



NOV 2025

Capstone project

Submitted By: Sumaiya Kalim

Github id: Hiromitachi

Student at: ITER, SOA

Bash Scripting Suite for System Maintenance

Introduction

This project is a Linux-based System Maintenance Suite developed using Bash scripting.

It automates common system administration tasks such as:

- Creating backups
- Updating packages
- Cleaning unused files
- Monitoring system logs
- Centralized menu-based control

The goal is to simplify daily maintenance and provide a reliable automation tool for personal or server environments.

Technologies Used

COMPONENT	TECHNOLOGY
OS	Ubuntu 24.04/WSL
LANGUAGE	Bash Shell Scripting
TOOLS	tar, apt, grep, tail, cron, find
VERSION CONTROL	Git + GitHub

Project Structure

```
linux-maintenance-project/
|
├── backup.sh
├── system_update.sh
├── monitor_logs.sh
├── maintenance_menu.sh
├── .env
├── exclude.txt
└── logs/
    └── screenshots/
        └── README.md
```

Script Explanations & Code

backup.sh

- Creates compressed backups
- Reads configuration from .env
- Supports excludes
- Rotates older backups
- Saves logs

```
#!/usr/bin/env bash
#Backup Script

set -Eeuo pipefail
IFS=$'\n\t'
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
LOG_DIR="${SCRIPT_DIR}/logs"
LOG_FILE="${LOG_DIR}/backup_$(date +%Y%m%d).log"
ENV_FILE="${SCRIPT_DIR}/.env"
EXCLUDE_FILE="${SCRIPT_DIR}/exclude.txt"
mkdir -p "$LOG_DIR"
log() { printf "[%s] %s\n" "$(date '+%F %T')" "$*" | tee -a "$LOG_FILE"; }
die() { log "ERROR: $*"; exit 1; }
usage() {
    cat <<USAGE
Usage: ${0##*/} [--dry-run] [--help]
      --dry-run Show actions without creating backup
      --help   Show this help
USAGE
}
[[ -f "$ENV_FILE" ]] || die "Missing .env file"; source "$ENV_FILE"
: ${SRC_DIR:="$HOME/Documents"}
: ${DEST_DIR:="$HOME/backups}"
: ${RETAIN:=5}
: ${ALWAYS_DRY_RUN:=0}
DRY_RUN=0
for arg in "${@:-}", do
    case "$arg" in
        --dry-run) DRY_RUN=1;;
        --help|-h) usage; exit 0 ;;
        *) die "Unknown argument: $arg" ;;
    esac
done
[[ "$ALWAYS_DRY_RUN"-eq 1 ]] && DRY_RUN=1
```

```

--help|-h) usage; exit 0;
die "Unknown argument: $arg"
esac
done
[[ "$ALWAYS_DRY_RUN" -eq 1 ]] && DRY_RUN=1

[[ -d "$SRC_DIR" ]] || die "SRC_DIR does not exist: $SRC_DIR"
mkdir -p "$DEST_DIR"

TS="$(date +%Y%m%d_%H%M%S)"
ARCHIVE="backup_${TS}.tar.gz"
ARCHIVE_PATH="${DEST_DIR}/${ARCHIVE}"

log "==== Backup started ===="
log "Source: $SRC_DIR"
log "Destination: $DEST_DIR"
log "Archive: $ARCHIVE_PATH"

TAR_CMD=(tar -czf "$ARCHIVE_PATH" -C "$SRC_DIR" .)
if [[ -s "$EXCLUDE_FILE" ]]; then
    TAR_CMD=(tar --exclude-from="$EXCLUDE_FILE" -czf "$ARCHIVE_PATH" -C "$SRC_DIR" .)
    log "Using exclude patterns"
fi

if [[ "$DRY_RUN" -eq 1 ]]; then
    log "[DRY-RUN] Would run: ${TAR_CMD[*]}"
    (cd "$SRC_DIR" && find . -type f | head -n 20) | tee -a "$LOG_FILE"
else
    "${TAR_CMD[@]}"
    log "Archive created: $ARCHIVE_PATH"
fi

# rotation: keep last N archives
mapfile -t archives < <(ls -1t "$DEST_DIR"/backup_*tar.gz 2>/dev/null || true)
count=${#archives[@]}
if (( count > RETAIN )); then
    to_delete=( "${archives[@]:RETAIN}" )
    for f in "${to_delete[@]}"; do
        if [[ "$DRY_RUN" -eq 1 ]]; then
            log "[DRY-RUN] Would remove: $f"
        else
            rm -f -- "$f" && log "Removed old backup: $f"
        fi
    done
else
    log "No rotation needed (have $count, retain $RETAIN)."
fi

log "==== Backup finished ===="

