

Empty ALL LOGs on a Linux Server

Download Script proxmox_subscription_autofix.sh

? One-Line Download & Execute:

```
clear && curl -fsSL https://docs.greenhome.stream/attachments/2 -o empty_log_files.sh && chmod +x empty_log_files.sh && clear && ./empty_log_files.sh
```

The below script is scan all the Linux Server from the [/] root folder of the system and all its sub directories for files with extension .log

Then it will make a file with filename "**log-files-list.txt**" and add line by line every log file it finds with it complete path.

After make the file "**log-files-list.txt**" then it takes line by line every log file and empty it, at then end it delete the file "**log-files-list.txt**".

Creating the script

Open the terminal and type the below command :

```
nano empty_log_files.sh
```

and then copy / paste the below code inside the file "**empty_log_files.sh**"

```
#!/bin/bash
clear
# Filename for log files list
LOG_FILE_LIST="log-files-list.txt"

# Function to find and list log files
find_log_files() {
    echo "Scanning for log files from /..."
    # Use find to locate files with `.log` extension
    find / -type f -name "*.log" > "$LOG_FILE_LIST"

    if [ $? -eq 0 ]; then
        echo "Log files found and saved to $LOG_FILE_LIST."
```

```

else
    echo "Failed to search for log files."
    exit 1
fi
}

# Function to empty log files
empty_log_files() {
    echo "Emptying the log files listed in $LOG_FILE_LIST..."
    # Read through the log file list and empty each log file
    while IFS= read -r log_file; do
        if [ -f "$log_file" ]; then
            > "$log_file" # Empty the log file
            echo "Emptied: $log_file"
        else
            echo "Warning: $log_file does not exist or is not a file."
        fi
    done < "$LOG_FILE_LIST"
}

# Main script execution
find_log_files
empty_log_files
rm log-files-list.txt
echo "Completed processing log files."

```

then press **Ctrl + x** it will prompt you [**Save modified buffer?**] press **y** and **enter**

Make the script executable

After that you need to make the script executable, so you will write the below command :

```
chmod +x empty_log_files.sh
```

Add the script to cron job to run automatically

To run the script every 5 minutes follow the below steps, you can change the minutes according to your need.

```
crontab -e
```

then go at the end of the file and add the below line, change the `your_path` with your folder where the script is locate.

```
* /5 * * * * /your_path/empty_log_files.sh
```

```
GNU nano 7.2
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/5 * * * * /root/empty_log_files.sh
```

then press **Ctrl + x** it will prompt you [`Save modified buffer?`] press **y** and **enter**

You are done the script in the location `/your_path/empty_log_files.sh` it will automatically run every 5 minutes.

Download the script empty_log_files.sh

Attachment Link [empty_log_files.sh](#)

Ins0mniA

Revision #12

Created 2025-04-21 20:35:49 EEST by Green

Updated 2025-09-04 02:05:30 EEST by Green