Lab Project - 2

Objective: Shell scripting for Automation labs

PRE-REQUISITES:

Oracle VirtualBox or VMWare, Ubuntu installed.

Lab 1: Automating System Backup

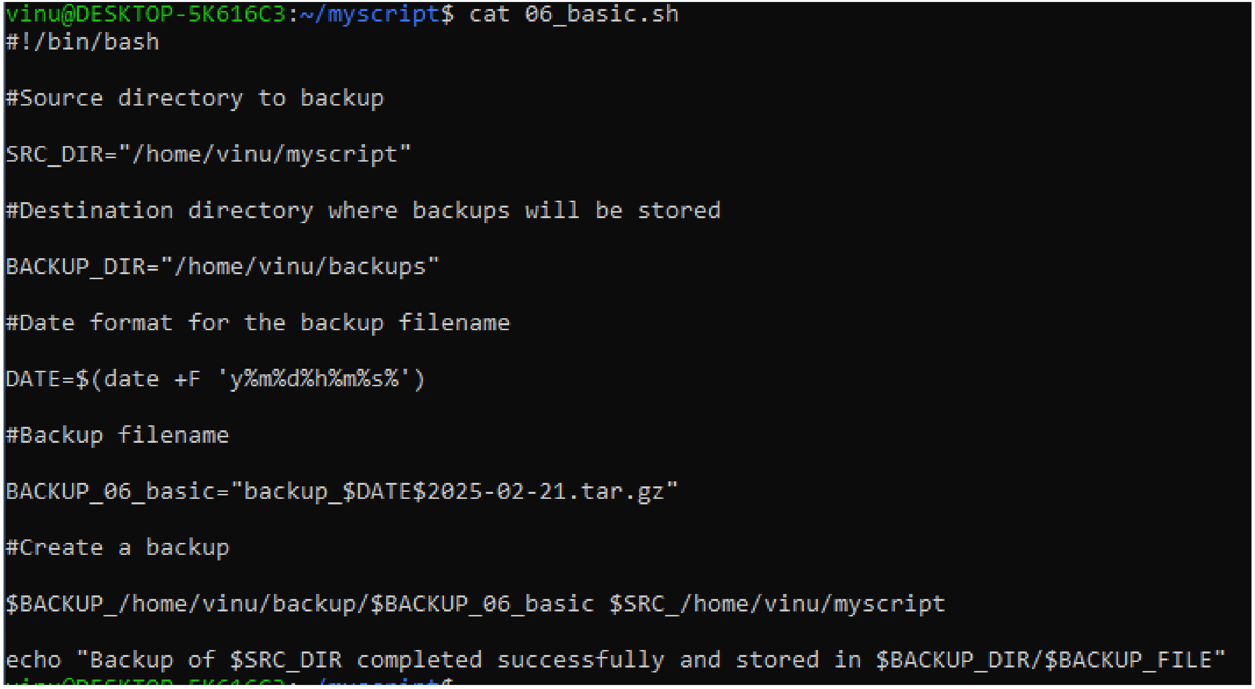
Objective:

• Learn to create a shell script that automates the process of backing up files and directories.

Tasks:

1.Create a Backup Script:
 O Create a script called backup.sh to automate the backup of a directory.

Add the following content to the script:



```
vinu@DESKTOP-5K616C3:~/myscript$ cat 06_basic.sh
#!/bin/bash

#Source directory to backup

SRC_DIR="/home/vinu/myscript"

#Destination directory where backups will be stored

BACKUP_DIR="/home/vinu/backups"

#Date format for the backup filename

DATE=$(date +F 'y%m%d%h%m%s%')

#Backup filename

BACKUP_06_basic="backup_$DATE$2025-02-21.tar.gz"

#Create a backup

$BACKUP_/home/vinu/backup/$BACKUP_06_basic $SRC_/home/vinu/myscript

echo "Backup of $SRC_DIR completed successfully and stored in $BACKUP_DIR/$BACKUP_FILE"
```

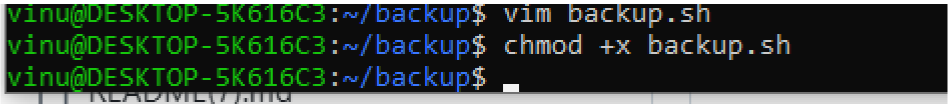
2.   Make the Script Executable:

bash

Copy code

chmod +x backup.sh

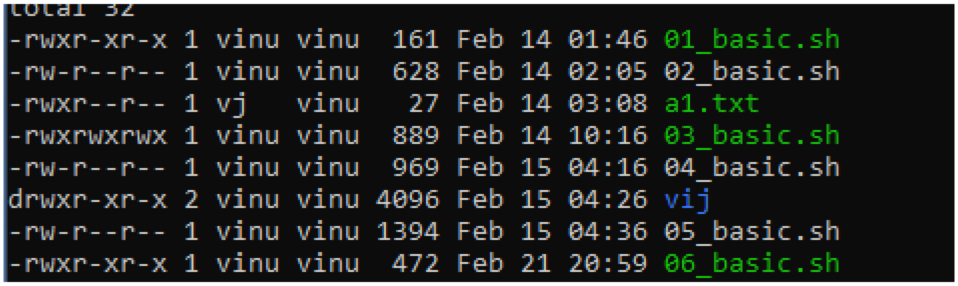
Output:-



```
vinu@DESKTOP-5K616C3:~/backup$ vim backup.sh
vinu@DESKTOP-5K616C3:~/backup$ chmod +x backup.sh
vinu@DESKTOP-5K616C3:~/backup$ _
```

3.   Run the Backup Script:

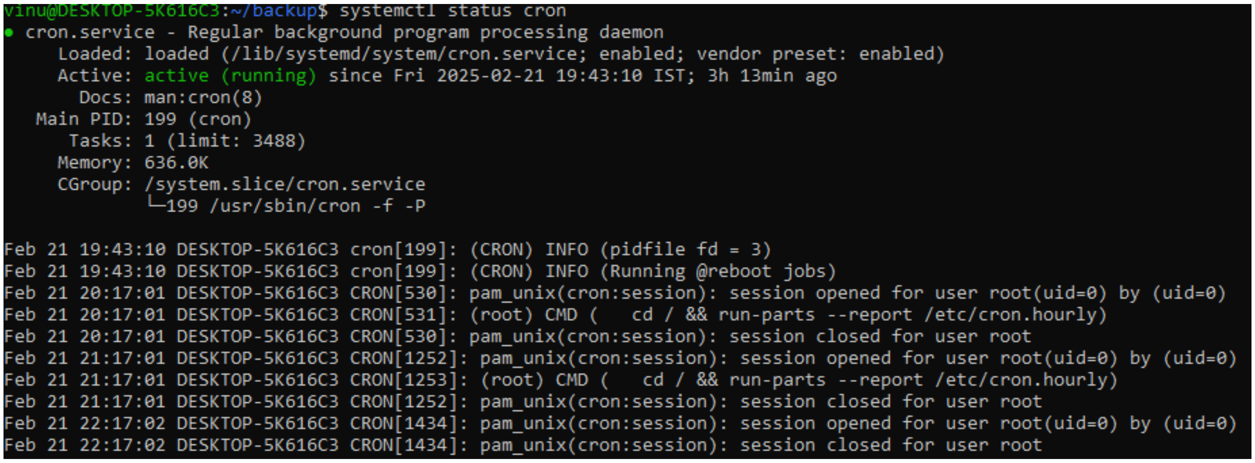
o   Run the script to create a backup:



```
total 32
-rwxr-xr-x 1 vinu vinu  161 Feb 14 01:46 01_basic.sh
-rw-r--r-- 1 vinu vinu  628 Feb 14 02:05 02_basic.sh
-rwxr--r-- 1 vj   vinu   27 Feb 14 03:08 a1.txt
-rwxrwxrwx 1 vinu vinu  889 Feb 14 10:16 03_basic.sh
-rw-r--r-- 1 vinu vinu  969 Feb 15 04:16 04_basic.sh
drwxr-xr-x 2 vinu vinu 4096 Feb 15 04:26 vij
-rw-r--r-- 1 vinu vinu 1394 Feb 15 04:36 05_basic.sh
-rwxr-xr-x 1 vinu vinu  472 Feb 21 20:59 06_basic.sh
```

4.   Schedule the Backup Using Cron:

o   Open the crontab file to schedule a daily backup.

