

Using Rclone FTPS

Ultra.cc support is unable to provide support for rclone, this is due to the large volume of variables and different configurations possible with rclone. The guides found here on the knowledge-base should be able to guide you through using rclone, and any further questions can easily be answered with a quick Google search. You may also be able to find community support for rclone through our community Discord server or the Rclone forums.

Rclone is a software primarily used with cloud storage providers; however, it is also an easy-to-use Command-line download manager for many protocols, including FTP/FTPS/SFTP. We will show you today how to install rclone onto a Linux machine on your local network and how to set up automatic downloading from your Ultra.cc Slot.

Requirements:

Host machine on your home/local network running a Linux distribution (Mac OS can also be used but is not in the scope of this guide; however, it may still work as a general guide)

Ultra.cc FTP Slot details

Installing Rclone on the Host Machine

Installing rclone is made incredibly easy by the developer. First login to your Host via SSH; this can be under a `sudo` user, but it is recommended to install rclone as `root` user and later set up rclone under a sudo/sub-user, never set up under `root`.

Once logged in, run this curl command

```
curl https://rclone.org/install.sh | sudo bash
```

If you receive a message telling you that the “curl command cannot be found,” you will need to install it and run the command above again. To install curl, run `apt-get install curl -y`

Once rclone is installed, it is recommended you switch to a new user and configure under that user. If you have not created a new user, the steps are simple.

`adduser <username>` and make sure to replace `<username>` with a username of your choice.

Follow the on-screen prompts for user information and password. Once that is completed, it is an excellent idea to promote this user to sudo; this allows for the execution of higher privileged commands, which are not required for Rclone but are certainly worth setting up.

To add a user to sudo after creation, run this command.

```
adduser <username> sudo
```

If, for some odd reason, your distro did not have sudo installed, now might be a good time to run an install just in case:

```
apt-get install sudo -y
```

Now we can switch to our `<username>` user

```
su <username>
```

Type `cd` to jump to the user home folder. This isn't strictly required but is best practice.

Configuring our Rclone FTP remote

The first step is to invoke the rclone config menu with the `rclone config` command

This will present a list of options

```
e) Edit existing remote

n) New remote

d) Delete remote

r) Rename remote

c) Copy remote

s) Set configuration password

q) Quit config

e/n/d/r/c/s/q\>
```

Press N then enter

Now enter a remote name: you will need this later when writing our automated script.

Now you will be presented with an extremely long list of options at the time of writing this `16` is FTP, but this may change as updates release

Field	Value
Host	Your Ultra Address (servername.usbx.me) or IP (185.123.123.123)

Field	Value
User	Your Ultra Username
Port	Port 21 is standard. You can press Enter
FTP password	Press Y, then enter your FTP Password. You will be asked for this twice.
Use FTPS over TLS (Implicit)	False
Use FTP over TLS (Explicit)	True
Edit advanced config? (y/n)	Press Y
Maximum number of FTP simultaneous connections, 0 for unlimited	Press enter
no_check_certificate	True
Disable using EPSV even if server advertises support	Press Enter
Enter a encoder.MultiEncoder value. Press Enter for the default ("Slash,Del,Ctl,RightSpace,Dot").	Press Enter
y) Yes this is OK (default)	Press Y to complete the rclone configuration

Testing Remote for access

Once the configuration of the remote is completed, it is a relatively simple job to check if all is configured correctly

One command should list all of the folders in your Ultra Slot Home folder.

```
rclone lsd remote-name:
```

 Replace remote-name with the remote name you chose previously.

Automating download from Ultra.cc

Before using this script on any machine other than your Ultra service, make sure `screen` is installed.

Our first step is to write a simple script. We will set up a `lock` folder for a file lock system so the script will not run if it is already running

Create the lock folder with `mkdir -p ~/lock`

If you ever get a flock failed command, delete any files in ~/lock with the command `rm -rdf ~/lock/*`

And create the script file

```
nano FTP.sh
```

Paste the following lines into it :

```
#!/bin/bash

exec {lock_fd}>/home/<username>/lock/FTPLock || exit 1
```

```
flock -n "$lock_fd" || { echo "ERROR: flock() failed." >&2; exit 1; }

if [ -z "$STY" ]; then exec screen -dm -S FTP /bin/bash "$0"; fi

rclone copy -v remote-name: /ultraslotfoldertocopy /home/<username>/pathtosavefiles

flock -u "$lock_fd"
```

Save it by pressing Ctrl+X then Y Enter.

You will need to change the paths `ultraslotfoldertocopy` to match the folder you wish to download. You will also be required to alter remote-name to match your rclone remote name that you chose previously. It can be found with the *rclone config* command or *rclone listremotes*. These need to be run via the SSH Command line.

Now we need to allow it to run with the command `chmod +x FTP.sh`

And finally, run it

```
./FTP.sh
```

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 inside `~/lock`. You can also check the progress of the backup script, which is running in a screen with the command

```
screen -rd FTP
```

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

Open crontab with

```
crontab -e
```

You may have a choice of editors. We recommend Nano

Inside the crontab add a single line under everything else in the file that looks like this

```
* * * * * /home/ultra/FTP.sh
```

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

The script will now run every 1 minute, checking for files that have changed and syncing them to the destination folder

Revision #8

Created 5 July 2021 15:10:35 by Joe

Updated 24 November 2024 17:39:43 by varg