Author: Riyan Uddin

Linux Challenge for DevOps Students

Objective:

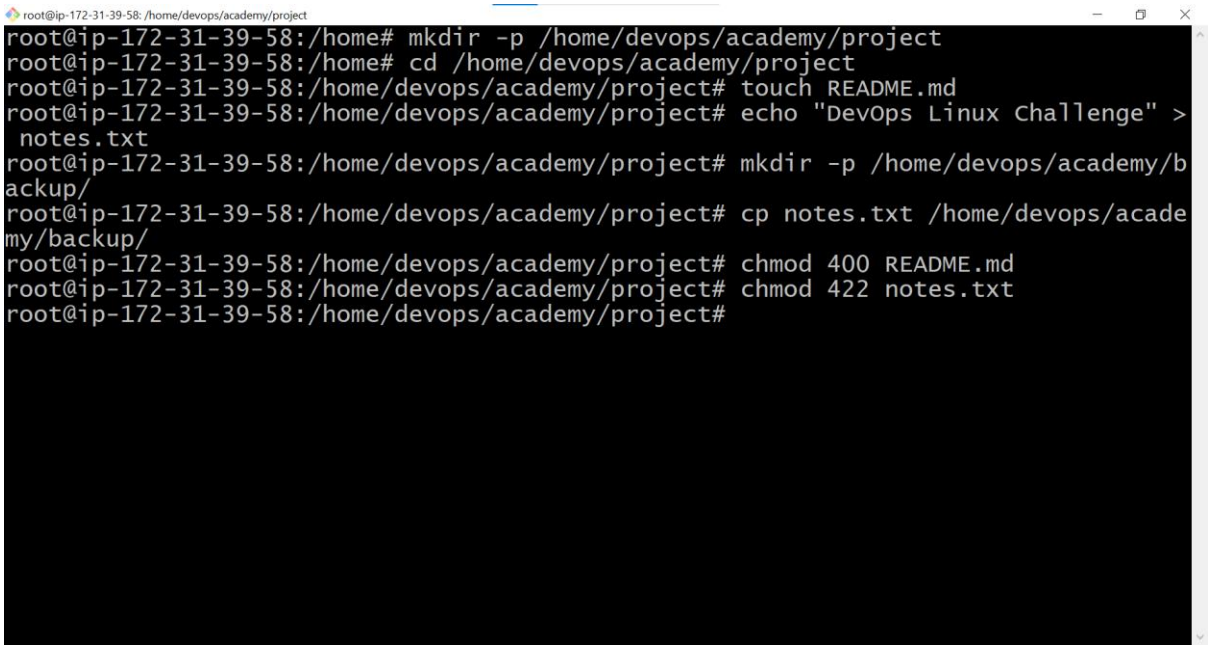
The objective of this challenge is to assess the students' ability to manage a Linux environment effectively, write shell scripts, and perform system administration tasks. They should be able to demonstrate their understanding of Linux file system hierarchy, user management, process management, networking,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

```
root@ip-172-31-39-58: /home/devops/academy/project
root@ip-172-31-39-58:/home/devops/academy/project# useradd student1
root@ip-172-31-39-58:/home/devops/academy/project# passwd student1
New password:
Retype new password:
passwd: password updated successfully
root@ip-172-31-39-58:/home/devops/academy/project# groupadd devops
root@ip-172-31-39-58:/home/devops/academy/project# usermod -aG devops student1
root@ip-172-31-39-58:/home/devops/academy/project# chown :devops /home/devops/academy/project
root@ip-172-31-39-58:/home/devops/academy/project# du -h /home > disk_usage.txt
root@ip-172-31-39-58:/home/devops/academy/project#
```

