Task 2

by: Ziad Ahmed Ibrahim

In this task I was required to create a service that runs daily at 2 AM and backup a directory into another directory with all info about it. it should start on boot

2 Methods I found:

1. Using the cronjob to automate a script without systemd
2. Using service and a timer for the service using systemd

Why systemd is better

1. they are services with timer events and are carefully logged in systemd journal
2. easier debugging
3. Can be enabled and disabled easily
4. Status can be checked easily
5. Delays can be set between execution

How I used it

1) The backup.service file

```
[Unit]
Description=Daily Backup Service

[Service]
Type=oneshot
ExecStart=/home/msi/data_backup.sh

[Install]
WantedBy=multi-user.target
```

Description:

- This .service file is created in etc/systemd/system directory
- We pass to the service the path of the .sh file to be executed when the service is starts(trigger happens)
- `Type=oneshot` part is not mandatory and can be removed, this line ensures that other services depending on this service will start after this one is finished
- ExecStart part is where the path of the script is provided

The backup.timer file

```
[Unit]
Description= timer for backup

[Timer]
OnCalendar=*-*-* 02:00:00
Unit=backup.service
Persistent=True

[Install]
WantedBy=multi-user.target
```

Description

- the OnCalendar part is the part where we tell the timer to trigger event for the service everyday at 2 Am
- Unit is the unit to run after the trigger
- Persistent part means that in case of reboot or restart the process will continue after booting process

Script made

```
#!/bin/bash
timestamp=$(date +%Y-%m-%d_%H-%M-%S)    #this line takes the time stamp at the running time of this script and stores it in this variable
source="/home/msi/Important"    #a variable that holds the source file that we need to backup
backup_dir="/home/msi/backup_folder" # a variable that hold the directory of the back up destination
backup_dest="$backup_dir/backup_$timestamp" #in this line i combine both the directory and the file of the back up with timestamp in it
mkdir -p "$backup_dest" #create a directory with the name and destination in the variable above
rsync -a "$source" "$backup_dest"  #using rsync we take copy of the directory stored in the variable "Source" in to "backup_dest"
#in next line we List the files then sort them in descending order, then get the last 6 elements with tail command, loop through them until you reach the last one, after the loop ends you delete the last file and keep the other 5 which will be the latest versions
ls -1 "$backup_dir" | sort -r | tail -n +6 | while read -r file; do
        rm -r "$backup_dir/$file"
done
```

- Ensured timestamps, permissions are all passed to the backup
- rsync installed

- using minutes and hours won't be mandatory in our script as we know it will execute everyday at 2 AM but to make sure the service runs correctly I added it