```

system_update.sh

- Updates system
- Removes unwanted packages
- Cleans cache & logs

```
#!/usr/bin/env bash
# System Update & Cleanup

set -Eeuo pipefail
IFS=$'\n\t'

SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
LOG_DIR="${SCRIPT_DIR}/logs"
LOG_FILE="${LOG_DIR}/update_$(date +%Y%m%d_%H%M%S).log"

mkdir -p "$LOG_DIR"

log(){ printf "[%s] %s\n" "$(date '+%F %T')" "$*" | tee -a "$LOG_FILE"; }

log "==== System Update Started ===="
log "Updating package lists..."
sudo apt update | tee -a "$LOG_FILE"

log "Upgrading packages..."
sudo apt upgrade -y | tee -a "$LOG_FILE"

log "Removing unused dependencies..."
sudo apt autoremove -y | tee -a "$LOG_FILE"

log "Cleaning apt cache..."
sudo apt clean

log "Clearing thumbnail cache..."
rm -rf ~/.cache/thumbnails/* || true

log "Cleaning system logs older than 7 days..."
sudo find /var/log -type f -mtime +7 -exec rm -f {} \; || true

log "==== System Update Finished ===="
```

monitor_logs.sh

- Detects errors/warnings
- Real-time alerting
- Saves alert logs

```
#!/usr/bin/env bash
# Log Monitoring (real-time + one-time scan with rate limit)

set -Eeuo pipefail
IFS=$'\n\t'
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
LOG_DIR="${SCRIPT_DIR}/logs"
ALERT_LOG="${LOG_DIR}/alerts_$(date +%Y%m%d).log"
ENV_FILE="${SCRIPT_DIR}/.env"
mkdir -p "$LOG_DIR"
# defaults (can be overridden in .env or CLI)
TARGET_FILE="/var/log/syslog"
PATTERN="error|failed|exception|segfault"
RATE_WINDOW=60    # seconds
RATE_THRESHOLD=5  # max matches per window before storm notice
[[ -f "$ENV_FILE" ]] && source "$ENV_FILE"
: "${TARGET_FILE:=${TARGET_FILE}}"
: "${PATTERN:=${PATTERN}}"
: "${RATE_WINDOW:=${RATE_WINDOW}}"
: "${RATE_THRESHOLD:=${RATE_THRESHOLD}}"
usage() {
cat <<USAGE
Usage: ${0##*/} [--file PATH] [--pattern REGEX] [--once] [--help]
--file PATH    Log file to monitor (default: $TARGET_FILE)
--pattern REGEX Case-insensitive regex (default: $PATTERN)
--once        Scan file once and exit
--help        Show this help
USAGE
}
ONCE=0
while [[ $# -gt 0 ]]; do
case "$1" in
--file)  TARGET_FILE="$2"; shift 2 ;;
--pattern) PATTERN="$2"; shift 2 ;;
--once)  ONCE=1; shift ;;
--help|-h) usage; exit 0 ;;
esac
done
```

```

*) echo "Unknown arg: $1"; usage; exit 1 ;;
esac
done
[[ -f "$TARGET_FILE" ]] || { echo "ERROR: File not found: $TARGET_FILE" | tee -a "$ALERT_LOG"; exit 1; }

log_alert() {
    local line="$1"
    printf "[%s] ALERT in %s: %s\n" "$(date '+%F %T')" "$TARGET_FILE" "$line" | tee -a "$ALERT_LOG"
}

storm_notice_printed=0
declare -A bucket
bucket_start=$(date +%-s)
process_line() {
    local line="$1"
    if [[ "$line" =~ $PATTERN ]] || echo "$line" | grep -Eq -- "$PATTERN"; then
        log_alert "$line"
        local now count
        now=$(date +%-s)
        if (( now - bucket_start >= RATE_WINDOW )); then
            bucket=(); bucket_start=$now; storm_notice_printed=0
        fi
        count=${bucket[@]:-0}; count=$((count+1)); bucket[@]+=( "$now" )
        if (( count > RATE_THRESHOLD )) && (( storm_notice_printed == 0 )); then
            printf "[%s] NOTICE: High alert rate (> %d matches in %ds)\n" \
                "$(date '+%F %T')" "$RATE_THRESHOLD" "$RATE_WINDOW" | tee -a "$ALERT_LOG"
            storm_notice_printed=1
        fi
    fi
}
if (( ONCE )); then
    while IFS= read -r line; do process_line "$line"; done < "$TARGET_FILE"
    echo "Scan complete. Alerts (if any) in: $ALERT_LOG"
    exit 0
fi
echo "Monitoring: $TARGET_FILE"
echo "Pattern: $PATTERN"
echo "Writing alerts to: $ALERT_LOG"
echo "Press Ctrl+C to stop."
tail -Fn0 "$TARGET_FILE" | while IFS= read -r line; do process_line "$line"; done

