Author: Riyan Uddin

Linux Challenge for DevOps Students

Objective:

The objective of this challenge is to assess the students' ability to manage a Linux environment effectively, write shell scripts, and perform system administration tasks. They should be able to demonstrate their understanding of Linux file system hierarchy, user management, process management, networking, and automation.

Challenge Tasks:

Beginner Level

Task 1: Basic Linux Commands

1. **Create and Navigate Directories:**
   - Create a directory structure as follows: `/home/devops/academy/project`.
   - Navigate into the `project` directory.
2. **File Management:**
   - Create an empty file named `README.md` inside the `project` directory.
   - Create another file named `notes.txt` and add the text "DevOps Linux Challenge" into it.
   - Copy `notes.txt` to `/home/devops/academy/backup/`.
3. **Permissions:**
   - Change the permissions of `README.md` to be readable and writable only by the owner.
   - Make `notes.txt` readable by everyone, but writable only by the owner.

Task 2: User and Group Management

1. **Create Users:**
   - Create a new user named `student1`.
   - Set a password for `student1`.
2. **Groups:**
   - Create a new group named `devops`.
   - Add `student1` to the `devops` group.
3. **User Permissions:**
   - Ensure that the `project` directory is accessible to members of the `devops` group.
4. **Disk Usage:**
   - Display disk usage in human-readable format for `/home` directory and save the output to a file named `disk_usage.txt`.

Intermediate Level

Task 3: Process Management

1. **List Processes:**

   - Display all running processes and redirect the output to a file named `process_list.txt`.
2. **Background Process:**
   - Start a simple background process that writes the current date and time to a file named `timestamp.txt` every minute using a `while` loop and `sleep`.
3. **Kill Process:**
   - Find the process ID (PID) of the background process started in the previous step and terminate it.
4. **CPU and Memory Usage:**
   - Display the current CPU and memory usage using `top` or `htop`, save a snapshot of this information to a file named `cpu_mem_usage.txt`.

Task 4: Networking

1. **Network Configuration:**
   - Display the current network configuration using `ifconfig` or `ip addr`.
2. **Ping Test:**
   - Ping `google.com` and save the output to a file named `ping_results.txt`.
3. **Open Ports:**
   - List all open ports on the system using `netstat` or `ss` command.
4. **Firewall Configuration:**
   - Check if the `ufw` (Uncomplicated Firewall) is installed and running. If not, install and enable it.
   - Allow incoming connections on port 80 (HTTP) and 443 (HTTPS).

Task 5: Shell Scripting

1. **Backup Script:**
   - Write a shell script named `backup.sh` that compresses the `project` directory into a tar.gz file and saves it in the `/home/devops/academy/backup` directory. The script should include error handling to ensure the backup only proceeds if the `project` directory exists.
2. **Automation Script:**
   - Create a script named `cleanup.sh` that deletes all files in the `/home/devops/academy/backup` directory that are older than 7 days. Schedule this script to run daily using `cron`.
3. **Log Rotation:**
   - Write a script named `rotate_logs.sh` to rotate the `timestamp.txt` file if it exceeds 1MB in size and save old logs with a timestamp in the filename.
   -

Advanced Level:

- **Advanced Shell Scripting:**

- Write a script named `resource_monitor.sh` that:
   - Monitors CPU, memory, and disk usage.
   - Logs usage data every 5 minutes to a file named `resource_usage.log`.
   - Sends an alert email if CPU usage exceeds 80% or available disk space falls below 20%.

- **User Management Automation:**

    - Write a script named `bulk_user_add.sh` that:
        - Reads a list of usernames from a file named `user_list.txt`.
        - Creates each user and sets a default password.
        - Adds each user to a specific group (e.g., `students`).
        - Ensures each user has a home directory created.