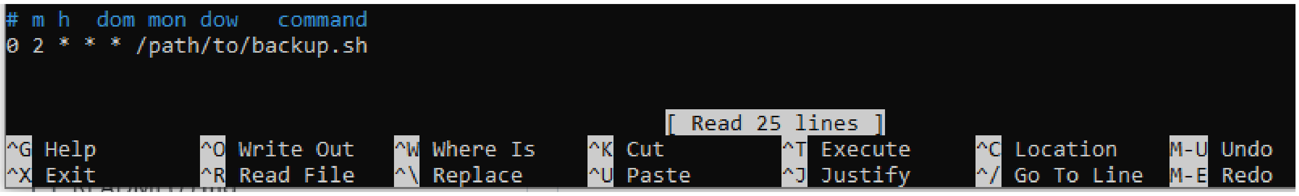


```
vinu@DESKTOP-5K616C3:~/backup$ systemctl status cron
● cron.service - Regular background program processing daemon
     Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2025-02-21 19:43:10 IST; 3h 13min ago
       Docs: man:cron(8)
   Main PID: 199 (cron)
      Tasks: 1 (limit: 3488)
     Memory: 636.0K
     CGroup: /system.slice/cron.service
             └─199 /usr/sbin/cron -f -P

Feb 21 19:43:10 DESKTOP-5K616C3 cron[199]: (CRON) INFO (pidfile fd = 3)
Feb 21 19:43:10 DESKTOP-5K616C3 cron[199]: (CRON) INFO (Running @reboot jobs)
Feb 21 20:17:01 DESKTOP-5K616C3 CRON[530]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Feb 21 20:17:01 DESKTOP-5K616C3 CRON[531]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Feb 21 20:17:01 DESKTOP-5K616C3 CRON[530]: pam_unix(cron:session): session closed for user root
Feb 21 21:17:01 DESKTOP-5K616C3 CRON[1252]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Feb 21 21:17:01 DESKTOP-5K616C3 CRON[1253]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Feb 21 21:17:01 DESKTOP-5K616C3 CRON[1252]: pam_unix(cron:session): session closed for user root
Feb 21 22:17:02 DESKTOP-5K616C3 CRON[1434]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
Feb 21 22:17:02 DESKTOP-5K616C3 CRON[1434]: pam_unix(cron:session): session closed for user root
```

crontab -e

o Add the following cron job to run the backup script every day at 2 AM

0 2 * * * /path/to/backup.sh



```
# m h  dom mon dow   command
0 2 * * * /path/to/backup.sh
```

[Read 25 lines]

Lab 2: Automating System Updates

Objective:

• Automate system package updates using a shell script to ensure the system is always up to date

1.Create the Update Script:

O Create a script called auto_update.sh to automate system updates for Debian-based systems (e.g., Ubuntu) or Red Hat-based systems (e.g., CentOS).

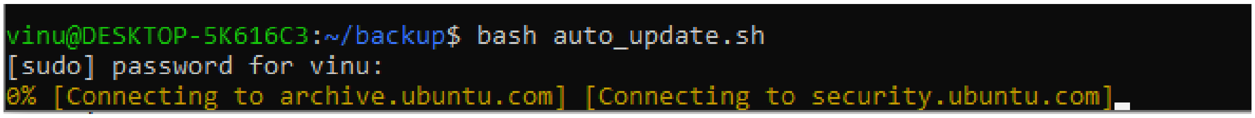
O Add the following content for Debian/Ubuntu

nano auto_update.sh

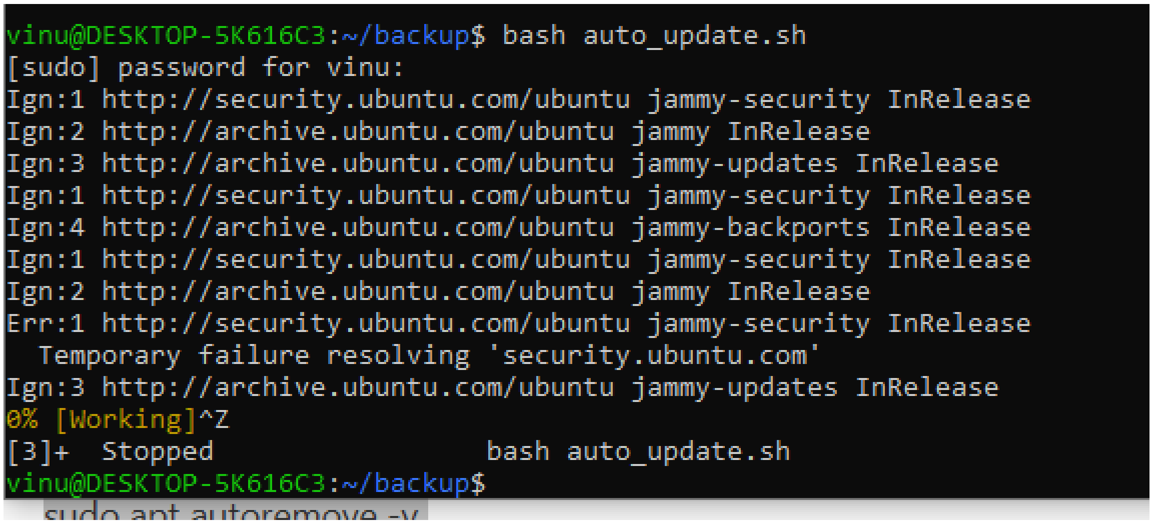
bash

Copy code

```
#!/bin/bash

#Update package list

sudo apt update

#Upgrade installed packages

sudo apt upgrade -y

#Clean up unused packages

sudo apt autoremove -y

echo "System update completed."
```



```
vinu@DESKTOP-5K616C3:~/backup$ bash auto_update.sh
[sudo] password for vinu:
0% [Connecting to archive.ubuntu.com] [Connecting to security.ubuntu.com]
```



```
vinu@DESKTOP-5K616C3:~/backup$ bash auto_update.sh
[sudo] password for vinu:
Ign:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Err:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
  Temporary failure resolving 'security.ubuntu.com'
Ign:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
0% [Working]^Z
[3]+  Stopped                 bash auto_update.sh
vinu@DESKTOP-5K616C3:~/backup$
```