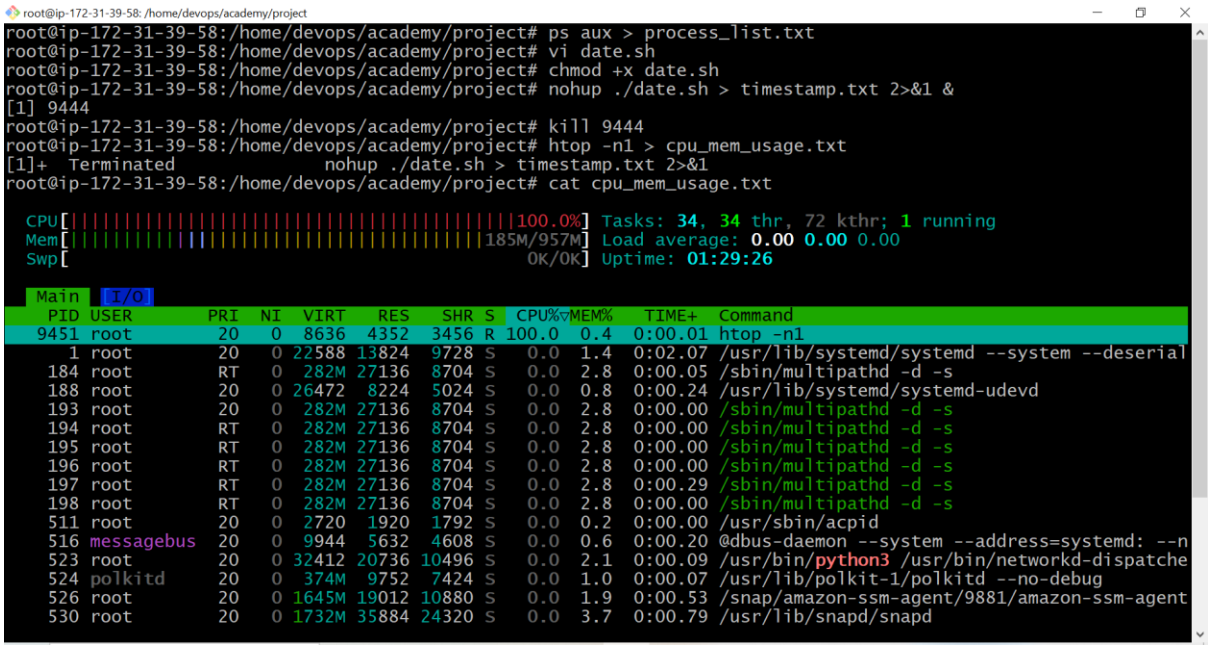
Intermediate Level

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`.

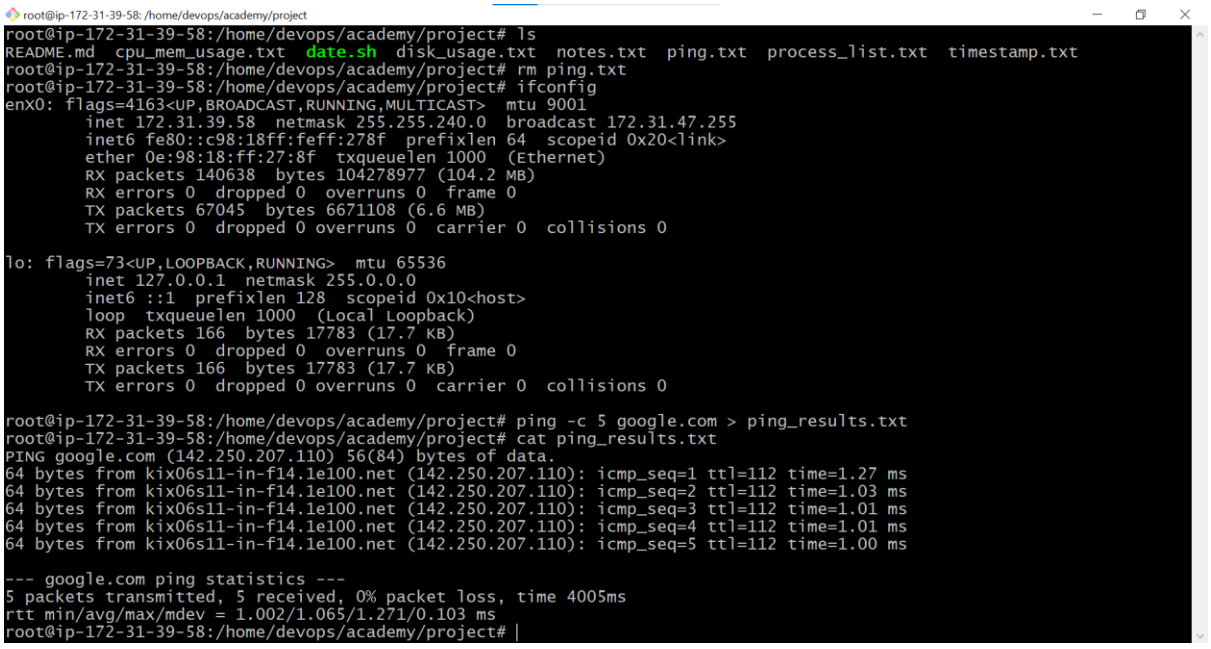
```
root@ip-172-31-39-58: /home/devops/academy/project
root@ip-172-31-39-58:/home/devops/academy/project# ps aux > process_list.txt
root@ip-172-31-39-58:/home/devops/academy/project# vi date.sh
root@ip-172-31-39-58:/home/devops/academy/project#
```

```
root@ip-172-31-39-58: /home/devops/academy/project
#!/bin/bash
#
#
while true
do
        echo $(date)
        sleep 60s
done
```



