

External Backup Script using Rclone

This script allows for an automatic backup of important information related to the configuration of your applications, excluding Plex. It can then be configured to store it on either Google drive or a local folder on your Ultra.cc service.

This is an unofficial script that is provided for your convenience. The script is provided as-is and may not be updated or maintained by Ultra.cc. Customers are welcome to use and customize unofficial scripts for their unique needs and requirements. Unofficial support may be offered via [Discord](#) only and at the sole discretion of Ultra.cc staff. Use at your own risk.

Initial Setup

- Create three new folders, one may already exist this is fine
 - `mkdir ~/lock`
 - `mkdir ~/scripts`
 - `mkdir ~/autobackup`
- Then you need to enter the new folder called scripts
 - `cd ~/scripts`
- And create the script file
 - `nano backup.sh`
- Paste the following lines into it :

```
#!/bin/bash
exec {lock_fd}>/home/your-username/lock/BackupLock || exit 1
flock -n "$lock_fd" || { echo "ERROR: flock() failed." >&2; exit 1; }
if [ -z "$STY" ]; then exec screen -dm -S autobackup /bin/bash "$0"; fi

DATE=$(date +%Y-%m-%d-%H%M%S)
tar --exclude="$HOME"/.config/plex -czvf "$HOME"/autobackup/MyUSBbackup-$DATE.tar.gz
"$HOME"/.apps "$HOME"/.config

rclone copy -v --bwlimit=5M /home/your-username/autobackup/ remotename:/"My USB Backup"
```

```
flock -u "$lock_fd"
```

- Save it by pressing `Ctrl+X` then `Y` then `Enter`.

You will need to change the paths `home/your - username` to match your own home and username. You will also be required to change remotename to match your rclone remote name which can be found with the *rclone config* command or *rclone listremotes*. These need to be run via SSH Command line.

Testing

- So to test first we navigate to `~/scripts` folder we made earlier
 - `cd ~/scripts`
- Then we need to allow the backup.sh permissions to run
 - `chmod +x backup.sh`
- And finally, run it
 - `./backup.sh`

If all went well `* tar -zvf ~/autobackup/.tar.gz` should show a bunch of files starting with `/home/username/.config` and `/home/username/.apps` inside.

If the script is running and you were to rerun it, you may see an error message “Flock Failed” this is a file lock to stop multiple downloads running and is normal. If you are sure it isn’t running you can delete the lock file from `~/lock`. You can also check the progress of the backup script, which is running in a screen with the command `screen -rd autobackup` and then to exit the screen while keeping the command running use the keyboard combo `CONTROL + A + D`

If all is well after the test, we can automate the check via crontab

- Open crontab with:
 - `crontab -e`
- Inside the crontab add a single line under everything else in the file that looks like this:
 - `0 0 */3 * * /home/your - username/scripts/backup.sh`
- Save it by pressing `Ctrl+X` then `Y` then `Enter`.

The script will now run every 3 days, checking for files that have changed and syncing them to the destination folder.

Revision #11

Created 14 July 2021 17:40:30 by Joe

Updated 1 September 2023 18:00:46 by varg