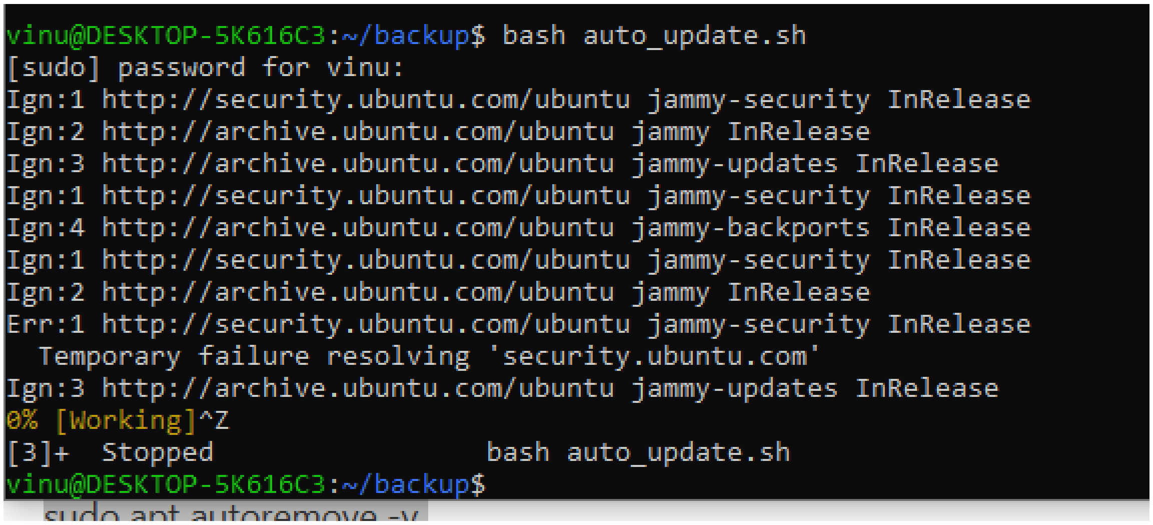
sudo apt autoremove -y

- **For Red Hat/CentOS, replace the content with:**

bash

Copy code

```
#!/bin/bash

#Update package list and upgrade packages

sudo yum update -y

#Clean up unused packages

sudo yum autoremove -y

echo "System update completed."
```

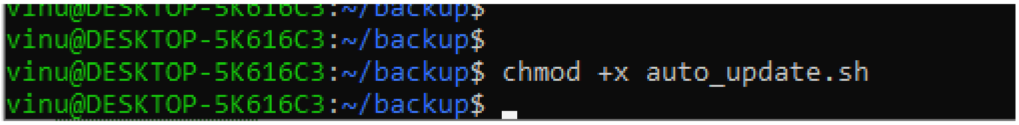


```
vinu@DESKTOP-5K616C3:~/backup$ bash auto_update.sh
[sudo] password for vinu:
Ign:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Err:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
  Temporary failure resolving 'security.ubuntu.com'
Ign:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
0% [Working]^Z
[3]+  Stopped                 bash auto_update.sh
vinu@DESKTOP-5K616C3:~/backup$
```

sudo apt autoremove -y

2.Make the Script Executable:

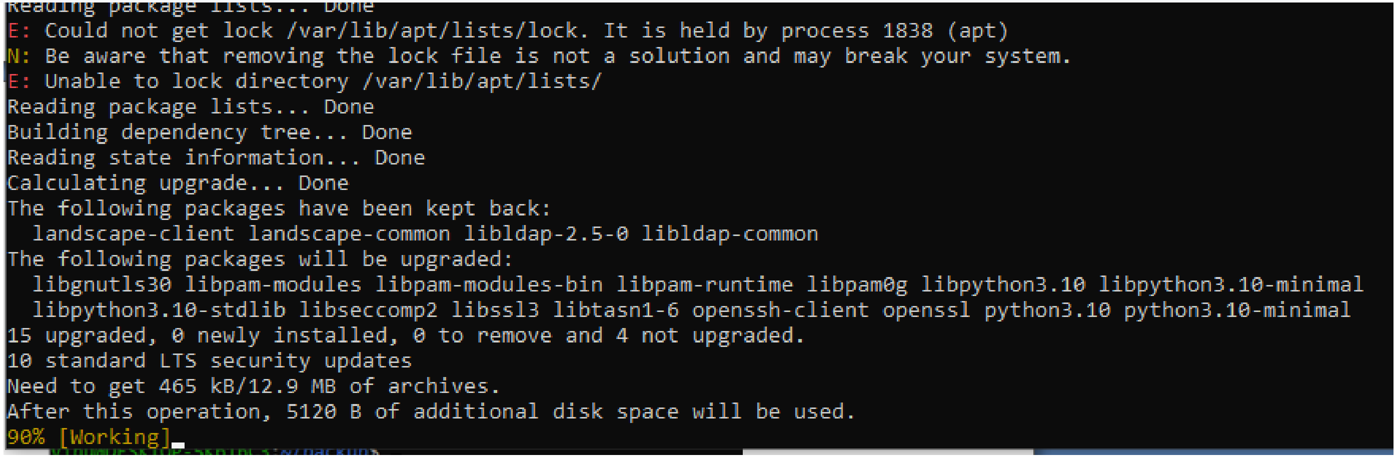
chmod +x auto_update.sh



```
vinu@DESKTOP-5K616C3:~/backup$
vinu@DESKTOP-5K616C3:~/backup$
vinu@DESKTOP-5K616C3:~/backup$ chmod +x auto_update.sh
vinu@DESKTOP-5K616C3:~/backup$ _
```

3.Run the Update Script:

- **Run the script to perform an update:**



```
Reading package lists... Done
E: Could not get lock /var/lib/apt/lists/lock. It is held by process 1838 (apt)
N: Be aware that removing the lock file is not a solution and may break your system.
E: Unable to lock directory /var/lib/apt/lists/
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  landscape-client landscape-common libldap-2.5-0 libldap-common
The following packages will be upgraded:
  libgnutls30 libpam-modules libpam-modules-bin libpam-runtime libpam0g libpython3.10 libpython3.10-minimal
  libpython3.10-stdlib libseccomp2 libssl3 libtasn1-6 openssh-client openssl python3.10 python3.10-minimal
15 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
10 standard LTS security updates
Need to get 465 kB/12.9 MB of archives.
After this operation, 5120 B of additional disk space will be used.
90% [Working]_
```

4.Schedule Automatic Updates Using Cron:

* Open the crontab file to schedule the update script to run weekly.

bash

Copy code

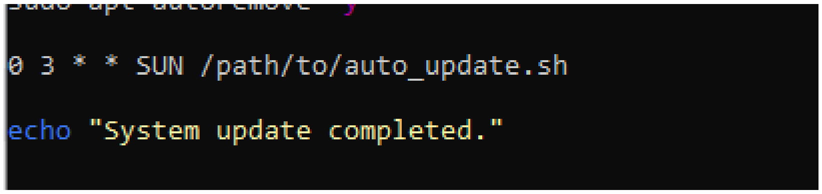
crontab –e

- Add the following cron job to run the script every Sunday at 3 AM:

bash

Copy code

0 3 * * SUN /path/to/auto_update.sh



```
sudo apt autoremove -y
0 3 * * SUN /path/to/auto_update.sh

echo "System update completed."
```

Lab 4: Automating Log Rotation

Objective:

• Automate the process of rotating logs to avoid disk space issues.