Task 4: Networking

1. **Network Configuration:**
   - Display the current network configuration using `ifconfig` or `ip addr`.
2. **Ping Test:** o 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.

```
root@ip-172-31-39-58: /home/devops/academy/project
root@ip-172-31-39-58:/home/devops/academy/project# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp6       0    440 ip-172-31-39-58.ap-:ssh 183.82.110.104.ac:50327 ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM      CONNECTED     8752
unix  3      [ ]         STREAM     CONNECTED     6906
unix  3      [ ]         STREAM     CONNECTED     8725
unix  2      [ ]         DGRAM      CONNECTED     6739
unix  3      [ ]         STREAM     CONNECTED     17706
unix  3      [ ]         DGRAM      CONNECTED     1704
unix  3      [ ]         STREAM     CONNECTED     6663     /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     17881    /run/systemd/journal/stdout
unix  2      [ ]         DGRAM                    8289     /run/user/1000/systemd/notify
unix  2      [ ]         DGRAM      CONNECTED     6636
unix  2      [ ]         DGRAM      CONNECTED     17832
unix  3      [ ]         STREAM     CONNECTED     17601    /run/dbus/system_bus_socket
unix  2      [ ]         DGRAM      CONNECTED     17894
unix  3      [ ]         STREAM     CONNECTED     6958     /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     17880    /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7325     /run/dbus/system_bus_socket
unix  3      [ ]         DGRAM      CONNECTED     17907
unix  3      [ ]         STREAM     CONNECTED     7393
unix  2      [ ]         DGRAM      CONNECTED     17716
unix  3      [ ]         STREAM     CONNECTED     8585
unix  3      [ ]         STREAM     CONNECTED     6952
unix  3      [ ]         SEQPACKET  CONNECTED     6793
unix  3      [ ]         STREAM     CONNECTED     17793
unix  2      [ ]         DGRAM      CONNECTED     7310
unix  2      [ ]         DGRAM      CONNECTED     17991
unix  2      [ ]         DGRAM      CONNECTED     6959
unix  2      [ ]         DGRAM                    19917
unix  3      [ ]         DGRAM      CONNECTED     1702     /run/systemd/notify
unix  3      [ ]         STREAM     CONNECTED     8294     /run/dbus/system_bus_socket
```

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).

```
root@ip-172-31-39-58: /home/devops/academy/project
root@ip-172-31-39-58:/home/devops/academy/project# which ufw
/usr/sbin/ufw
root@ip-172-31-39-58:/home/devops/academy/project# ufw status
Status: inactive
root@ip-172-31-39-58:/home/devops/academy/project# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@ip-172-31-39-58:/home/devops/academy/project# ufw allow http
Rule added
Rule added (v6)
root@ip-172-31-39-58:/home/devops/academy/project# ufw allow https
Rule added
Rule added (v6)
root@ip-172-31-39-58:/home/devops/academy/project#
```

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.

```
root@ip-172-31-3-177: /home/devops/academy
#!/bin/bash
#
#
backupDir="/home/devops/academy/backup"
projectDir="/home/devops/academy/project"

if [ -d "$projectDir" ]; then
        echo "project dir exists"
        tar -cf "$backupDir/project.tar.gz" "$projectDir"
else
        echo "project dir not exists"
fi
~
```

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`.

Crontab -e

```
root@ip-172-31-3-177: /home/devops/academy
#!/bin/bash
#
#
backupDir="/home/devops/academy/backup"
home="/home"


if [ -d $backupDir ]; then
        echo "backup dir exists"
        for file in $(find "$backupDir" -type f -mtime +7); do
                echo "i got file $file"
                rm -rf $file
                echo "deleted file $file"
        done
else
        echo "backup dir does not exists"
fi
~
```

```
root@ip-172-31-3-177:/home/devops/academy# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

0 2 * * * /home/devops/academy/cleanup.sh >> /home/devops/academy/cleanup.log
root@ip-172-31-3-177:/home/devops/academy#
```