```

maintenance menu.sh

- Provides interactive menu
- Calls all scripts inside

```
#!/usr/bin/env bash
# Menu to run all tools

set -Eeuo pipefail
IFS=$'\n\t'
SCRIPT_DIR=$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
show_menu() {
    clear
    echo "===== SYSTEM MAINTENANCE MENU ====="
    echo "1) Run Backup"
    echo "2) System Update & Cleanup"
    echo "3) Monitor Logs (real-time)"
    echo "4) Scan Logs Once"
    echo "5) Exit"
    echo "====="
    read -p "Enter choice: " choice
}

run_backup() {
    echo "Running Backup..."
    if [[ -f "$SCRIPT_DIR/backup.sh" ]]; then
        "$SCRIPT_DIR/backup.sh"
    else
        echo "ERROR: backup.sh not found!"
    fi
    read -p "Press Enter to continue..." 
}

run_update() {
    echo "Running System Update..."
    if [[ -f "$SCRIPT_DIR/system_update.sh" ]]; then
        "$SCRIPT_DIR/system_update.sh"
    else
        echo "ERROR: system_update.sh not found!"
    fi
    read -p "Press Enter to continue..." 
}

monitor_logs() {
    echo "Real-time log monitoring (Ctrl + C to stop)"
}
```

```
if [[ -f "$SCRIPT_DIR/monitor_logs.sh" ]]; then
"$SCRIPT_DIR/monitor_logs.sh"
else
echo "ERROR: monitor_logs.sh not found!"
fi
read -p "Press Enter to continue..."
}
```

```
scan_logs_once() {
echo "Scanning logs once"
if [[ -f "$SCRIPT_DIR/monitor_logs.sh" ]]; then
"$SCRIPT_DIR/monitor_logs.sh" --once
else
echo "ERROR: monitor_logs.sh not found!"
fi
read -p "Press Enter to continue..."
}
```

```
while true; do
show_menu
case "$choice" in
1) run_backup ;;
2) run_update ;;
3) monitor_logs ;;
4) scan_logs_once ;;
5) echo "Exiting..."; exit 0 ;;
*) echo "Invalid option"; sleep 1 ;;
esac
done
```

Screenshots

Initialisation

```
sumaiya@sumi-pc:~$ mkdir ~/linux-maintenance-project
mkdir: cannot create directory '/home/sumaiya/linux-maintenance-project': File exists
sumaiya@sumi-pc:~$ cd ~/linux-maintenance-project
sumaiya@sumi-pc:~/linux-maintenance-project$ nano .env
sumaiya@sumi-pc:~/linux-maintenance-project$ nano exclude.txt
sumaiya@sumi-pc:~/linux-maintenance-project$ nano backup.sh
sumaiya@sumi-pc:~/linux-maintenance-project$ chmod +x backup.sh
sumaiya@sumi-pc:~/linux-maintenance-project$ ./backup.sh --dry-run
[2025-11-07 19:00:03] ERROR: SRC_DIR does not exist: /home/sumaiya/Documents
sumaiya@sumi-pc:~/linux-maintenance-project$ mkdir -p ~/Documents
./backup.sh --dry-run
[2025-11-07 19:00:53] === Backup started ===
[2025-11-07 19:00:53] Source:      /home/sumaiya/Documents
[2025-11-07 19:00:53] Destination: /home/sumaiya/backups
[2025-11-07 19:00:53] Archive:     /home/sumaiya/backups/backup_20251107_190053.tar.gz
[2025-11-07 19:00:53] Using exclude patterns
[2025-11-07 19:00:53] [DRY-RUN] Would run: tar
--exclude-from=/home/sumaiya/linux-maintenance-project/exclude.txt
-czf
/home/sumaiya/backups/backup_20251107_190053.tar.gz
-C
/home/sumaiya/Documents
.

[2025-11-07 19:00:53] === Backup finished ===
sumaiya@sumi-pc:~/linux-maintenance-project$ ./backup.sh
[2025-11-07 19:01:43] === Backup started ===
[2025-11-07 19:01:43] Source:      /home/sumaiya/Documents
[2025-11-07 19:01:43] Destination: /home/sumaiya/backups
[2025-11-07 19:01:44] Archive:     /home/sumaiya/backups/backup_20251107_190143.tar.gz
[2025-11-07 19:01:44] Using exclude patterns
[2025-11-07 19:01:44] Archive created: /home/sumaiya/backups/backup_20251107_190143.tar.gz
[2025-11-07 19:01:44] === Backup finished ===
sumaiya@sumi-pc:~/linux-maintenance-project$ ls ~/backups
backup_20251107_190143.tar.gz
sumaiya@sumi-pc:~/linux-maintenance-project$ nano system_update.sh
sumaiya@sumi-pc:~/linux-maintenance-project$ chmod +x system_update.sh
sumaiya@sumi-pc:~/linux-maintenance-project$ ./system_update.sh
```

