rclone backup to Cloud Storage

This tutorial covers how to backup your ROMs, BIOS and SAVE-games of your Batocera machine to a Cloud Storage. This allows you to recover your data easily to a previous state in case of data loss or restore data on a new installation.

Some small words at the beginning: What is **rclone**? Copying files can be done with numerous user interfaces, but rclone is a command-line tool specifically designed for copying files between cloud servers and another server or workstation. With simple tools, an interruption in service stops the file synchronization and copying processes. The rclone tool can recover if the connection between the workstation and server is interrupted.

Several notable cloud services support rclone including Amazon S3 buckets, Microsoft OneDrive, Dropbox, Google Cloud Storage, Google Drive, Microsoft Azure Blob and File Storage, and several other providers. Administrators use rclone for unidirectional file synchronization, meaning that files are copied from a source to a destination only.

Supported Cloud Services

Up to now (06/2025) these providers are supported:

1Fichier, Akamai Netstorage, Amazon S3 (or S3 compatible), Backblaze B2, Box, Citrix ShareFile, Cloudinary, Dropbox, Enterprise File Fabric, Files.com, FTP, Gofile, Google Cloud Storage, Google Drive, Google Photos, HDFS, HiDrive, HTTP, iCloud Drive, Internet Archive, Jottacloud, Koofr, Linkbox, Mail.ru Cloud, Mega, Memory, Microsoft Azure Blob Storage, Microsoft Azure Files Storage, Microsoft OneDrive, OpenDrive, OpenStack Swift, Oracle Object Storage, pCloud, PikPak, Pixeldrain, premiumize.me, put.io, Proton Drive, QingStor, Quatrix by Maytech, Seafile, SFTP, Sia, **SMB**, SugarSync, Storj, Uloz.to, Uptobox, **WebDAV**, Yandex Disk, Zoho WorkDrive, **The local filesystem**
The bold marked entries suggest that **rclone** can be used with really any Provider or network system. But consider then that for local FS it would be better to use rsync for example

You can always take a look here for a supported cloud storages

Setup rclone in Batocera

Now we want to setup **rclone** from Batocera. These instructions only work for Koofr, please visit

- Open a terminal or SSH session
- Enter `rclone config`
    - ```
      No remotes found, make a new one?
      n) New remote
      s) Set configuration password
      q) Quit config
      ```
      ```

```
n/s/q> n
```

    - Choose n
    - Give your cloud device a name, maybe `backup` or your provider name `yourCloudProvider`
- From here go to https://rclone.org/ and select your storage provider you are using or google `yourCloudProvider rclone`
- If everything went correct you will see a windows like

    - ```
      --------------------
      [yourProvider]
      type = yourCloudProvider
      provider = yourCloudProvider
      user = USERNAME
      password = *** ENCRYPTED ***
      --------------------
      y) Yes this is OK (default)
      e) Edit this remote
      d) Delete this remote
      y/e/d> y
      ```

- Enter y and everything is set and ready, wen need to make some test now

It's **worth** the matter but please ... we can't provide information for every usecase.

Testing

Remeber which name you entered? `backup` or `yourCloudProvider`?

List directories in top level of your `yourCloudProvider`

```
rclone lsd yourCloudProvider:
```

List all the files in your `yourCloudProvider`

```
rclone ls yourCloudProvider:
```

To copy a local directory to a directory called backup

```
rclone copy /home/source yourCloudProvider:backup
```

Service Script

A fellow user created a nice service script that will hit the purpose. So thank you Peter Boszo for your nice service script
```

Be aware this script syncs only to the storage. So if you fire up this on a new installation, all your files will be overwritten. This is caused by the –sync flag.

For preservatoin purposes it is included into this wiki as well with a nice addition that the name of the setted Provider can be changed.

RCLONE_SERVICE

```
#!/bin/bash
# ------------------------------ CONFIG BEGIN ------------------------
--------

# The location of Batocera data on your cloud storage.
destDir=/Batocera

# Waiting time between two backups in seconds.
# The below configuration means that the service will do a backup every
minute.
waitSeconds=60

# Change name setted from rclone config as name, default is backup
rcloneName=backup

# If set to true, logs from all runs of the service are preserved.
# If set to false, only logs since the last start of the service are
kept.
keepLogs=false

# ------------------------------ CONFIG END ------------------------
--------

runFile=/var/run/backup-service
logFile=/var/log/backup-service.log

# This needs to be set explicitly, because $HOME is not set when the
service is started by Batocera.
export RCLONE_CONFIG=/userdata/system/.config/rclone/rclone.conf

manual() {
    echo "> CONFIG sync" ----------------------------------------------
-------------------------------------
    rclone sync /userdata/system/batocera.conf
${rcloneName}:${destDir}/system --progress

    echo "> SAVES sync" -----------------------------------------------
-------------------------------------
    rclone sync /userdata/saves ${rcloneName}:${destDir}/saves --
exclude "flatpak/**" --progress
```

```
    echo "> ROMS sync" ----------------------------------------------
-------------------------------------
    rclone sync /userdata/roms ${rcloneName}:${destDir}/roms --exclude
"flatpak/**" --progress

    echo "> BIOS sync" ----------------------------------------------
-------------------------------------
    rclone sync /userdata/bios ${rcloneName}:${destDir}/bios --progress
}

start() {
    touch ${runFile}

    if [ "$keepLogs" = false ]; then
        rm -f ${logFile}
    fi

    timeStamp="$(date)"
    echo -e "\n--- SERVICE START - [${timeStamp}] ---\n" >>${logFile}

    while test -e ${runFile}; do
        sleep ${waitSeconds}
        rclone sync /userdata/saves ${rcloneName}:${destDir}/saves --
exclude "flatpak/**" --log-file ${logFile} --log-level INFO
    done
}

stop() {
    rm -f ${runFile}
}

status() {
    if test -e ${runFile}; then
        echo "Backup service is running. Logs: ${logFile}"
    else
        echo "Backup service is not running."
    fi
}

case "$1" in
    manual)
        manual
        ;;
    start)
        start &
        ;;
    stop)
        stop
        ;;
    status)
        status
```

```
        ;;
    *)
        echo "Usage:
https://gitlab.com/peterbozso/batocera-backup-service"
        ;;
esac
```

From:
https://wiki.batocera.org/ - **Batocera.linux - Wiki**

Permanent link:
https://wiki.batocera.org/rclone_cloudbackup

Last update: **2025/05/31 11:54**