Linux
– Bash Scripting
Assignment

Done By : Jafar Farwaneh

Table Of Contents

1.0 Introduction

- In this Assignment, I was asked to write two Bash scripts, that automate common Linux system administration tasks, to exhibit advanced skills in using common Linux commands and shell scripting.
- The first script is a Backup script, that automates backing up user-specified directories, and the second one is a system health check script that generates a health check report for the user's system.

2.0 Backup Script

2.1 Overview

- This script offers backup service for user-specified directories.
- I implemented it to be interactive and user friendly, it also can be executed without any user interface using the options specified for it.
- In addition to the normal (Full) backup, it supports Incremental backup, which is a backup that only stores the changed file since the last full or incremental backup, which improves performance a lot.
- Auto backups are offered; daily, weekly, or monthly full backups can be automated through an option in the script.
- This script initializes an organized directories hierarchy in the destination directory of the backup, which enhances the user experience in managing the backups.

2.2 Functionality Breakdown

2.2.1 Interactive user interface

- Implemented a user friendly and interactive user interface, that reads multiple arguments and details that are crucial to run the backup.
- It also allows the user to know the result of the backup and any errors encountered.

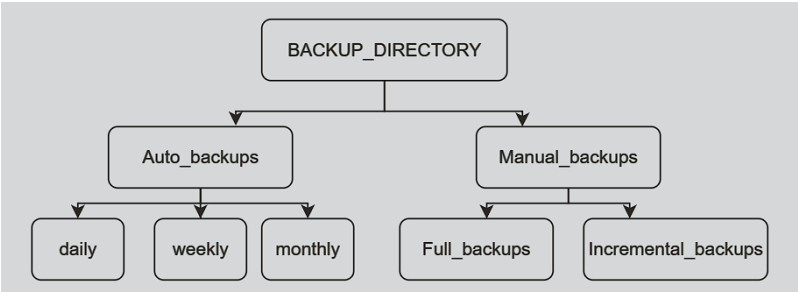
```
jafar@jafar:~/Desktop$ ./backup_script.sh
-> Enter directories to back up, separated by spaces: /home/jafar/Desktop/UNO
-> Enter destination directory( full path from /home )
   If performing an incremental backup , it must contain the last full backup in its sub directories : /home/jafar/Desktop/backups
-> Choose how you want to back up:

  1. Full backup (creates a complete copy of all selected directories)
  2. Incremental backup (only backs up files changed since the last backup)

Which option do you want? (1/2): 1
Welcome to the Backup Utility!
Creating a full backup, please be patient this may take some time...
Checking and initializing the backup directory structure , this requires creating new directories in the destination directory if not created before
**********************************************************
Started a manual full backup for /home/jafar/Desktop/UNO
Creating a new directory with new timestamp for UNO
Size of /home/jafar/Desktop/UNO before compression : 27M
Backing up /home/jafar/Desktop/UNO to /home/jafar/Desktop/backups/Manual_backups/Full_backups/UNO/UNO-2024.01.14-21:40:45/backup-UNO-2024.01.14-21:4
0:45.tar.gz...
Backup of /home/jafar/Desktop/UNO completed successfully!
Size of backup files : 24M
**********************************************************
Backup finished!
```

2.2.2 Directory hierarchy initialization

- As mentioned in the Introduction, and before anything, the script creates a well-defined hierarchy to store the backups in, this is done by the function `initialize_dirs` , the hierarchy is as shown in the below figure .



2.2.3 Full backup

- A normal full backup is performed by this script using the built in package (`tar`) that offers archiving and compressing, All the directories specified get backed up to the destination directory, after creating a new name-matching directory for each one of them there, to further

enhance the structure of the backups and to group different backups of the same directory together.

- To show the effect of compressing in this backup type and all other types, the size of the files before compressing is recorded, printed to the user, and logged, the same goes for the size of the backup archive after compressing.