Tasks:

1. Create a Log Rotation Script:
 o Create a script called log_rotation.sh to rotate log files in a directory.

bash

Copy code

nano log_rotation.sh

 * Add the following content:

bash

Copy code

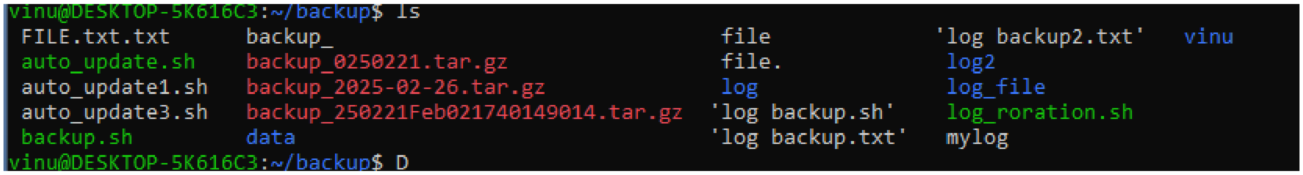
#!/bin/bash

#Directory where logs are stored

LOG_DIR="/var/log/myapp"

#Backup directory for rotated logs

```
BACKUP_DIR="/var/log/myapp/backup"

#Log file to rotate

LOG_FILE="myapp.log"

#Create backup directory if it doesn't exist

mkdir -p $BACKUP_DIR

#Rotate the log file by renaming it with a timestamp

mv $LOG_DIR/$LOG_FILE $BACKUP_DIR/$LOG_FILE-$(date +%F-%T)
```



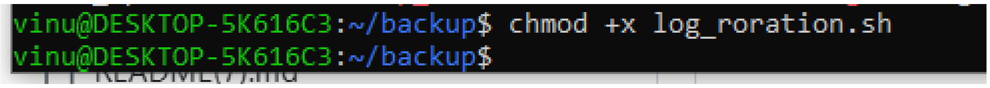
```
vinu@DESKTOP-5K616C3:~/backup$ ls
 FILE.txt.txt      backup_                              file              'log backup2.txt'   vinu
 auto_update.sh    backup_0250221.tar.gz                file.              log2
 auto_update1.sh   backup_2025-02-26.tar.gz             log                log_file
 auto_update3.sh   backup_250221Feb021740149014.tar.gz 'log backup.sh'     log_roration.sh
 backup.sh         data                                'log backup.txt'    mylog
vinu@DESKTOP-5K616C3:~/backup$ D
```

2.Make the Script Executable:

bash

Copy code

```
chmod +x log_rotation.sh
```



```
vinu@DESKTOP-5K616C3:~/backup$ chmod +x log_roration.sh
vinu@DESKTOP-5K616C3:~/backup$
```

3.Run the Script:

o Run the script to perform log rotation:

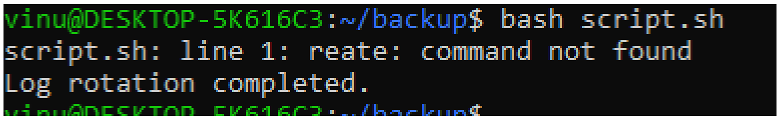
#Create a new empty log file

touch $LOG_DIR/$LOG_FILE

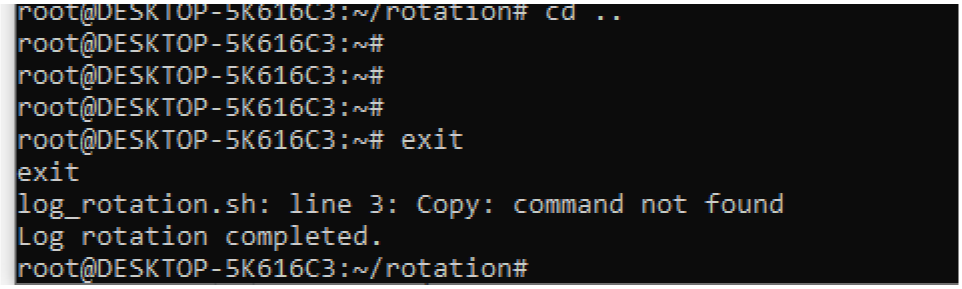
#Set permissions on the new log file

chmod 644 $LOG_DIR/$LOG_FILE

echo "Log rotation completed."



```
vinu@DESKTOP-5K616C3:~/backup$ bash script.sh
script.sh: line 1: reate: command not found
Log rotation completed.
vinu@DESKTOP-5K616C3:~/backup$
```



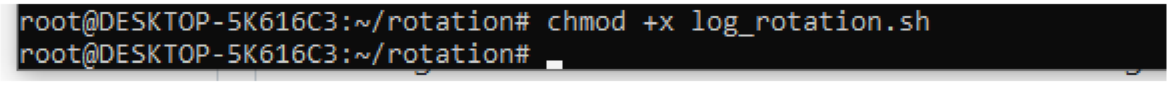
```
root@DESKTOP-5K616C3:~/rotation# cd ..
root@DESKTOP-5K616C3:~#
root@DESKTOP-5K616C3:~#
root@DESKTOP-5K616C3:~#
root@DESKTOP-5K616C3:~# exit
exit
log_rotation.sh: line 3: Copy: command not found
Log rotation completed.
root@DESKTOP-5K616C3:~/rotation#
```

2. Make the Script Executable:

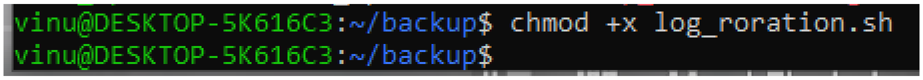
bash

Copy code

chmod +x log_rotation.sh



```
root@DESKTOP-5K616C3:~/rotation# chmod +x log_rotation.sh
root@DESKTOP-5K616C3:~/rotation#
```



```
vinu@DESKTOP-5K616C3:~/backup$ chmod +x log_roration.sh
vinu@DESKTOP-5K616C3:~/backup$
```

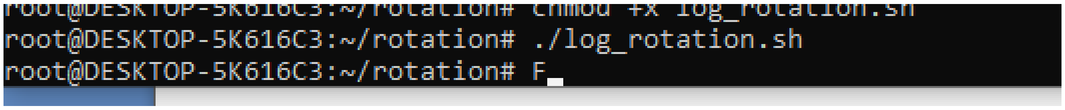
3. Run the Script:

o Run the script to perform log rotation:

bash

Copy code

./log_rotation.sh



```
root@DESKTOP-5K616C3:~/rotation# chmod +x log_rotation.sh
root@DESKTOP-5K616C3:~/rotation# ./log_rotation.sh
root@DESKTOP-5K616C3:~/rotation# F
```

Lab 3: Automating Disk Space Monitoring

Objective:

• Automate disk space monitoring and send an alert when disk space usage exceeds a threshold.

Tasks:

1.Create a Disk Space Monitoring Script:

oCreate a script called disk_space_monitor.sh to check disk usage and send an email alert if usage exceeds 80%.

bash

Copy code

nano disk_space_monitor.sh

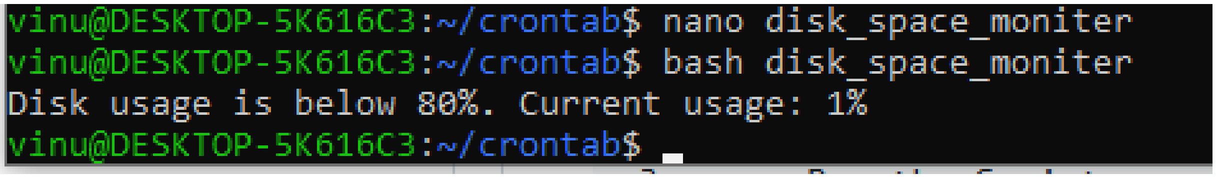
o **Add the following content:**

bash

Copy code

```
#!/bin/bash

#Set the threshold for disk space usage

THRESHOLD=80

#Get the current disk usage percentage

DISK_USAGE=$(df / | grep / | awk '{ print $5 }' | sed 's/%//g')

#Check if disk usage is above the threshold

if [ $DISK_USAGE -gt $THRESHOLD ]; then

echo "Warning: Disk usage is above $THRESHOLD%.

Current usage: $DISK_USAGE%" | mail -s "Disk Space

Alert" user@example.com

else

echo "Disk usage is below $THRESHOLD%. Current usage: $DISK_USAGE%"

fi
```

Ans:-



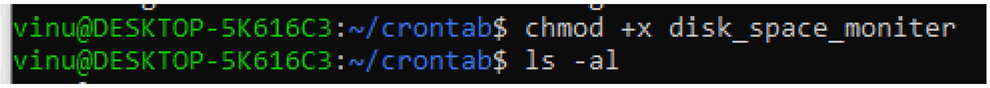
```
vinu@DESKTOP-5K616C3:~/crontab$ nano disk_space_moniter
vinu@DESKTOP-5K616C3:~/crontab$ bash disk_space_moniter
Disk usage is below 80%. Current usage: 1%
vinu@DESKTOP-5K616C3:~/crontab$ _
```

