# 1. System Monitoring Setup

## 1. Installing and configuring htop or nmon

### 1.1. Installing htop (CPU, memory, and processes)

On **Ubuntu/Debian**:

sudo apt update

sudo apt install htop

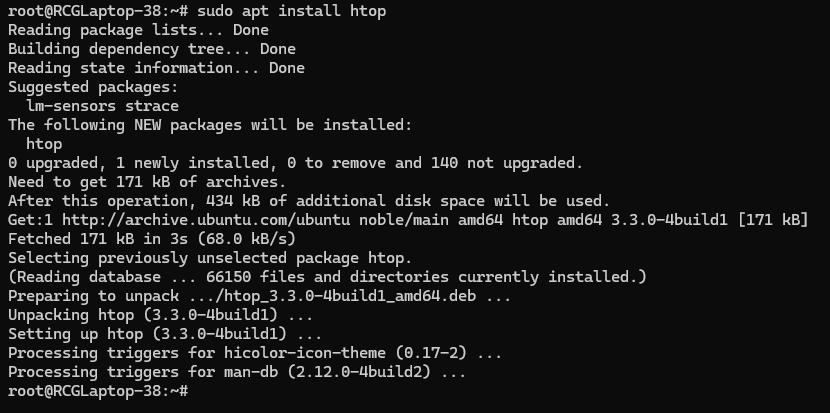
On **CentOS/RHEL**

sudo yum install htop

On **Fedora**

sudo dnf install htop

Run htop command: htop



A computer screen shot of a program

AI-generated content may be incorrect.

### 1.2. Installing nmon (Performance Monitoring)

On **Ubuntu/Debian**:

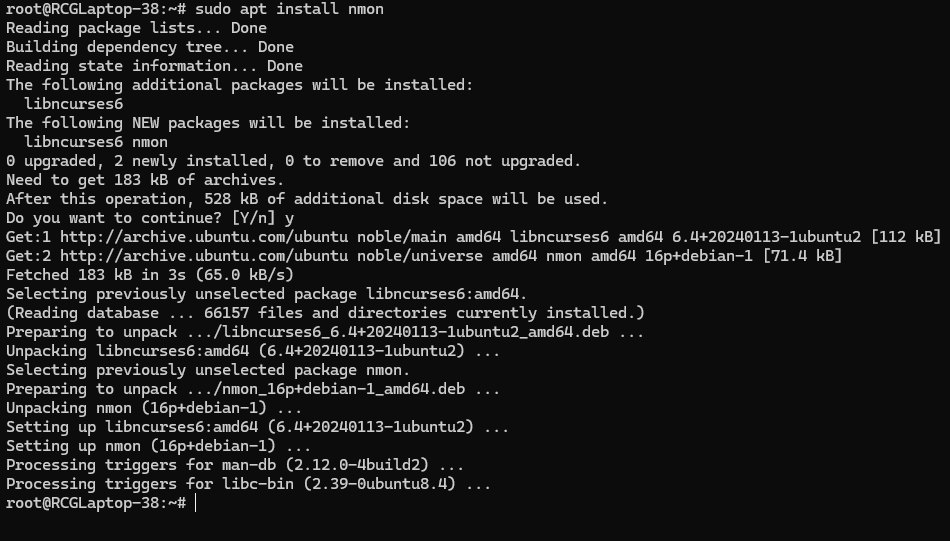
sudo apt install nmon

On **CentOS/RHEL**

sudo yum install nmon

On **Fedora**

sudo dnf install nmon



* Run nmon command nmon

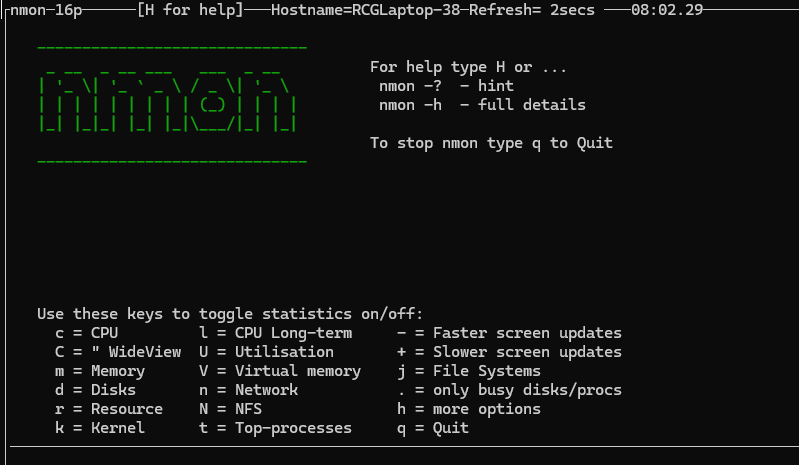
Use the following keys for monitoring different metrics:

c: for CPU usage

M: for memory usage

d: for disk usage

n: for network activity



### 1.3 Tracking Disk Usage with df and du

* Run df to check disk usage

df -h

* Run du to find disk usage for a specific directory

du -sh /path/to/directory

### 1.4 basic reporting

* Run the command below to save the log in /var/log directory

nmon -f /var/log/ -s 5 -c 60

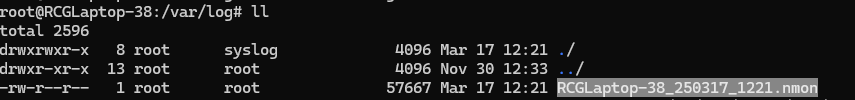
-f: Output to a file

-s 5: Collect data every 5 seconds

-c 60: Collect data for 60 cycles



* Check the log file in /var/log directory. file will be created by .nmon extensions.



# 2. User Management and Access Control

## 2.1 user accounts

Use command below to create the user. e.g. create the user account name as “Sarah”

useradd -m <username>

Note -m used for creating the home directory

e.g. useradd -m Sarah

check created user by using below command.

id <username>

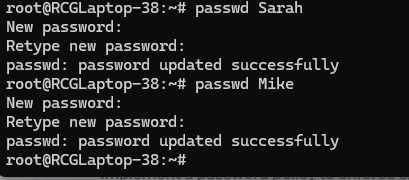
e.g. id Sarah



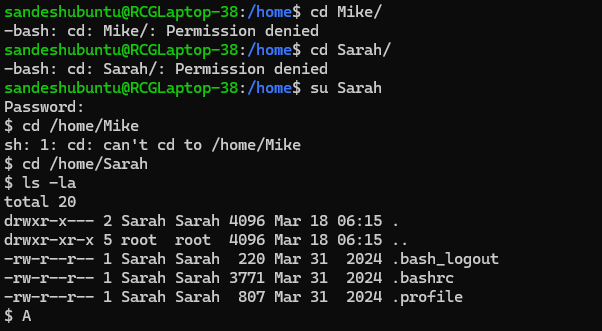
Set the password for created users

passwd <username>

e.g. passwd Sarah



Loing with crated user, e.g. su Sarah and make sure the login use cannot access the other user’s home directory



Create the workspace directory for users

mkdir /home/Sarah/workspace

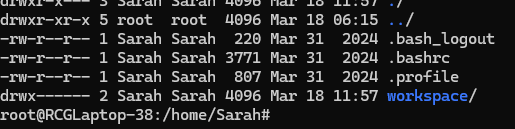
mkdir /home/Mike/workspace

cd /home/Sarah

change the ownership of workspace folder

chown Sarah:Sarah workspace

chown Mike:Mike workspace



### 2.2 Password policy

1. Install the perquisite liybarary

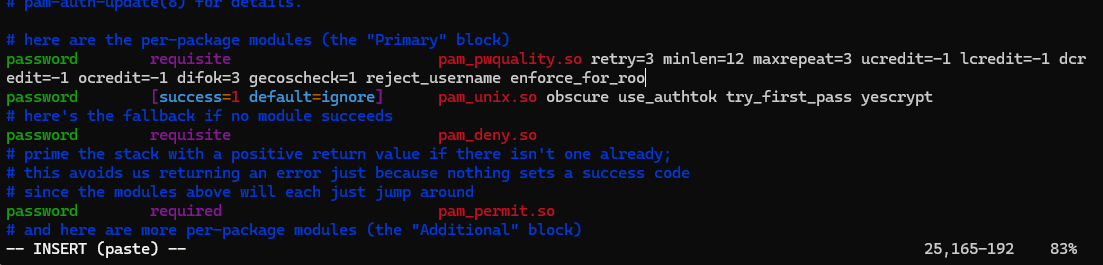
sudo apt-get -y install libpam-pwquality