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. o

```
root@ip-172-31-3-177: /home/devops/academy
#!/bin/bash
#
#
logDir="/home/devops/academy/logs"
logFile="timestamp.txt"
timestamp=$(date "+%Y%m%d_%H%M")
mb1=$(( 1024 * 1024))
fileSize=$(stat -c %s "$logDir/$logFile")

if [ $fileSize -ge $mb1 ]; then
        echo "file is greater than 1mb"
        echo "moving file to archive directory"
        mv $logDir/$logFile $logDir/archive/"timestamp_$timestamp.txt"
        touch $logDir/$logFile
else
        echo "file is less than 1mb"
fi
```

Advanced Level:

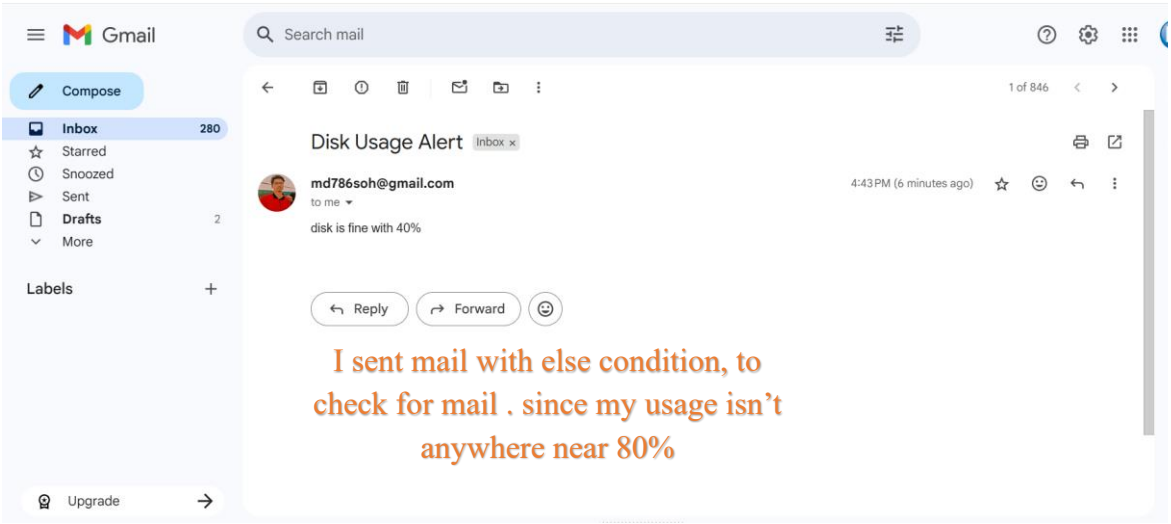
- **Advanced Shell Scripting:**

- 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%.

root@ip-172-31-3-177: /home/devops/academy

```
#!/bin/bash
#
#
logFile="/home/devops/academy/project/resource_usage.log"


echo "logging data --*--*--*-- $(date)"

cpuUsage=$(vmstat 1 2 | tail -1 | awk '{print 100 - $15}')
echo "------------- cpu usage ------------------"
echo $(vmstat | awk 'NR==1 {print $6}')
echo $(vmstat | awk 'NR==2 {print $13" "$14" "$15" "$16" "$17" "$18}')
echo $(vmstat | awk 'NR==3 {print $13" "$14" "$15" "$16" "$17" "$18}')
echo ""

diskUsage=$(df / | awk 'NR==2 {print$5}' | tr -d '%')
echo "------------- disk usage ------------------"
echo $(df / | awk 'NR==1')
echo $(df / | awk 'NR==2')
echo ""

echo "------------- memory usage ------------------"
echo $(free -m | awk 'NR==1')
echo $(free -m | awk 'NR==2')
echo ""

echo "cpu usage is $cpuUsage || disk usage is $diskUsage"

if [ $cpuUsage -ge 80 ]; then
        echo "cpu usage exceeded 80%" |  mail -s "Cpu usage Alert" sohail.bfl@gmail.com
else
        echo "cpu is fine with $cpuUsage %"
fi

if [ $diskUsage -ge 80 ]; then
        echo "disk usage exceeded 80%" | mail -s "Disk Usage Alert" sohail.bfl@gmail.com
else
        echo "disk is fine with $diskUsage%"
fi

echo "--**--**-- logging completed waiting for 5-minutes --**--**--**"
sleep 300s
```