How I tested my service

1) Testing the timer

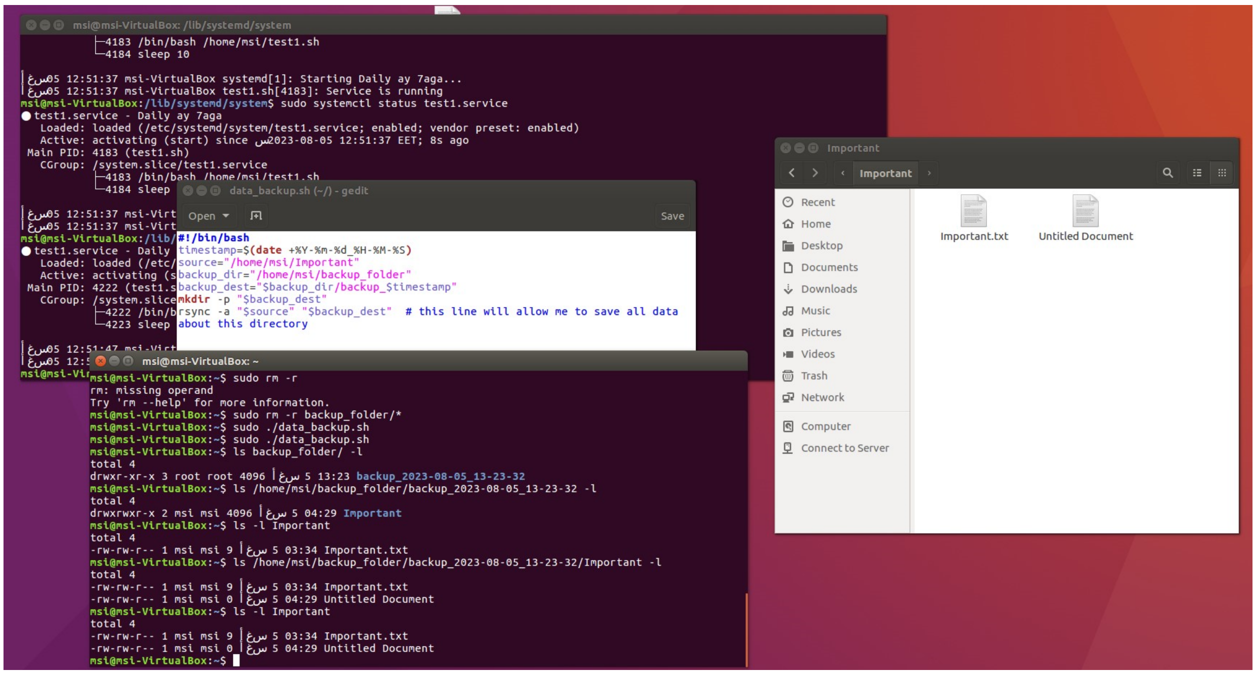
- Instead of triggering everyday at 2 AM, the service is to be triggered every 5 seconds
- To activate a service every 5 Seconds add this instead of the OnCalendar

```
[Timer]
        OnActiveSec=5
        OnUnitActiveSec=5
```

- every 5 seconds a script is called that will echo anything to another file and we check it every 5 seconds

2) Testing the script

- After I made sure the timer works and the service is called correctly
- I made wrote a script that will backup a directory in another directory and tested it by executing it manually with terminal
- another approach is given to allow the timestamp signature in the name of the folder of the backed up folder
- an algorithm is implemented to only keep the latest 5 and delete the oldest one from the 6
- My script was tested by executing the back up service every 5 seconds
- after I changed the timer back to trigger every day at 2 AM



```
         ├─4183 /bin/bash /home/msi/test1.sh
         └─4184 sleep 10

أغس05 12:51:37 msi-VirtualBox systemd[1]: Starting Daily ay 7aga...
أغس05 12:51:37 msi-VirtualBox test1.sh[4183]: Service is running
msi@msi-VirtualBox:/lib/systemd/system$ sudo systemctl status test1.service
● test1.service - Daily ay 7aga
   Loaded: loaded (/etc/systemd/system/test1.service; enabled; vendor preset: enabled)
   Active: activating (start) since س2023-08-05 12:51:37 EET; 8s ago
 Main PID: 4183 (test1.sh)
   CGroup: /system.slice/test1.service
           ├─4183 /bin/bash /home/msi/test1.sh
           └─4184 sleep

أغس05 12:51:37 msi-Virt
أغس05 12:51:37 msi-Virt
msi@msi-VirtualBox:/lib/
● test1.service - Daily
   Loaded: loaded (/etc/
   Active: activating (s
 Main PID: 4222 (test1.s
   CGroup: /system.slice
           ├─4222 /bin/b
           └─4223 sleep

أغس05 12:51:47 msi-Virt
أغس05 12:5
msi@msi-Vir
```