2.2.4 Incremental backup

- An incremental backup is a backup that includes only the changes after the last full or incremental backup, this leads to an optimized performance in both space and time, it's also implemented in the script using the (tar) package specifically by using the (--listed-incremental) option it provides.

- For that to work, a snapshot is required, the purpose of this file is to help determine which files have been changed, added, or deleted since the last backup, so that the next incremental backup will contain only modified files, this file is created in the script for any backup whether it's full or incremental, manual, or auto.

- The script automatically searches for the snapshot file of the last backup of the directory in the backup directory and fetches it and copies it to the newly created directory of the current incremental backup, it then reads the contents of the snapshot file and backs up the changed files, then overwrites the snapshot to prepare it for potential future incremental backups.

2.2.5 Auto backups

- To further increase the automation of backups, I included an option in the script that enables automatic backups periodically (daily, weekly, or monthly), this is done by creating a (cronjob) which is a defined task to run in a given time period, using the (crontab) command.

2.3 Optimizations

2.3.1 Suitable interfaces for all users

- By supporting both user interface and script options, the script accommodates all types of users, whether they prefer an organized user interface, or a quick command with options.

```
jafar@jafar:~/Desktop$ ./backup_script.sh
-> Enter directories to back up, separated by spaces: /home/jafar/Desktop/UNO
-> Enter destination directory( full path from /home )
   If performing an incremental backup , it must contain the last full backup in its sub directories : /home/jafar/Desktop/backups
-> Choose how you want to back up:

  1. Full backup (creates a complete copy of all selected directories)
  2. Incremental backup (only backs up files changed since the last backup)

Which option do you want? (1/2): 1
Welcome to the Backup Utility!
Creating a full backup, please be patient this may take some time...
```

```
jafar@jafar:~/Desktop$ ./backup_script.sh -m 1 -s /home/jafar/Desktop/UNO -d /home/jafar/Desktop/
backups
Welcome to the Backup Utility!
Creating a full backup, please be patient this may take some time...
```

2.3.2 Providing options for quick execution

- This complements the previous point, the options the script offers are shown in the help interface:

```
jafar@jafar:~/Desktop$ ./backup_script.sh -h
Backup Utility Help Menu
Options:
  -h, usage : -h                          Show this help message and exit
  -m, usage : -m [1,2]                    Choose backup mode (1 for full, 2 for incremental)
  -s, usage : -s [DIR1 DIR2 ...]          Set the source directories to back up
  -d, usage : -d PATH                     Set the destination directory to back up to
  -a, usage : -auto [daily,weekly,monthly]    Enable automatic backups (daily, weekly, or monthly all at midnight(00:00))
```

2.3.3 User input validation

- Input validation was ensured in various input reading operations, to ensure only properly formed data is entering the script.

2.3.4 Auto deletion of old auto backups

- To optimize the memory space consumption of the auto backups, a (cronjob) that deletes backups after a set period of time (7 days for daily backups, 31 for weekly backups and 365 for monthly backups) is added in addition to the backup cronjob, this gives the backups a kind of self-organization to not bloat the memory with old and unnecessary backups.

2.3.5 Incremental backups

- Optimization of both space and time consumed for backups is the ultimate reason for performing such a backup, hence its inclusion in the script is a big optimization.

2.3.6 Logging

- To record the events that occur during the backup and any issues encountered, they are written to 2 log files, the first is a general log for all backups in that directory and for all types of backups, the second one is a log file specific for the current backup, which is stored with the backup archive and its snapshot.

2.3.7 Timestamped names

- Every backup has a unique name that resembles the name of the source directory followed by a timestamp that is generated using the (date) command.

2.4 Insights from performance analyses

2.4.1 Modularity:

- The script is structured with functions, enhancing modularity and code organization. Each function serves a specific purpose, making the script more readable and maintainable.

2.4.2 User Interaction:

- The script engages with the user through informative prompts and help messages. This improves the user experience and ensures that users with basic Linux knowledge can understand and interact with the script effectively.