43,0-1 All

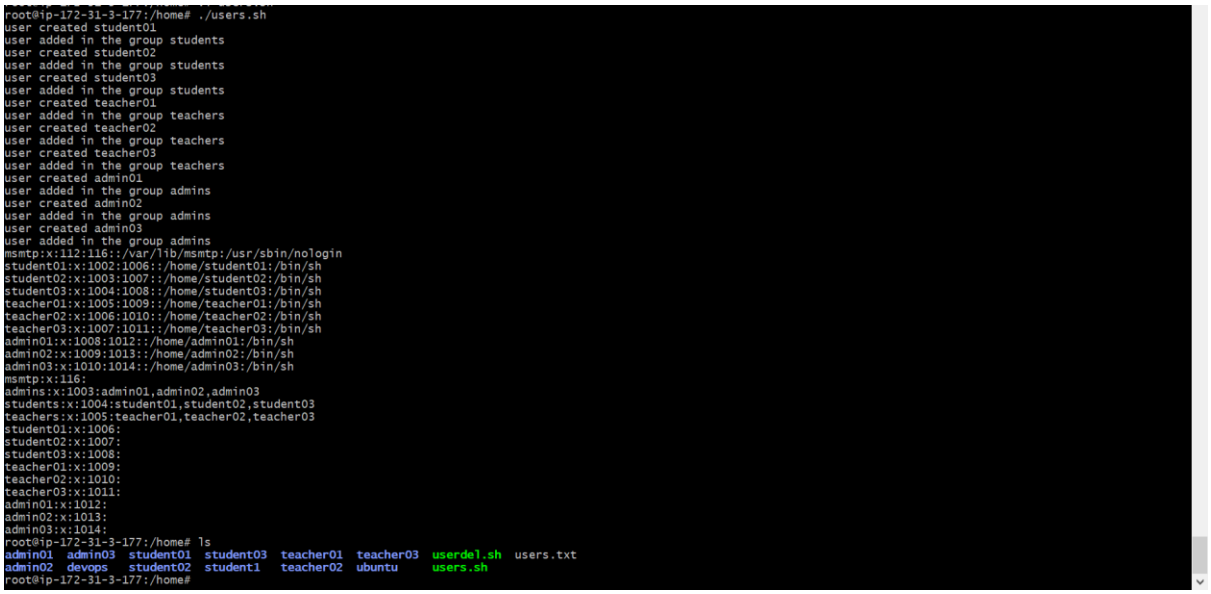


- **User Management Automation:**

- 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.

root@ip-172-31-3-177: /home

```
#!/bin/bash
#
#
usersFile="/home/users.txt"

password="12345678"
groupadd admins
groupadd students
groupadd teachers
for user in $(cat $usersFile); do
        useradd -m $user

        echo "user created $user"
        echo "$user:$password" | sudo chpasswd

        if [[ $user == student* ]]; then
                usermod -aG students $user
                echo "user added in the group students"
        elif [[ $user == admin* ]]; then
                usermod -aG admins $user
                echo "user added in the group admins"
        elif [[ $user == teacher* ]]; then
                usermod -aG teachers $user
                echo "user added in the group teachers"
        fi
done

tail -10 /etc/passwd
tail -13 /etc/group
~
~
```


root@ip-172-31-3-177:/home# ./users.sh
user created student01
user added in the group students
user created student02
user added in the group students
user created student03
user added in the group students
user created teacher01
user added in the group teachers
user created teacher02
user added in the group teachers
user created teacher03
user added in the group teachers
user created admin01
user added in the group admins
user created admin02
user added in the group admins
user created admin03
user added in the group admins
msmtp:x:112:116::/var/lib/msmtp:/usr/sbin/nologin
student01:x:1002:1006::/home/student01:/bin/sh
student02:x:1003:1007::/home/student02:/bin/sh
student03:x:1004:1008::/home/student03:/bin/sh
teacher01:x:1005:1009::/home/teacher01:/bin/sh
teacher02:x:1006:1010::/home/teacher02:/bin/sh
teacher03:x:1007:1011::/home/teacher03:/bin/sh
admin01:x:1008:1012::/home/admin01:/bin/sh
admin02:x:1009:1013::/home/admin02:/bin/sh
admin03:x:1010:1014::/home/admin03:/bin/sh
msmtp:x:116:
admins:x:1003:admin01,admin02,admin03
students:x:1004:student01,student02,student03
teachers:x:1005:teacher01,teacher02,teacher03
student01:x:1006:
student02:x:1007:
student03:x:1008:
teacher01:x:1009:
teacher02:x:1010:
teacher03:x:1011:
admin01:x:1012:
admin02:x:1013:
admin03:x:1014:
root@ip-172-31-3-177:/home# ls
admin01 admin03 student01 student03 teacher01 teacher03 userdel.sh users.txt
admin02 devops student02 student1 teacher02 ubuntu users.sh
root@ip-172-31-3-177:/home#