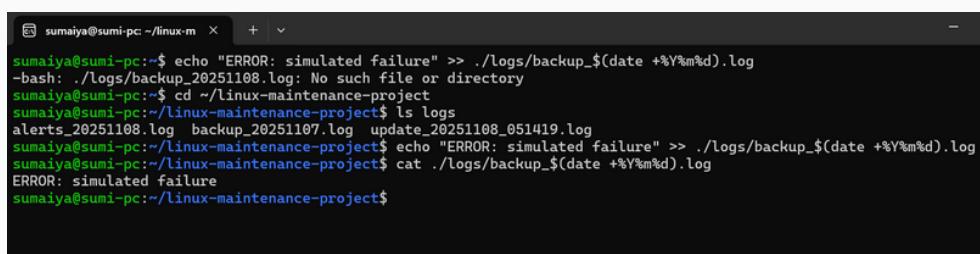
system completed

```
Scan complete. Alerts (if any) in: /home/sumaiya/linux-maintenance-project/logs/alerts_20251108.log
sumaiya@sumi-pc:~/linux-maintenance-project$
```

backup finished

```
2025-11-07 19:01:44] Archive created: /home/sumaiya/backups/backup_2
20251107_190143.tar.gz
2025-11-07 19:01:44] === Backup finished ===
sumaiya@sumi-pc:~/Linux-maintenance-project$ ls ~/backups
backup_20251107_190143.tar.gz
sumaiya@sumi-pc:~/Linux-maintenance-project$ |
```

stimulated error



The screenshot shows a terminal window with the following session:

```
sumaiya@sumi-pc:~/Linux-maintenance-project$ echo "ERROR: simulated failure" >> ./logs/backup_$(date +%Y%m%d).log
-bash: ./logs/backup_20251108.log: No such file or directory
sumaiya@sumi-pc:~/Linux-maintenance-project$ cd ~/Linux-maintenance-project
sumaiya@sumi-pc:~/Linux-maintenance-project$ ls logs
alerts_20251108.log  backup_20251107.log  update_20251108_051419.log
sumaiya@sumi-pc:~/Linux-maintenance-project$ echo "ERROR: simulated failure" >> ./logs/backup_$(date +%Y%m%d).log
sumaiya@sumi-pc:~/Linux-maintenance-project$ cat ./logs/backup_$(date +%Y%m%d).log
ERROR: simulated failure
sumaiya@sumi-pc:~/Linux-maintenance-project$
```