2.4.3 Input Validation:

- The script includes input validation to ensure the correctness of user-provided options. It handles scenarios where users might enter invalid directories or modes and prompts them until valid inputs are provided.

2.4.4 Automatic Backup Configuration:

- The `check_auto_config` function efficiently configures automatic backups based on user preferences. It uses cronjobs to schedule backups at specified intervals (daily, weekly, or monthly), demonstrating a well-integrated and automated solution.

2.4.5 Logging:

- The script generates detailed log files, providing a comprehensive record of backup activities. Each log entry includes timestamps, contributing to better traceability and understanding of the backup process.

2.4.6 Snapshot Handling:

- The `fetch_last_snapshot_file` function retrieves the snapshot file of the last backup in incremental mode. This ensures the integrity of

incremental backups by using the most recent full backup as a reference.

2.4.7 Efficient Directory Initialization:

- The `initialize_dirs` function creates the necessary directory structure, ensuring that required directories are present before performing backups. This is crucial for the success of the backup process.

2.4.8 Backup Process:

- The script employs a loop to iterate through specified directories, creating backups based on user preferences. It distinguishes between full and incremental backups, creating appropriate directory structures and handling log files accordingly.

2.4.9 Error Handling:

- The script includes error handling mechanisms, providing descriptive error messages when directories are not found or when users provide invalid inputs. This enhances the script's robustness and user-friendliness.

3.0 System Health Check Script

3.1 Overview

- This script runs health checks for multiple system components (hardware and software) and generates a user-friendly report with the results of these checks.

- The checks include:
    1. General system information (hostname, kernel version, uptime, last reboot time)
    2. Disk usage
    3. Memory usage
    4. Running processes (top 10 by CPU and RAM usage)
    5. Active system services
    6. Recent system updates and installations
    7. Recommendations based on findings.
- The report is saved as "system_health_report.txt" in the current directory.

3.2 Functionality Breakdown

3.2.1 System Information:

- Displays essential system information such as hostname, kernel version, uptime, and last reboot time.

```
================================================================================
                        Health Check Report
================================================================================

Hostname         : jafar
Kernel Version   : 6.5.0-14-generic
Uptime           :  7:20
Last Reboot Time : 2024-01-14 21:16

```

3.2.2 Disk Usage Check:

- Provides a summary of disk usage for all partitions.

- Warns if any partition exceeds 90% usage and offers a recommendation to address the issue.

```
10 ================================================================================
11                             Disk Usage
12 ================================================================================
13         ( <90% healthy , >=90% Needs Caution , >=95% Unhealthy )
14
15 Filesystem      Size  Used Avail Use% Mounted on
16 tmpfs           266M  1.5M  265M   1% /run
17 /dev/sda3        24G   15G  8.7G  62% /
18 tmpfs           1.3G     0  1.3G   0% /dev/shm
19 tmpfs           5.0M  4.0K  5.0M   1% /run/lock
20 /dev/sda2       512M  6.1M  506M   2% /boot/efi
21 tmpfs           266M  120K  266M   1% /run/user/1000
22
23 [INFO] Disk usage is healthy.
24
```

3.2.3 Memory Usage Check:

- Presents an overview of memory usage, including total, used, and free memory.
- Warns if memory usage exceeds 80% and provides insights into the system's memory health.

```
25 ================================================================================
26                            Memory Usage
27 ================================================================================
28
29                total        used        free      shared  buff/cache   available
30 Mem:           2.6Gi       917Mi        98Mi        53Mi       1.6Gi       1.5Gi
31 Swap:          2.6Gi       0.0Ki       2.6Gi
32
33 [INFO] Memory usage is healthy.
34
```