2. Make the Script Executable:

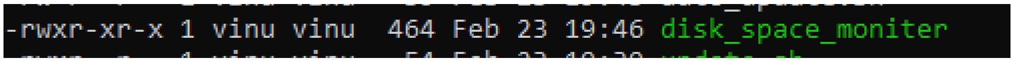
bash

Copy code

chmod +x disk_space_monitor.sh



```
vinu@DESKTOP-5K616C3:~/crontab$ chmod +x disk_space_moniter
vinu@DESKTOP-5K616C3:~/crontab$ ls -al
```



```
-rwxr-xr-x 1 vinu vinu  464 Feb 23 19:46 disk_space_moniter
```

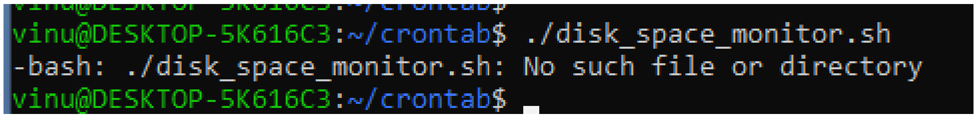
3. Run the Script:

o Run the script to check disk usage:

bash

Copy code

./disk_space_monitor.sh



```
vinu@DESKTOP-5K616C3:~/crontab$ ./disk_space_monitor.sh
-bash: ./disk_space_monitor.sh: No such file or directory
vinu@DESKTOP-5K616C3:~/crontab$
```

4.Schedule the Monitoring Script Using Cron:

o Open the crontab file to schedule the script to run every day at 6 AM:

bash

Copy code

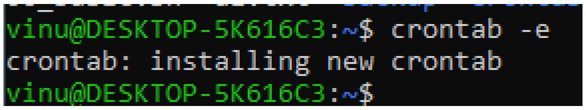
```
crontab -e
```

- o Add the following cron job:

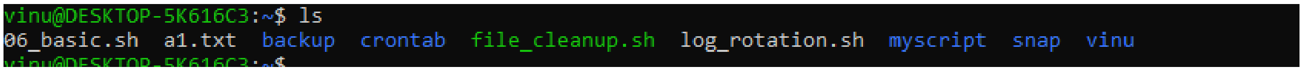
bash

Copy code

```
0 6 * * * /path/to/disk_space_monitor.sh
```



```
vinu@DESKTOP-5K616C3:~$ crontab -e
crontab: installing new crontab
vinu@DESKTOP-5K616C3:~$
```



```
vinu@DESKTOP-5K616C3:~$ ls
06_basic.sh  a1.txt  backup  crontab  file_cleanup.sh  log_rotation.sh  myscript  snap  vinu
vinu@DESKTOP-5K616C3:~$
```

Lab 4: Automating Log Rotation

Objective:

- Automate the process of rotating logs to avoid disk space issues.

Tasks:

1. Create a Log Rotation Script:
   - o Create a script called log_rotation.sh to rotate log files in a directory.

bash

Copy code

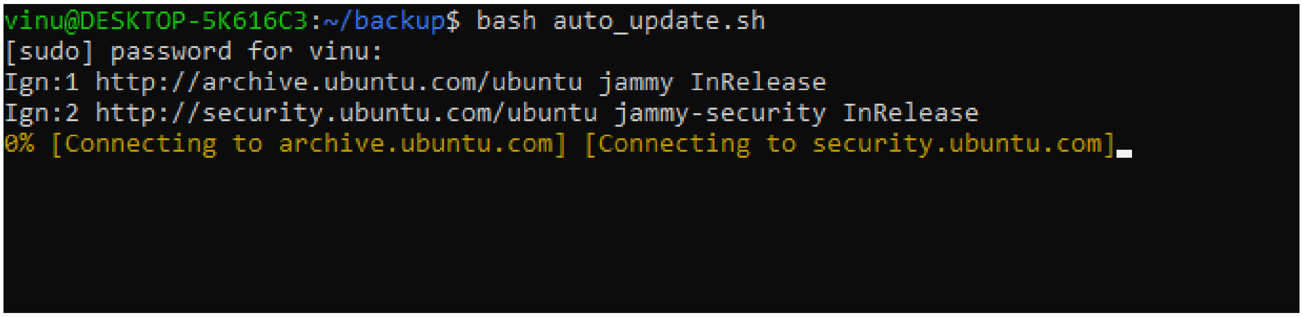
```
nano log_rotation.sh
```

- Add the following content:

bash

Copy code

```
#!/bin/bash
#Directory where logs are stored
LOG_DIR="/var/log/myapp"
#Backup directory for rotated logs
BACKUP_DIR="/var/log/myapp/backup"
#Log file to rotate
LOG_FILE="myapp.log"
#Create backup directory if it doesn't exist
mkdir -p $BACKUP_DIR
#Rotate the log file by renaming it with a timestamp
mv $LOG_DIR/$LOG_FILE $BACKUP_DIR/$LOG_FILE-$(date +%F-%T)
#Create a new empty log file
touch $LOG_DIR/$LOG_FILE
#Set permissions on the new log file
chmod 644 $LOG_DIR/$LOG_FILE
echo "Log rotation completed."
```



```
vinu@DESKTOP-5K616C3:~/backup$ bash auto_update.sh
[sudo] password for vinu:
Ign:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
0% [Connecting to archive.ubuntu.com] [Connecting to security.ubuntu.com]
```

2.   Make the Script Executable:

bash

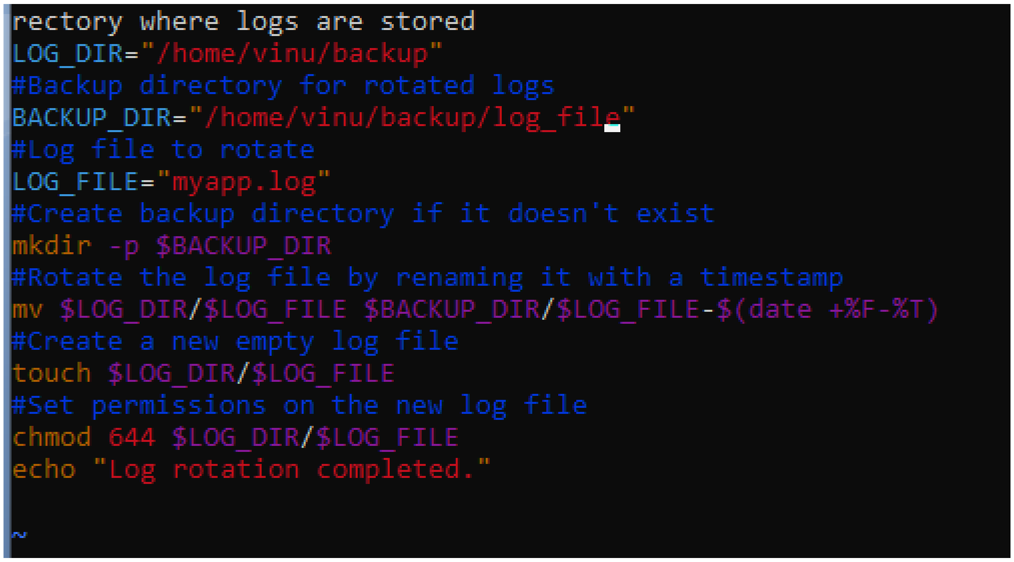
Copy code

chmod +x log_rotation.sh

3.Run the Script:

o Run the script to perform log rotation:

bash



```
rectory where logs are stored
LOG_DIR="/home/vinu/backup"
#Backup directory for rotated logs
BACKUP_DIR="/home/vinu/backup/log_file"
#Log file to rotate
LOG_FILE="myapp.log"
#Create backup directory if it doesn't exist
mkdir -p $BACKUP_DIR
#Rotate the log file by renaming it with a timestamp
mv $LOG_DIR/$LOG_FILE $BACKUP_DIR/$LOG_FILE-$(date +%F-%T)
#Create a new empty log file
touch $LOG_DIR/$LOG_FILE
#Set permissions on the new log file
chmod 644 $LOG_DIR/$LOG_FILE
echo "Log rotation completed."

~
```