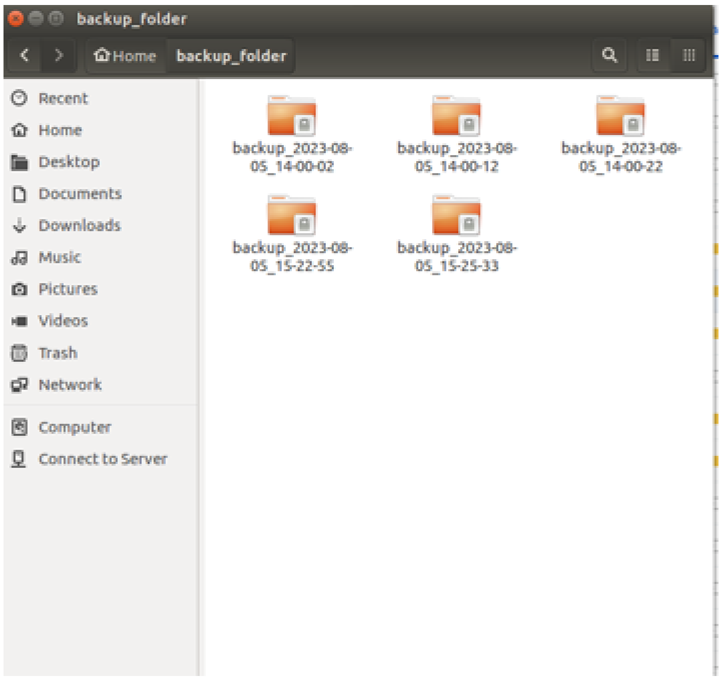
```
#!/bin/bash
timestamp=$(date +%Y-%m-%d_%H-%M-%S)
source="/home/msi/Important"
backup_dir="/home/msi/backup_folder"
backup_dest="$backup_dir/backup_$timestamp"
mkdir -p "$backup_dest"
rsync -a "$source" "$backup_dest"  # this line will allow me to save all data
about this directory
```

```
msi@msi-VirtualBox:~$ sudo rm -r
rm: missing operand
Try 'rm --help' for more information.
msi@msi-VirtualBox:~$ sudo rm -r backup_folder/*
msi@msi-VirtualBox:~$ sudo ./data_backup.sh
msi@msi-VirtualBox:~$ sudo ./data_backup.sh
msi@msi-VirtualBox:~$ ls backup_folder/ -l
total 4
drwxr-xr-x 3 root root 4096 أغس 5 13:23 backup_2023-08-05_13-23-32
msi@msi-VirtualBox:~$ ls /home/msi/backup_folder/backup_2023-08-05_13-23-32 -l
total 4
drwxrwxr-x 2 msi msi 4096 أغس 5 04:29 Important
msi@msi-VirtualBox:~$ ls -l Important
total 4
-rw-rw-r-- 1 msi msi 9 أغس 5 03:34 Important.txt
msi@msi-VirtualBox:~$ ls /home/msi/backup_folder/backup_2023-08-05_13-23-32/Important -l
total 4
-rw-rw-r-- 1 msi msi 9 أغس 5 03:34 Important.txt
-rw-rw-r-- 1 msi msi 0 أغس 5 04:29 Untitled Document
msi@msi-VirtualBox:~$ ls -l Important
total 4
-rw-rw-r-- 1 msi msi 9 أغس 5 03:34 Important.txt
-rw-rw-r-- 1 msi msi 0 أغس 5 04:29 Untitled Document
msi@msi-VirtualBox:~$ 
```

This screenshot was taken during testing the script before fully completing it

Conclusion:

In essence, we conclude that linux offers a really helpful environment to create scripts that helps developers do tasks they want and test them effectively. Linux also offers a very easy to use scripting tools, and above all Linux is an open source in which many developers posting on forums tutorials and guides on how to use Linux terminal and how to write good scripts.


backup_folder
Home
backup_folder
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
Connect to Server
backup_2023-08-05_14-00-02
backup_2023-08-05_14-00-12
backup_2023-08-05_14-00-22
backup_2023-08-05_15-22-55
backup_2023-08-05_15-25-33

Final result

Sources

press on the hyper link

rsync Guide

Cron guide

Creating new systemd service

Cron vs systemd

Services commands guide