3.2.4 Top Processes Check:

- Lists the top 10 processes consuming the most RAM and CPU resources.
- Helps identify resource-intensive processes that may impact system performance.

```
35 ================================================================
36                     Top 10 Processes by RAM Usage
37 ================================================================
38
39 USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
40 jafar       2851  2.5 14.6 5044152 398112 ?      Ssl  Jan14  11:04 /usr/bin/gnome-shell
41 jafar       3240  0.0  2.9 529176 81164 ?        Ssl  Jan14   0:00 /usr/libexec/gsd-xsettings
42 jafar       3644  0.4  2.9 654140 80716 ?        Sl   Jan14   1:46 /usr/bin/gedit --gapplication-service
43 jafar       3306  0.0  2.9 2883560 80444 ?       Sl   Jan14   0:02 gjs /usr/share/gnome-shell/extensions/din
   ding@rastersoft.com -M 0 -D 0:0:1920:969:1:27:0:70:0:0
44 jafar       3513  0.3  2.9 938544 79608 ?        Sl   Jan14   1:23 /usr/bin/nautilus --gapplication-service
45 jafar       3124  0.0  2.4 819544 67048 ?        Sl   Jan14   0:00 /usr/libexec/evolution-data-server/evolut
46 jafar       3204  0.0  2.4 205584 65636 ?        S    Jan14   0:10 /usr/bin/Xwayland :0 -rootless -noreset -
   5 -displayfd 6 -initfd 7
47 jafar       3436  0.0  2.0 555972 54496 ?        Ssl  Jan14   0:07 /usr/libexec/gnome-terminal-server
48 root         202  0.0  1.8  94148 49024 ?        S<s  Jan14   0:00 /lib/systemd/systemd-journald
49 jafar       3123  0.0  1.5 350196 42652 ?        Sl   Jan14   0:04 /usr/libexec/ibus-extension-gtk3
50
```

```
51 ================================================================
52                     Top 10 Processes by CPU Usage
53 ================================================================
54
55 USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
56 jafar       2851  2.5 14.6 5051832 398112 ?      Ssl  Jan14  11:04 /usr/bin/gnome-shell
57 jafar       3644  0.4  2.9 654140 80716 ?        Sl   Jan14   1:46 /usr/bin/gedit --gapplication-service
58 jafar       3513  0.3  2.9 938544 79608 ?        Sl   Jan14   1:23 /usr/bin/nautilus --gapplication-service
59 root           1  0.0  0.4 166736 11708 ?        Ss   Jan14   0:01 /sbin/init splash
60 root           2  0.0  0.0      0     0 ?        S    Jan14   0:00 [kthreadd]
61 root           3  0.0  0.0      0     0 ?        I<   Jan14   0:00 [rcu_gp]
62 root           4  0.0  0.0      0     0 ?        I<   Jan14   0:00 [rcu_par_gp]
63 root           5  0.0  0.0      0     0 ?        I<   Jan14   0:00 [slub_flushwq]
64 root           6  0.0  0.0      0     0 ?        I<   Jan14   0:00 [netns]
65 root          11  0.0  0.0      0     0 ?        I<   Jan14   0:00 [mm_percpu_wq]
66
```

3.2.5 Running Services Check:

- Displays a list of currently running services using the (systemctl)command.
- Provides an overview of active services on the system.

```
67 ======================================================================
68                       Running Services
69 ======================================================================
70
71   UNIT                                                    LOAD   ACTIVE SUB       DESCRIPTION
72   proc-sys-fs-binfmt_misc.automount                       loaded active running   Arbitrary Executable
73   acpid.path                                              loaded active running   ACPI Events Check
74   cups.path                                               loaded active running   CUPS Scheduler
75   init.scope                                              loaded active running   System and Service Ma
76   session-2.scope                                         loaded active running   Session 2 of User jaf
77   accounts-daemon.service                                 loaded active running   Accounts Service
78   acpid.service                                           loaded active running   ACPI event daemon
79   avahi-daemon.service                                    loaded active running   Avahi mDNS/DNS-SD Sta
80   colord.service                                          loaded active running   Manage, Install and G
81   cron.service                                            loaded active running   Regular background pr
```