Copy code

./log_rotation.sh

4.Schedule Log Rotation Using Cron:

O Open the crontab file to schedule the log rotation to run every day at midnight:

bash

Copy code

```
crontab -e
```

o Add the following cron job:

bash

Copy code

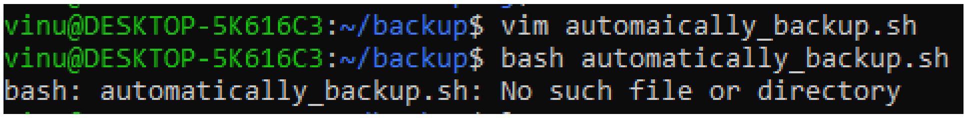
```
0 0 * * * /path/to/log_rotation.sh
```

bash

Copy code

```
#!/bin/bash

#Directory where logs are stored

LOG_DIR="/var/log/myapp"

#Backup directory for rotated logs

BACKUP_DIR="/var/log/myapp/backup"

#Log file to rotate
```

```
LOG_FILE="myapp.log"

#Create backup directory if it doesn't exist

mkdir -p $BACKUP_DIR

#Rotate the log file by renaming it with a timestamp

mv $LOG_DIR/$LOG_FILE $BACKUP_DIR/$LOG_FILE-$(date +%F-%T)

#Create a new empty log file

touch $LOG_DIR/$LOG_FILE

#Set permissions on the new log file

chmod 644 $LOG_DIR/$LOG_FILE

echo "Log rotation completed."
```



```
vinu@DESKTOP-5K616C3:~/backup$ vim automaically_backup.sh
vinu@DESKTOP-5K616C3:~/backup$ bash automatically_backup.sh
bash: automatically_backup.sh: No such file or directory
```

Lab 5: Automating User Account Management

Objective:

• Automate the process of adding and removing users in Linux.

Tasks:

1. Create a Script to Add Users:
 o Create a script called add_user.sh to automate adding a user to the system.

bash

Copy code

```
nano add_user.sh
```

 o Add the following content:

bash

Copy code

```
#!/bin/bash

#Check if username is provided

if [ -z "$1" ]; then

echo "Error: Please provide a username."

exit 1

fi

#Add user to the system

sudo useradd $1
```

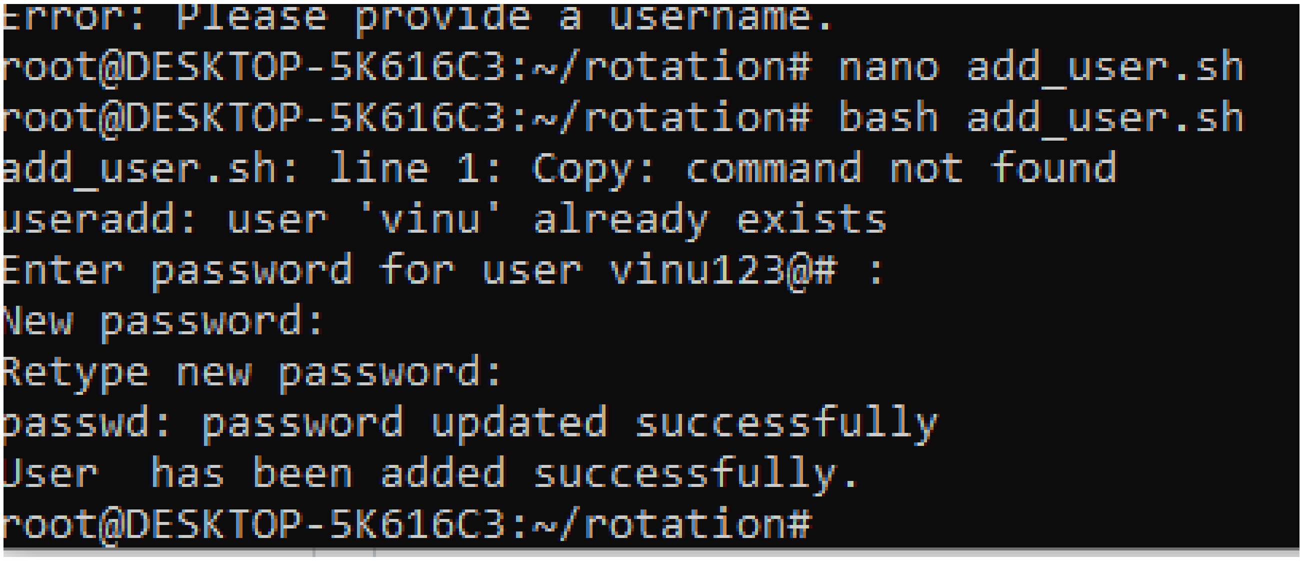
#Set password for the new user

echo "Enter password for user $1:"

sudo passwd $1

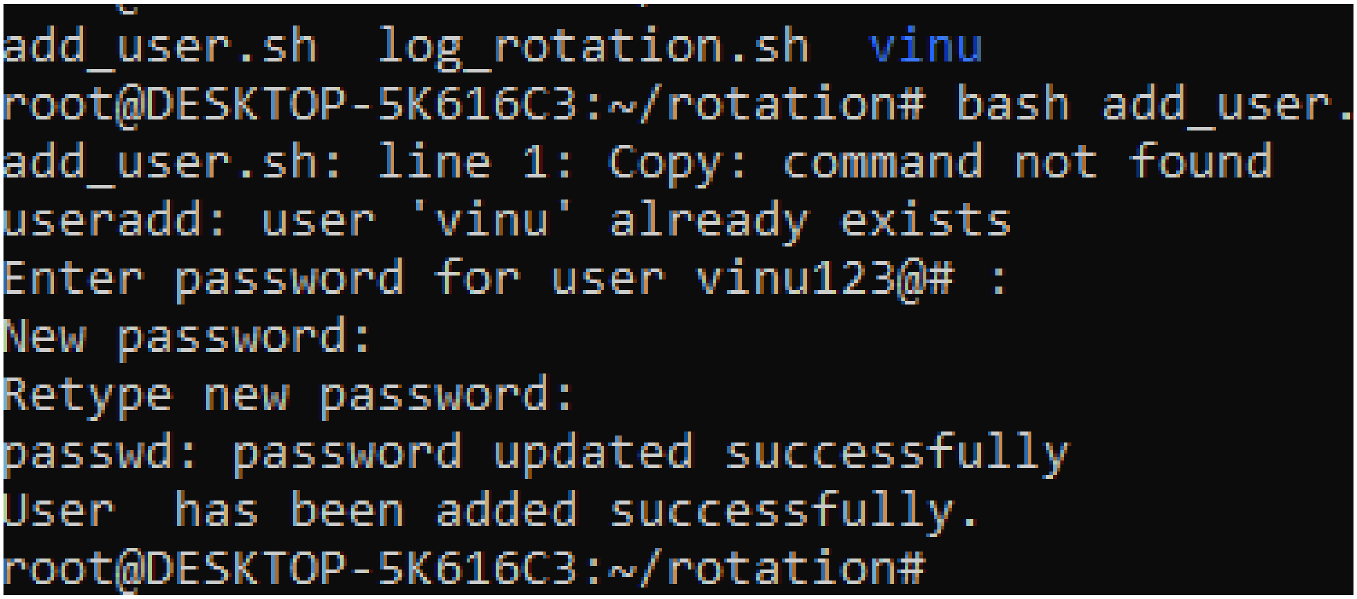
echo "User $1 has been added successfully."

