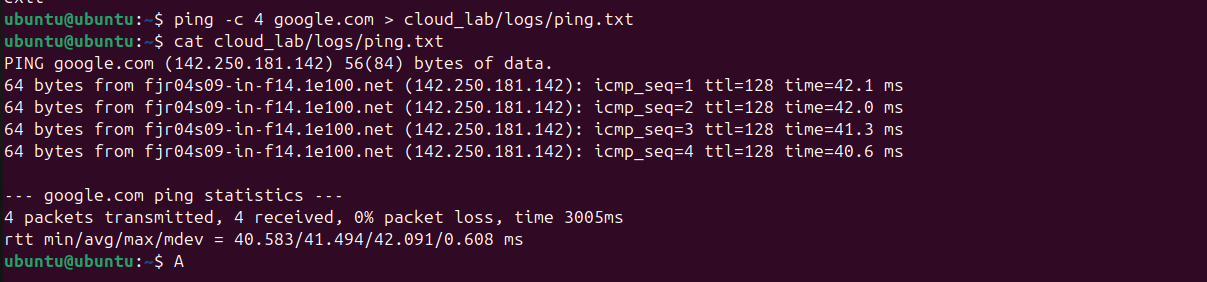
A screenshot of a computer program

AI-generated content may be incorrect.

3. Configure network settings: ip addr, ping google.com and save results.

A screenshot of a computer program

AI-generated content may be incorrect.

  
4. Explore disk management: lsblk, fdisk -l, mount/unmount drives.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

5. Create cron jobs to log uptime every hour into

/home/cloud\_user/logs/uptime.log.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

## Part C: Advanced Linux Operations (30 Marks)

Goal: Develop advanced control over Linux systems and automation.

1. Write a Bash Script that creates a directory named with the current date, copies all .log files into it, and compresses it into a .tar.gz file.

A screenshot of a computer screen

AI-generated content may be incorrect.

2. Configure firewall using ufw (allow SSH, HTTP, HTTPS, deny others).

A screenshot of a computer program

AI-generated content may be incorrect.

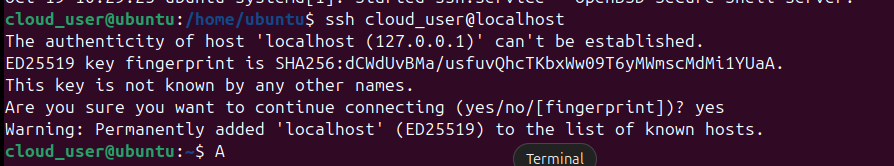
A screenshot of a computer program

AI-generated content may be incorrect.

3. Implement SSH Key Authentication using ssh-keygen.

A screenshot of a computer

AI-generated content may be incorrect.

4. Monitor system performance using vmstat, iostat, sar, and dstat.  
5. Create a system backup using rsync from /home/cloud\_user/data to /mnt/backup.

## Part D: Expert-Level Tasks – Cloud Integration and Automation (25 Marks)

Goal: Connect Linux skills with real Cloud Computing workflows.

1. Install and configure Docker on Ubuntu and run a sample container (e.g., nginx).  
2. Create a Bash automation script to deploy a Docker container automatically.  
3. Install and configure AWS CLI or Azure CLI, and list resources.  
4. Create an Ansible playbook to install and enable NGINX.  
5. Bonus (+5): Create a local Kubernetes cluster with Minikube and deploy NGINX pod.

## Deliverables

- A report (DOCX or PDF) including steps, commands, screenshots, and explanations.  
- A ZIP folder with all bash scripts and backup/configuration files.

## Evaluation Criteria

|  |  |
| --- | --- |
| Criteria | Marks |
| Basic Tasks Completed Correctly | 20 |
| Intermediate Admin Tasks | 25 |
| Advanced Scripting & Configuration | 30 |
| Cloud Integration Tasks | 25 |
| Total | 100 |