OUTPUT:

```
sumaiya@sumi-pc: ~/linux-m X + ▾
===== SYSTEM MAINTENANCE MENU =====
1) Run Backup
2) System Update & Cleanup
3) Monitor Logs (real-time)
4) Scan Logs Once
5) Exit
=====
Enter choice: 1
Running Backup...
[2025-11-08 07:32:11] === Backup started ===
[2025-11-08 07:32:11] Source:      /home/sumaiya/Documents
[2025-11-08 07:32:11] Destination: /home/sumaiya/backups
[2025-11-08 07:32:11] Archive:     /home/sumaiya/backups/backup_20251108_073211.tar.gz
[2025-11-08 07:32:11] Using exclude patterns
[2025-11-08 07:32:11] Archive created: /home/sumaiya/backups/backup_20251108_073211.tar.gz
[2025-11-08 07:32:11] === Backup finished ===
Press Enter to continue...|
```

```
sumaiya@sumi-pc: ~/linux-m X + ▾
Enter choice: 2
Running System Update...
[2025-11-08 07:32:19] === System Update Started ===
[2025-11-08 07:32:19] Updating package lists...

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists...
Building dependency tree...
Reading state information...
All packages are up to date.
[2025-11-08 07:32:23] Upgrading packages...

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
[2025-11-08 07:32:24] Removing unused dependencies...

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Reading package lists...
Building dependency tree...
Reading state information...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
[2025-11-08 07:32:25] Cleaning apt cache...
[2025-11-08 07:32:25] Clearing thumbnail cache...
[2025-11-08 07:32:25] Cleaning system logs older than 7 days...
[2025-11-08 07:32:25] === System Update Finished ===
Press Enter to continue...|
```

```
sumaiya@sumi-pc: ~/linux-m ~ + ^
```

```
===== SYSTEM MAINTENANCE MENU =====
1) Run Backup
2) System Update & Cleanup
3) Monitor Logs (real-time)
4) Scan Logs Once
5) Exit
=====
Enter choice: 3
Real-time log monitoring (Ctrl + C to stop)
Monitoring: /var/log/syslog
Pattern: error|failed|exception|segfault
Writing alerts to: /home/sumaiya/linux-maintenance-project/logs/alerts_20251108.log
Press Ctrl+C to stop.
```

```
sumaiya@sumi-pc: ~/linux-m ~ + ^
```

```
===== SYSTEM MAINTENANCE MENU =====
1) Run Backup
2) System Update & Cleanup
3) Monitor Logs (real-time)
4) Scan Logs Once
5) Exit
=====
Enter choice: 4
Scanning logs once
[2025-11-08 07:31:35] ALERT in /var/log/syslog: 2025-11-07T18:35:53.392373+00:00 sumi-pc kernel: Failed to register legacy timer interrupt
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.392507+00:00 sumi-pc kernel: ACPI: _OSC evaluation for CPUs failed, trying _PDC
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.392609+00:00 sumi-pc (udev-worker)[150]: sdc: Process '/usr/bin/umount -m /usr/bin/snap auto-import --mount=/dev/sdc' failed with exit code 1.
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.392613+00:00 sumi-pc (udev-worker)[153]: sdb: Process '/usr/bin/umount -m /usr/bin/snap auto-import --mount=/dev/sdb' failed with exit code 1.
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.392620+00:00 sumi-pc (udev-worker)[156]: sda: Process '/usr/bin/umount -m /usr/bin/snap auto-import --mount=/dev/sda' failed with exit code 1.
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.392623+00:00 sumi-pc (udev-worker)[161]: sdd: Process '/usr/bin/umount -m /usr/bin/snap auto-import --mount=/dev/sdd' failed with exit code 1.
[2025-11-08 07:31:36] NOTICE: High alert rate (> 5 matches in 60s)
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.392633+00:00 sumi-pc systemd[1]: apport-autoreport.path - Process error reports when automatic reporting is enabled (file watch) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/autoreport).
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.392638+00:00 sumi-pc systemd[1]: apport-autoreport.timer - Process error reports when automatic reporting is enabled (timer based) was skipped because of an unmet condition check (ConditionPathExists=/var/lib/apport/autoreport).
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.392865+00:00 sumi-pc kernel: RAS: Correctable Errors collector initialized.
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.393163+00:00 sumi-pc kernel: WSL (265) ERROR: CheckConnection: getaddrinfo() failed: -5
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.393166+00:00 sumi-pc kernel: misc dxg: dxgk: dxgkio_is_feature_enabled: Ioctl failed: -22
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.393167+00:00 sumi-pc kernel: misc dxg: dxgk: dxgkio_query_adapter_info: Ioctl failed: -22
[2025-11-08 07:31:36] ALERT in /var/log/syslog: 2025-11-07T18:35:53.393170+00:00 sumi-pc kernel: message repeated 2 times: [ misc dxg
```

results:

- All scripts ran successfully
- Automation reduced manual effort
- Menu gives easy control
- Logs stored for reference

Conclusion:

The Linux Maintenance Suite improves system usability by automating routine tasks. It demonstrates shell scripting ability, file automation, logging, and real system administration concepts useful in real-world DevOps environments.