SHELL SCRIPTING

"You're told to monitor a service (like Apache) every 5 minutes and log its status to a file. You are not allowed to install any third-party tools. How would you automate this?"

setup the check_apache_scripts.sh

- #!/bin/bash

```
LOGFILE="/var/log/apache_status.log
TIMESTAMP=$(date '+%Y-%m-%d %H:%M:%S)
SERVICE='httpd'

echo "[TIMESTAMP]" check apache status >> $LOGFILE

if systemctl is-active --quite $SERVICE; then
        echo "[TIMESTAMP] apache is running" >> $LOGFILE
else
  echo "[TIMESTAMP] apache has been stopped" >> $LOGFILE
fi
```

```
chmod +x /home/user/scripts/check_apache_scripts.sh

crontab -e

5 * * * * * /home/user/scripts/check_apache_scripts
```

Create an automated email alert system for the apache server ?

```
#!/bin/bash

TIMESTAMP=$(date '+%y-%m-%d %H-%M-%S')
```

```
EMAIL="exampleemail.com"

LOGFILE="/var/log/apache_status.log

SERVICE='httpd'


echo "[TIMESTAMP] CHECKING APACHE STATUS " >> $LOGFILE


if systemctl is-active --quite $SERVICE; then

        echo "[TIMESTAMP] the service is running " >> $LOGFILE

else

        echo "[TIMESTAMP] the service is inactive" >> $LOGFILE

        ech0 "ALERT VIA EMAIL , SERVICE $SERVICE is down $(hostname) at $TIMESTAMP"
| mail -s "apache server down at $(hostname)" $EMAIL

FI
```

Write a script that checks if a file called `/etc/passwd` exists and is readable. Log the result with a timestamp.

```
#!/bin/bash

clear


PATH='/etc/passwd'

TIMESTAMP=$(date '%y-%m-%d %H-%M-%S')


if [-e $PATH]; then

        echo '[TIMESTAMP] the file exists and its readable : $PATH'

else

        echo '[TIMESTAMP] the file is missing and its not readablle : $PATH'

fi
```

Create a script that runs `df -h` and stores the output in a file named `disk_report_<date>.log` in `/var/log/custom/` .

```
#!/bin/bash
clear

mkdir /var/log/custom/

DATE=$(date '%y-%m-%d %H-%M-%S')

LOGFILE="/var/log/custom/disk_report_$DATE.log"

echo "Running the command and storing here " >> $LOGFILE
df -h >> $LOGFILE

echo " Storing the disk report $LOGFILE"
```

Write a script that checks if `https://zoho.com` is reachable (using `curl`) and logs `UP` or `DOWN` with timestamp.

```
#!/bin/bash
clear
LOGFILE="/var/log/zoho_curlcheck.log

mkdir -p $(dirname $LOGFILE)

TIMESTAMP=$(date %y-%m-%d %H-%M-%S)

curl -s --head --request GET https://zoho.com | grep "200 ok" >> /dev/null

if[$? -eq 0]; then
        echo "[TIMESTAMP] zoho is running " >> $LOGFILE
else
```

```
        echo "[TIMESTAMP] zoho is not running " >> $LOGFILE
    fi
```

Ask the user for a directory path and output how many files it contains.

```
    #!/bin/bash
    clear

    read -p "enter the path" DIR

    TIMESTAMP=$(date %y-%m-%d %H-%M-%S)

    if[-d "DIR"]; then
        FILECOUNT=$(ls -p $DIR | grep -v / | wc -l)
        echo "[TIMESTAMP] the number if files in '$DIR' is %FILECOUNT"
    else
        echo "[TIMESTAMP] the file doesnt '$DIR' exist "
```

Write a script that sends a warning message (use `wall`) 1 minute before a scheduled reboot using `at`.

```
    #!/bin/bash
    clear

    read -p "enter the reboot time like in minutes " REBOOT_TIME

    WARNING_TIME=$(echo "$REBOOT_TIME" | awk '{print $1 , $2 , "+" , $4-1 , $5}')

    echo "wall 'system is going to reboot , please save your work' " | $WARNING_TIME

    echo 'shutdown the system ' | at $REBOOT_TIME

    echo " reboot time schedule at '$REBOOT_TIME' with an 1 min warning "
```