Ans:-



```
Error: Please provide a username.
root@DESKTOP-5K616C3:~/rotation# nano add_user.sh
root@DESKTOP-5K616C3:~/rotation# bash add_user.sh
add_user.sh: line 1: Copy: command not found
useradd: user 'vinu' already exists
Enter password for user vinu123@# :
New password:
Retype new password:
passwd: password updated successfully
User  has been added successfully.
root@DESKTOP-5K616C3:~/rotation#
```

:-Create a script called add_user.sh to automate adding a user to the system.



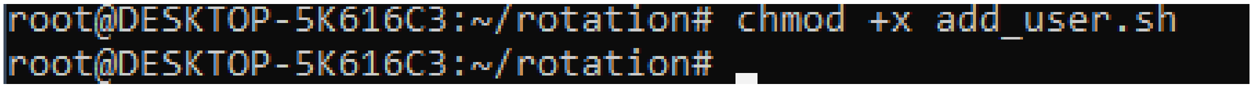
```
add_user.sh  log_rotation.sh  vinu
root@DESKTOP-5K616C3:~/rotation# bash add_user.
add_user.sh: line 1: Copy: command not found
useradd: user 'vinu' already exists
Enter password for user vinu123@# :
New password:
Retype new password:
passwd: password updated successfully
User  has been added successfully.
root@DESKTOP-5K616C3:~/rotation#
```

Q2:- Make the Script Executable:

bash

Copy code

chmod +x add_user.sh



```
root@DESKTOP-5K616C3:~/rotation# chmod +x add_user.sh
root@DESKTOP-5K616C3:~/rotation#
```

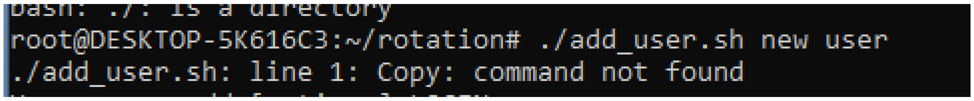
3.Run the Script:

- Run the script to add a new user:

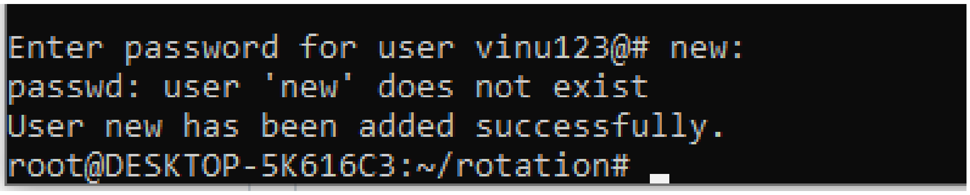
bash

Copy code

./add_user.sh newuser



```
bash: ./: is a directory
root@DESKTOP-5K616C3:~/rotation# ./add_user.sh new user
./add_user.sh: line 1: Copy: command not found
```



```
Enter password for user vinu123@# new:
passwd: user 'new' does not exist
User new has been added successfully.
root@DESKTOP-5K616C3:~/rotation# _
```

4. Create a Script to Remove Users:
   - Create a script called remove_user.sh to automate removing a user.

bash

Copy code

nano remove_user.sh

- Add the following content:

bash

Copy code

#!/bin/bash

#Check if username is provided

if [-z "$1"]; then

echo "Error: Please provide a username."

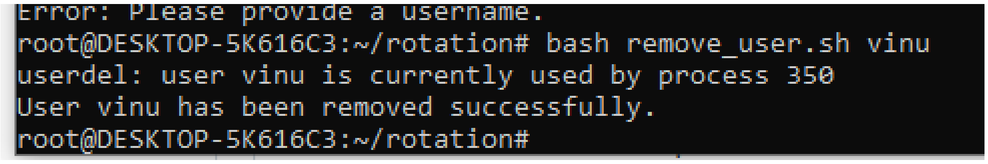
exit 1

fi

#Remove user from the system

sudo userdel $1

echo "User $1 has been removed successfully."



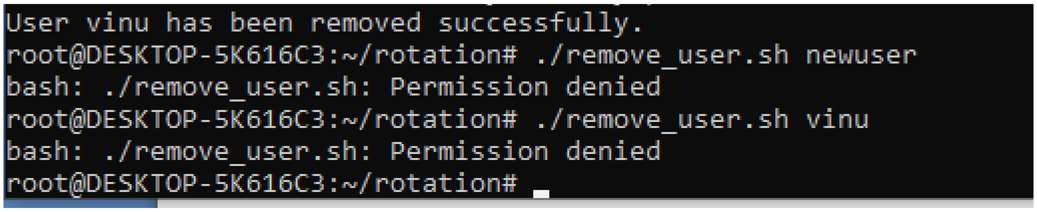
```
Error: Please provide a username.
root@DESKTOP-5K616C3:~/rotation# bash remove_user.sh vinu
userdel: user vinu is currently used by process 350
User vinu has been removed successfully.
root@DESKTOP-5K616C3:~/rotation#
```

5.Run the Script to Remove a User:

bash

Copy code

./remove_user.sh newuser



```
User vinu has been removed successfully.
root@DESKTOP-5K616C3:~/rotation# ./remove_user.sh newuser
bash: ./remove_user.sh: Permission denied
root@DESKTOP-5K616C3:~/rotation# ./remove_user.sh vinu
bash: ./remove_user.sh: Permission denied
root@DESKTOP-5K616C3:~/rotation#
```

Lab 6: Automating File Cleanup

Objective:

- Automate the deletion of old files in a directory to free up disk space.

Tasks:

1.Create a Cleanup Script:

o Create a script called file_cleanup.sh to remove files older than 30 days.

bash

Copy code

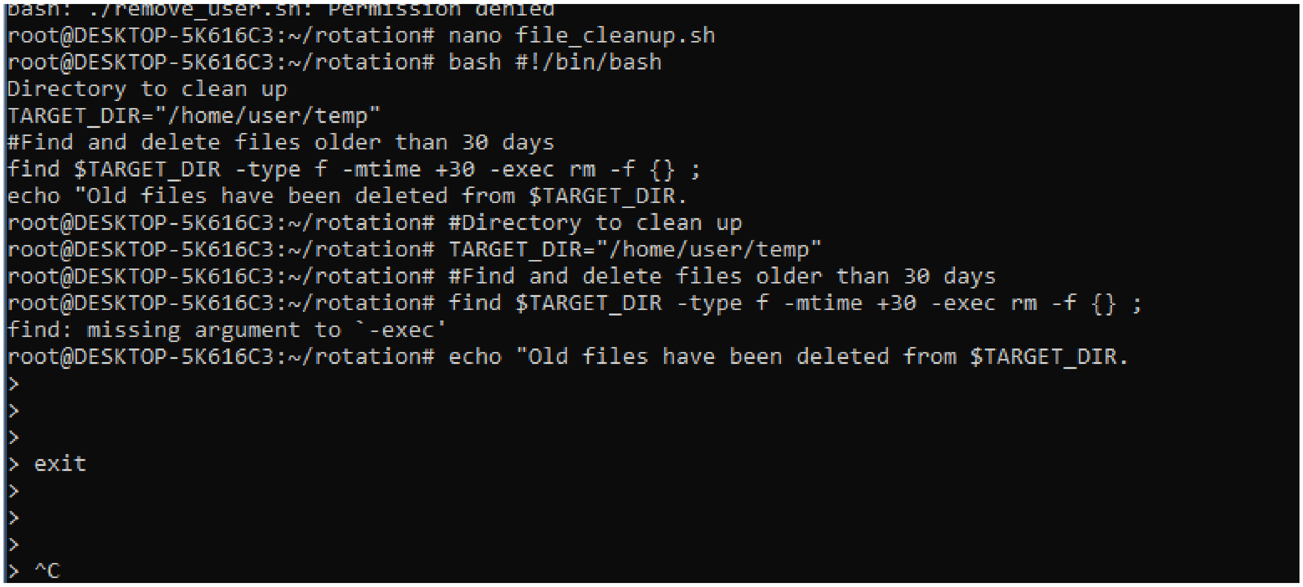
```
nano file_cleanup.sh
```

o Add the following content:

bash

Copy code

```
#!/bin/bash

#Directory to clean up

TARGET_DIR="/home/user/temp"

#Find and delete files older than 30 days

find $TARGET_DIR -type f -mtime +30 -exec rm -f {} ;

echo "Old files have been deleted from $TARGET_DIR.
```