1. Edit the /etc/pam.d/common-password file

etc/pam.d/common-password

1. Add below line next to retry=3,

minlen=12 maxrepeat=3 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1 difok=3 gecoscheck=1 reject\_username enforce\_for\_roo



**retry=3**: Prompt a user two times before returning with an error. Keep this low as you don’t want someone continually hammering at the door.

**minlen=12**: The password length should be as long as practically possible as this hugely increases the time to decryption.

**maxrepeat=3**: Allow a maximum of 2 repeated characters in your password.

**ucredit=-1**: Require at least 1 uppercase character.

**dcredit=-1**: Must have at least 1 lowercase character.

**dcredit=-1**: Must have at least 1 number.

**difok=3**: The number of characters in the new password that must not have been present in the previous password.

**gecoscheck=1**: Words in the [GECOS](https://en.wikipedia.org/wiki/General_Comprehensive_Operating_System) field of the user’s password entry aren’t contained in the new password.

**reject\_username**: Rejects the password if it contains the name of the user in either straight or reversed form.

**enforce\_for\_root**: Enforce the password policy for the root user.

# Web backup

## Backup Script

1. Create the Backup Directory

sudo mkdir -p /backups

sudo chmod 700 /backups

1. Backup Script for Sarah (Apache Server)

cd /home/Sarah

create script with below content

#!/bin/bash

# Variables

BACKUP\_DIR="/backups"

APACHE\_CONF="/etc/httpd"

APACHE\_ROOT="/var/www/html"

DATE=$(date +%Y-%m-%d)

BACKUP\_FILE="$BACKUP\_DIR/apache\_backup\_$DATE.tar.gz"

# Create the backup

tar -czf $BACKUP\_FILE $APACHE\_CONF $APACHE\_ROOT

# Verify the backup

if tar -tzf $BACKUP\_FILE &>/dev/null; then

echo "Backup verification successful for $BACKUP\_FILE" >> /backups/backup\_verification.log

else

echo "Backup verification FAILED for $BACKUP\_FILE" >> /backups/backup\_verification.log

fi

1. Make the Script Executable

sudo chmod +x /home/Sarah/apache\_backup.sh



1. Backup Script for Mike (Nginx Server)

cd /home/Mike

create script with below content

#!/bin/bash

# Variables

BACKUP\_DIR="/backups"

NGINX\_CONF="/etc/nginx"

NGINX\_ROOT="/usr/share/nginx/html"

DATE=$(date +%Y-%m-%d)

BACKUP\_FILE="$BACKUP\_DIR/nginx\_backup\_$DATE.tar.gz"

# Create the backup

tar -czf $BACKUP\_FILE $NGINX\_CONF $NGINX\_ROOT

# Verify the backup

if tar -tzf $BACKUP\_FILE &>/dev/null; then

echo "Backup verification successful for $BACKUP\_FILE" >> /backups/backup\_verification.log

else

echo "Backup verification FAILED for $BACKUP\_FILE" >> /backups/backup\_verification.log

fi

1. Make the Script Executable

sudo chmod +x /home/Mike/nginx\_backup.sh



## Schedule

Sarah’s Cron Job

1. Run the command below

sudo crontab -e

1. Add the below line

0 0 \* \* 2 /home/Sarah/apache\_backup.sh

Mike’s Cron Job

1. Run the command below

sudo crontab -e

1. Add the below line

0 0 \* \* 2 /home/Mike/nginx\_backup.sh