

Installation

You can use any Raspberry Pi with Rclone, I mentioned Raspberry Pi 3 as the latest revision. I skip Raspbian installation as it is not the main goal of this article and it should not be a challenge. All details related to Raspbian installation you can find [here](#).

Download [rclone arm 32 bit](#) version and copy it to **/usr/local/bin**.

```
wget https://downloads.rclone.org/rclone-current-linux-arm.zip
unzip rclone-current-linux-arm.zip
sudo cp rclone-v1.41-linux-arm/rclone /usr/local/bin/
```

IMPORTANT:

Current version can be different, not 1.41, so please keep in mind to update copy command accordingly.

Setup

A comprehensive manual and documentation can be found [here](#).

I will describe setup for Google Drive (as my project was related to Google Drive usage). The initial setup for Google Drive involves getting a token from Google Drive which you need to do in your browser. So please create Google Account if you have not already done that and create folder on its root you want to use with **rclone** (for example rclone).

Run:

```
rclone config
```

and follow the prompt:

```
No remotes found - make a new one
n) New remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
n/r/c/s/q> n
name> remote
```

Type of storage to configure.

Choose a number from below, or type in your own value

- 1 / Alias for a existing remote
 \ "alias"
- 2 / Amazon Drive
 \ "amazon cloud drive"
- 3 / Amazon S3 Compliant Storage Providers (AWS, Ceph, Dreamhost, IBM COS, Minio)
 \ "s3"
- 4 / Backblaze B2
 \ "b2"
- 5 / Box
 \ "box"
- 6 / Cache a remote
 \ "cache"
- 7 / Dropbox
 \ "dropbox"
- 8 / Encrypt/Decrypt a remote
 \ "crypt"
- 9 / FTP Connection
 \ "ftp"
- 10 / Google Cloud Storage (this is not Google Drive)
 \ "google cloud storage"
- 11 / Google Drive
 \ "drive"
- 12 / Hubic
 \ "hubic"
- 13 / Local Disk
 \ "local"
- 14 / Mega
 \ "mega"
- 15 / Microsoft Azure Blob Storage
 \ "azureblob"
- 16 / Microsoft OneDrive
 \ "onedrive"
- 17 / Openstack Swift (Rackspace Cloud Files, Memset Memstore, OVH)
 \ "swift"
- 18 / Pcloud

```

    \ "pcloud"
19 / QingCloud Object Storage
    \ "qingstor"
20 / SSH/SFTP Connection
    \ "sftp"
21 / Webdav
    \ "webdav"
22 / Yandex Disk
    \ "yandex"
23 / http Connection
    \ "http"
Storage> 11
Google Application Client Id - leave blank normally.
client_id>
Google Application Client Secret - leave blank normally.
client_secret>
Scope that rclone should use when requesting access from drive.
Choose a number from below, or type in your own value
 1 / Full access all files, excluding Application Data Folder.
    \ "drive"
 2 / Read-only access to file metadata and file contents.
    \ "drive.readonly"
    / Access to files created by rclone only.
 3 | These are visible in the drive website.
    | File authorization is revoked when the user deauthorizes the app.
    \ "drive.file"
    / Allows read and write access to the Application Data folder.
 4 | This is not visible in the drive website.
    \ "drive.appfolder"
    / Allows read-only access to file metadata but
 5 | does not allow any access to read or download file content.
    \ "drive.metadata.readonly"
scope> 1
ID of the root folder - leave blank normally. Fill in to access "Computers" folders. (see docs).
root_folder_id>
Service Account Credentials JSON file path - leave blank normally.

```

```
Needed only if you want use SA instead of interactive login.
service_account_file>
Remote config
Use auto config?
  * Say Y if not sure
  * Say N if you are working on a remote or headless machine or Y didn't work
y) Yes
n) No
y/n> n
If your browser doesn't open automatically go to the following link: http://127.0.0.1:53682/auth
Log in and authorize rclone for access
Enter verification code> SOME_CODE_GOT_FROM_BROWSER
Configure this as a team drive?
y) Yes
n) No
y/n> n
-----
[remote]
client_id =
client_secret =
token = {"AccessToken":"xxxx.x.xxxxx_xxxxxxxxxxxx","RefreshToken":"1/xxxxxxxxxxxxxxxxxx_xxxxxxx","Expiry":"2014-03-16T13:57:58.955387075Z","Extra":null}
-----
y) Yes this is OK
e) Edit this remote
d) Delete this remote
y/e/d> y
Name Type
====
remote drive
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> q
```

IMPORTANT:

- Please keep in mind there can be new storage types under "**Type of storage to configure**" step, so just double check you select right cloud type.
- If you are going to call rclone from other application you need to copy **rclone.config** to the **.config** folder under **application's root** and assign **owner and group** to it same as **application** uses:

```
sudo mkdir -p /SOME_APP/.config/rclone
sudo cp /home/pi
sudo chown -R some_owner:some_group /SOME_APP/.config
```

Usage

Syncing data (checks if files are modified at source and destination):

```
rclone -v sync /home/pi/some_project_data remote:rclone --delete-before --include *.jpg
```

Moving data (copies data from source to destination and deletes it at source):

```
rclone -v move /home/pi/some_project_data remote:rclone --delete-after --include *.jpg --include *.log
```

More usage can be found [here](#).

Auto-run Rclone Tasks

Create an empty file with extension .sh and name it whatever you want.

```
sudo nano /home/pi/your_project/script.sh
```

Enter **rclone** code to be executed.

```
#!/bin/sh

sudo /usr/local/bin/rclone -v move /home/pi/some_project_data remote:rclone --delete-after --include *.jpg --include *.log
```

Save it and make it executable:

```
sudo chmod +x /home/pi/your_project/script.sh
```

Desktop Shortcut

Create a shortcut to your script file that you created as above. Create an empty file with extension **.desktop** under `/home/pi/.config/autostart` folder:

```
sudo nano /home/pi/.config/autostart/rc1one.desktop
```

If you don't see autostart folder, then create a folder called autostart within .config folder.

```
sudo mkdir /home/pi/.config/autostart
```

Copy and paste the following in the empty desktop file except you will need to change any value.

```
[Desktop Entry]
Type=Application
Name=Your_Name
Exec=/home/pi/your_project/script.sh
X-GNOME-Autostart-enabled=true
```

Save and close the file after changing all the necessary fields. You are done. Just test it out.

Crontab

The crontab (short for "cron table") is a list of commands that are scheduled to run at regular time intervals on your computer system.

If you wished to have your **rc1one** task run every day at 3am, your crontab entry would look like as follows. First, install your cronjob by running the following command:

```
crontab -e
```

Append the following entry:

```
0 3 * * * /home/pi/your_project/script.sh
```

Save and close the file.

Easy to remember cronjob syntax format hint:

```
* * * * * command to be executed
- - - - -
| | | | |
| | | | ---- Day of week (0 - 7) (Sunday=0 or 7)
| | | ----- Month (1 - 12)
| | ----- Day of month (1 - 31)
| ----- Hour (0 - 23)
----- Minute (0 - 59)
```

More details about crontab can be found [here](#).

Other Options

You can find more options to automatically run your scripts/tasks in my article how to add [led power button](#) to your project.

More Ideas

With Rclone you can even [mount](#) your remote cloud storage as file system. It is currently experimental, so use it with care but do not hesitate to try it as R&D project.