Write a script to extract the last 5 login records of the current user and save them to `~/last_logins.txt`.

```
#!/bin/bash

clear


CURRENT_USR = $(whoami)


OUTPUT_FILE = "$HOME/last_logins.txt"


last "$CURRENT_USR" | head -n 5 > $OUTFILE_FILE


echo " the last 5 logins are saved in $OUTFILE_FILE"
```

Check if the script is being run as root. If not, exit with a message: "Permission Denied – Run as root only."

```
#!/bin/bash

clear


if ["EUID" -ne 0 ]; then

        echo "permissions denied - run as root "

        exit 1

else

        echo "permissions is approved proceed with execution "
```

Ask the user to enter a port number and use `netstat` or `ss` to check if it's listening.

```
#!/bin/bash

clear

read -p " enter the port number to check " PORT_CHECK
```

```
if ! [[ "PORT_CHECK" =~ ^[0-9]+$]]; then
        echo " invalid only enter the numbers"
        exit 1
fi

if ss -tuln |  grep-q ':$PORT_CHECK"; then
        echo " port number $PORT_CHECK is listening"
else
        echo " port $PORT_CHECK is NOT listen"
fi

if netstat -tuln | grep -q ":$PORT_CHECK"; then
        echo "port number $PORT_CHECK is listening"
else
        echo "port is not listening $PORT_CHECK"
fi
```

Create a script that zips the contents of `/etc` into `/tmp/etc_backup_<date>.tar.gz`

```
#!/bin/bash
clear

DATE==$(date %y-%m-%d %H-%M-%S)
BACKUP_FILE="/tmp/etc_backup_$DATE.tar.gz"

tar -cvf "$BACKUP_FILE" /etc

if[$? -eq 0 ]; then
        echo " backup created successfully at : $BACKUP_FILE"
else
        echo "backup is not created"
```

```
fi
```

Monitor the Apache service (`httpd`). If it is inactive, restart it and log the action with date/time.

```
#!/bin/bash
clear

LOGFILE="/var/log/apache_monitor.log
TIMESTAMP=$(date %y-%m-%d %H-%M-%S)

if systemctl is-active --quite httpd; then
        echo "[TIMESTAMP] apache is running " >> LOGFILE
else
        echo "[TIMESTAMP] apache is not running" >> LOGFILE

        systemctl restart httpd

        if systemctl is-active --quite httpd; then
                echo "[TIMESTAMP] apache restart success " >> LOGFILE
        else
                echo "[TIMESTAMP] apache didnt restart" >> LOGFILE
        fi
fi
```

Accept a list of services (httpd, sshd, cron) and check if each is running. If not, restart and log the action with a timestamp.

```
#!/bin/bash
clear

LOGFILE="/var/log/servicelogs.log"
TIMESTAMP=$(date %y-%m-%d %H-%M-%S)
```

```
if systemctl is-active --quiet httpd ,
systemctl is-active --quiet sshd ,
systemctl is-active --quiet cron; then
        echo "[TIMESTAMP] the services are running" >> "$LOGFILE"
else
        echo "[TIMESTAMP] the services are not running" >> "$LOGFILE"

        systemctl restart httpd
        systemctl restart sshd
        systemctl restart cron

        if system is-active --quiet httpd ,
        systemctl is-active --quiet sshd ,
        systemctl is-active --quiet cron ; then
                echo "[TIMESTAMP] the services have been restarted" >> "$LOGFILE"
        else
                echo "[TIMESTAMP] the services have not been restarted " >> "$LOGFILE"
        fi
fi
```

BY USING FOR LOOP:

```
#!/bin/bash
clear

LOGFILE="/var/log/servicelogs.log"
TIMESTAMP=$(date %y-%m-%d %H-%M-%S)
SERVICE=$(httpd , sshd , cron )

for SERVICE in "${SERVICE[@]}"; do
```

```
                if systemctl is-active --quiet "$SERVICE"; then
                        echo "[TIMESTAMP] the service is running " >> "$LOGFILE"
                else
                        echo "[TIMESTAMP] the service is not running" >> "$LOGFILE"

                systemctl restart "$SERVICE"

                if systemctl is-active --quiet "$SERVICE"; then
                        echo "[TIMESTAMP] the service is restarted sucessfully " >> "$LOGFILE"
                else
                        echo "[TIMESTAMP] the service is not restarted" >> "$LOGFILE"
                fi
        fi
done
```