3.2.6 System Updates and last installations Check:

- Highlights recent system installation commands from the APT history log.
- Reviews recent changes and updates from the dpkg.log.
- Warns if the system hasn't been updated in the last 14 days and offers a recommendation to perform updates.

```
121 ======================================================================
122                  Recent System Installation Commands
123 ======================================================================
124
125 Commandline: apt-get install -y borgbackup
126 Commandline: /usr/bin/unattended-upgrade
127 Commandline: /usr/bin/unattended-upgrade
128 Commandline: /usr/bin/unattended-upgrade
129 Commandline: /usr/bin/unattended-upgrade
130 Commandline: /usr/bin/unattended-upgrade
131 Commandline: /usr/bin/unattended-upgrade
132 Commandline: /usr/bin/unattended-upgrade
133 Commandline: apt install gpaste
134 Commandline: apt install sysstat
135
136 ======================================================================
137                    Recent System Changes/Updates
138 ======================================================================
139
140 2024-01-13 16:35:30 status triggers-pending man-db:amd64 2.10.2-1
141 2024-01-13 16:35:30 status unpacked sysstat:amd64 12.5.2-2ubuntu0.2
142 2024-01-13 16:35:30 startup packages configure
143 2024-01-13 16:35:30 configure sysstat:amd64 12.5.2-2ubuntu0.2 <none>
144 2024-01-13 16:35:30 status unpacked sysstat:amd64 12.5.2-2ubuntu0.2
145 2024-01-13 16:35:30 status half-configured sysstat:amd64 12.5.2-2ubuntu0.2
146 2024-01-13 16:35:33 status installed sysstat:amd64 12.5.2-2ubuntu0.2
147 2024-01-13 16:35:33 trigproc man-db:amd64 2.10.2-1 <none>
148 2024-01-13 16:35:33 status half-configured man-db:amd64 2.10.2-1
149 2024-01-13 16:35:33 status installed man-db:amd64 2.10.2-1
150
151 [INFO]: System updates frequency is normal and healthy .
```

3.2.7 Recommendations:

- Summarizes recommendations based on the health check results.
- Provides guidance on actions to take for concerns related to disk usage, memory, processes, services, and system updates.

```
153 ================================================================================
154                             Recommendations
155 ================================================================================
156
157 If any of the checks above are concerning, consider the following actions:
158 - Disk Usage: Consider cleaning up unnecessary files or expanding your storage.
159 - Memory Usage: Consider closing unnecessary applications or expanding your memory.
160 - Processes: Consider investigating any unfamiliar processes using a large amount of resources.
161 - Services: Consider disabling unnecessary services.
162 - Updates: Consider updating your system if it has not been updated recently.
163
164 ================================================================================
165                             End Of The Report
166 ================================================================================
```

3.3 Optimizations

- In the department of optimizations , I didn't have much room , because the script deals with hardware , so the interactivity with the user is very minimal , and the checks aren't super long for me to divide them into separate checks and ask the user which check he/she wants , In addition to that The script primarily utilizes shell built-in commands (e.g., `df`, `free`, `ps`, `systemctl`), which often execute faster than any external tool , so I didn't have much to optimize.

- I tried to make the report as human-readable and organized as possible, this was the thing I tried to optimize, and I believe that I achieved a good result.

3.4 Insights from performance analyses

3.4.1 Modularity:

- The script is well-organized into functions, promoting code modularity and readability.

3.4.2 Informative Output:

- The script generates a detailed and informative report, aiding users in understanding the system's health.

3.4.3 User-Friendly Recommendations:

- The recommendations section is user-friendly, offering actionable advice for addressing identified issues.

3.4.4 Data Presentation:

- Data presentation, such as disk usage and memory information, is clear and concise, making it easy for users to interpret.

The End