VersionCleanup 2.0

A simple Script

Documentation Version 2.0

Contents

Overview

`VersionCleanup.sh` is a Bash script designed to automate the cleanup of outdated files and folders matching the pattern you define within a specified directory on your server. It retains only the latest two versions of these files/folders, based on sorting criteria, and deletes the rest. Additionally, it logs the actions taken during each execution, including deletions, to a log file.

Features

- **Selective Deletion:** Targets only files matching a specific naming pattern, deleting all but the latest two versions.
- **Logging:** Records all deletion actions in a dated log file for auditing and review purposes.
- **Safety Checks:** Includes basic safety checks to prevent execution in unintended directories.

Requirements

- A Linux operating system with Bash shell.
- Read, write, and delete permissions in the target directory for the files concerned.

Installation

The script requires no special installation. Simply download or copy it to your desired directory and make it executable:

```
chmod +x VersionCleanup.sh
```

Usage

To run the script manually, navigate to the script's directory and execute it as follows:

```
./VersionCleanup.sh
```

Configuration

Edit the script to set the `directory` path at the beginning of the script to your specific target directory where the files should be cleaned up.

Automation with Cron

To automate the script execution, add a cron job that runs `VersionCleanup.sh` according to your schedule. For example, to run the script on the first day of every month at 3 AM, add the following entry to your crontab:

0 3 1 * * /path/to/VersionCleanup.sh

Replace `/path/to/` with the actual path where `VersionCleanup.sh` is stored.

Logging

The script writes a log of each execution with timestamps to a file named in the format `deletionLog_$(date +%Y-%m-%d_%H-%M-%S).log`, stored in the same directory as the script. Regularly check these files to ensure the script is functioning as expected.

Safety Tips

- Execute the script in a test environment before deploying it in a production environment to avoid unintended data loss.
- Ensure you have backups of important data before running the script.

Support

For questions or assistance with using this script, please contact the author or your system administrator.