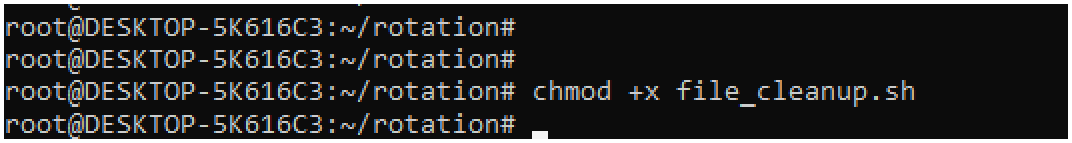
```
bash: ./remove_user.sh: Permission denied
root@DESKTOP-5K616C3:~/rotation# nano file_cleanup.sh
root@DESKTOP-5K616C3:~/rotation# bash #!/bin/bash
Directory to clean up
TARGET_DIR="/home/user/temp"
#Find and delete files older than 30 days
find $TARGET_DIR -type f -mtime +30 -exec rm -f {} ;
echo "Old files have been deleted from $TARGET_DIR.
root@DESKTOP-5K616C3:~/rotation# #Directory to clean up
root@DESKTOP-5K616C3:~/rotation# TARGET_DIR="/home/user/temp"
root@DESKTOP-5K616C3:~/rotation# #Find and delete files older than 30 days
root@DESKTOP-5K616C3:~/rotation# find $TARGET_DIR -type f -mtime +30 -exec rm -f {} ;
find: missing argument to `-exec'
root@DESKTOP-5K616C3:~/rotation# echo "Old files have been deleted from $TARGET_DIR.
>
>
>
> exit
>
>
>
> ^C
```

Q2 :-Make the Script Executable:

bash

Copy code

chmod +x file_cleanup.sh



```
root@DESKTOP-5K616C3:~/rotation#
root@DESKTOP-5K616C3:~/rotation#
root@DESKTOP-5K616C3:~/rotation# chmod +x file_cleanup.sh
root@DESKTOP-5K616C3:~/rotation#
```

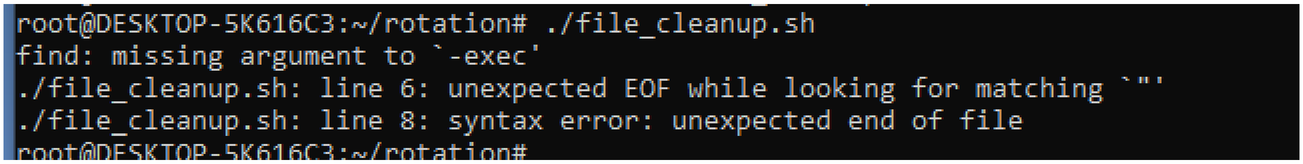
Q3 :-Run the Cleanup Script:

- o Run the script to clean up old files:

bash

Copy code

./file_cleanup.sh



```
root@DESKTOP-5K616C3:~/rotation# ./file_cleanup.sh
find: missing argument to `-exec'
./file_cleanup.sh: line 6: unexpected EOF while looking for matching `"'
./file_cleanup.sh: line 8: syntax error: unexpected end of file
root@DESKTOP-5K616C3:~/rotation#
```

4. Schedule the Cleanup Script Using Cron:

- Open the crontab file to schedule the cleanup script to run weekly.

bash

Copy code

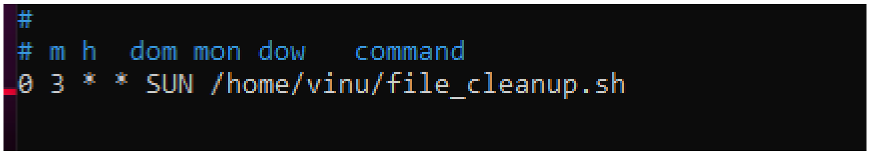
crontab -e

- Add the following cron job:

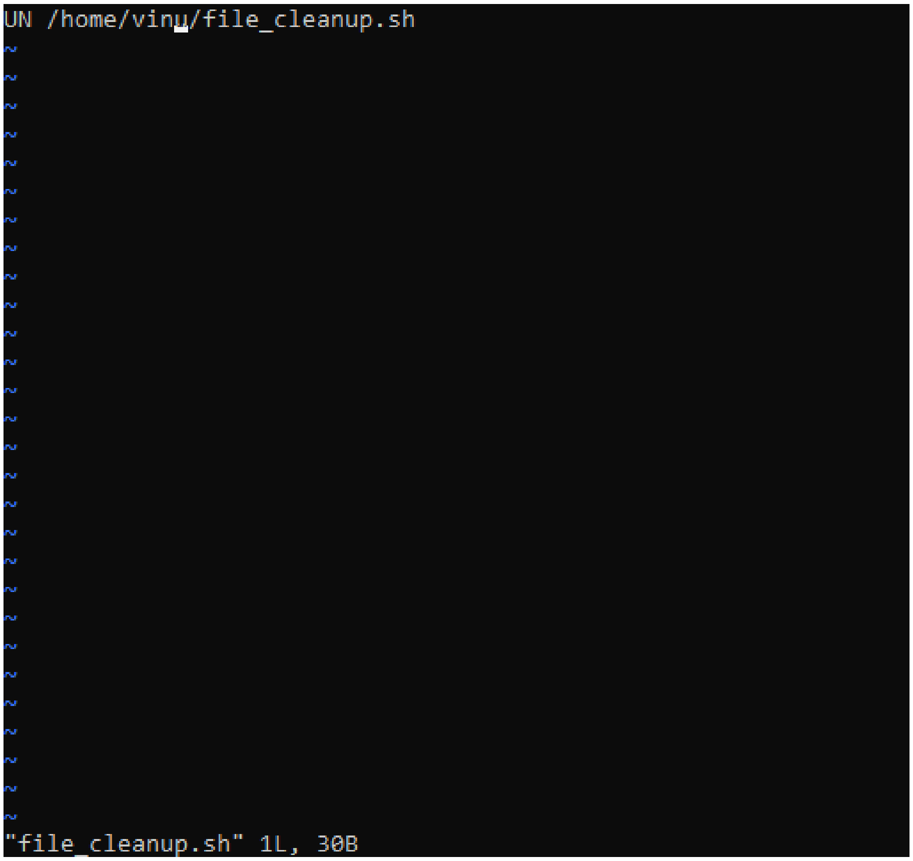
bash

Copy code

0 3 * * SUN /path/to/file_cleanup.sh



```
#
# m h  dom mon dow   command
0 3 * * SUN /home/vinu/file_cleanup.sh
```



```
UN /home/vinu/file_cleanup.sh
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"file_cleanup.sh" 1L, 30B
```