

## Setting up Backblaze B2 on Linux

### CentOS 7

Below are the commands needed to install Rclone or Duplicity.

#### Rclone

These step by step instructions are from [rclone.org](https://rclone.org)

### Linux installation from precompiled binary

Fetch and unpack

```
curl -O https://downloads.rclone.org/rclone-current-linux-amd64.zip
unzip rclone-current-linux-amd64.zip
cd rclone-*-linux-amd64
```

Copy binary file

```
sudo cp rclone /usr/bin/
sudo chown root:root /usr/bin/rclone
sudo chmod 755 /usr/bin/rclone
```

Install manpage

```
sudo mkdir -p /usr/local/share/man/man1
sudo cp rclone.1 /usr/local/share/man/man1/
sudo mandb
```

Run `rclone config` to setup. See [rclone config docs](#) for more details.

```
rclone config
```

#### Duplicity

To install duplicity on CentOS, simply run the command below:

```
yum install duplicity
```

## Ubuntu

Below are the commands needed to install both Rclone and Duplicity.

### Rclone

These step by step instructions are from [rclone.org](https://rclone.org)

## Linux installation from precompiled binary

Fetch and unpack

```
curl -O https://downloads.rclone.org/rclone-current-linux-amd64.zip
unzip rclone-current-linux-amd64.zip
cd rclone-*-linux-amd64
```

Copy binary file

```
sudo cp rclone /usr/bin/
sudo chown root:root /usr/bin/rclone
sudo chmod 755 /usr/bin/rclone
```

Install manpage

```
sudo mkdir -p /usr/local/share/man/man1
sudo cp rclone.1 /usr/local/share/man/man1/
sudo mandb
```

Run `rclone config` to setup. See [rclone config docs](#) for more details.

```
rclone config
```

### Duplicity

Run the following command to install all necessary packages to run duplicity.

```
sudo apt-get install duplicity
```

## Debian

Below are the commands needed to install both Rclone and Duplicity.

## Rclone

These step by step instructions are from [rclone.org](https://rclone.org)

### Linux installation from precompiled binary

Fetch and unpack

```
curl -O https://downloads.rclone.org/rclone-current-linux-amd64.zip
unzip rclone-current-linux-amd64.zip
cd rclone-*-linux-amd64
```

Copy binary file

```
sudo cp rclone /usr/bin/
sudo chown root:root /usr/bin/rclone
sudo chmod 755 /usr/bin/rclone
```

Install manpage

```
sudo mkdir -p /usr/local/share/man/man1
sudo cp rclone.1 /usr/local/share/man/man1/
sudo mandb
```

Run `rclone config` to setup. See [rclone config docs](#) for more details.

```
rclone config
```

## Duplicity

Run the following commands to install all necessary packages for duplicity.

```
wget https://code.launchpad.net/duplicity/0.7-series/0.7.13.1/+download/duplicity-0.7.13.1.tar.gz
```

```
tar -xvzf duplicity-0.7.13.1.tar.gz
```

```
apt install python-pip
```

```
apt install librsync-dev
```

```
apt install python-dev
```

```
cd duplicity-0.7.13.1/
```

```
python setup.py install
```

## Using Rclone and Duplicity

### Rclone

Shown below is an example of the Config process and setting up Rclone to your B2 account

You will be prompted to answer a few questions, the answers are shown below:

```
root@Jail1:/rclone-v1.35-freebsd-amd64 # ./rclone config
No remotes found - make a new one
n) New remote
s) Set configuration password
q) Quit config
n/s/q>
```

→ Enter 'n' to create a new remote

You will then be prompted to enter a name → for example, we simply chose 'remote'

```
Type of storage to configure.
Choose a number from below, or type in your own value
1 / Amazon Drive
  \ "amazon cloud drive"
2 / Amazon S3 (also Dreamhost, Ceph, Minio)
  \ "s3"
3 / Backblaze B2
  \ "b2"
4 / Dropbox
  \ "dropbox"
5 / Encrypt/Decrypt a remote
  \ "crypt"
6 / Google Cloud Storage (this is not Google Drive)
  \ "google cloud storage"
7 / Google Drive
  \ "drive"
8 / Hubic
  \ "hubic"
9 / Local Disk
  \ "local"
10 / Microsoft OneDrive
  \ "onedrive"
11 / Openstack Swift (Rackspace Cloud Files, Memset Memstore, OVH)
  \ "swift"
12 / Yandex Disk
  \ "yandex"
Storage>
```

→ Enter '3' to configure Backblaze B2

Next you will be prompted for you Backblaze B2 Account ID and Application Key. To access this, login to your Backblaze B2 account and click on “Buckets” in the “B2 Cloud Storage” section. Click on the “Show Account ID and Application Key” link shown below.

Welcome rmacqueen, [Sign Out](#)

## B2 Cloud Storage Buckets

With Backblaze B2 Cloud Storage you can store data in the Backblaze Cloud. Any size, filetype or number of files.  
New to B2 Cloud Storage? Check out the [B2 Starter Guide](#).

[Show Account ID and Application Key](#)

- B2 Cloud Storage
- Buckets**
- [Browse Files](#)
- [Snapshots](#)
- [Reports](#)
- [Caps & Alerts](#)

After entering that, you will be prompted to enter an endpoint. It is recommended to leave this blank and just hit 'enter'.

You will then get to review your information and decide if you're satisfied or need to go back in and edit.

Configuration is now complete, and ready to use.

Sync is used to fully backup a whole directory.

Copy is used to only backup new/changed files.

This remote is called `remote` and can now be used like this

See all buckets

```
rclone lsd remote:
```

Make a new bucket

```
rclone mkdir remote:bucket
```

List the contents of a bucket

```
rclone ls remote:bucket
```

Sync `/home/local/directory` to the remote bucket, deleting any excess files in the bucket.

```
rclone sync /home/local/directory remote:bucket
```

## Duplicity

Now to actually use duplicity to send files up to B2:

```
duplicity [type (full or incremental)] [--no-encryption (if wanted)] [/directory to share]  
b2://account_id:[application_key]@bucket_name.
```

For example, in testing I ran the following command see screenshot:

```
root@Jail:/stuff # duplicity full --no-encryption /stuff/ b2://21b99f4d4820:0015b4dbc8a19641e5ba43ce1f1c1aedc0a737faa0@45d-duplicity

Local and Remote metadata are synchronized, no sync needed.
Last full backup left a partial set, restarting.
Last full backup date: Fri Mar  3 05:24:03 2017
RESTART: The first volume failed to upload before termination.
Restart is impossible...starting backup from beginning.
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: none
----- [ Backup Statistics ]-----
StartTime 1488547776.13 (Fri Mar  3 05:29:36 2017)
EndTime 1488548368.21 (Fri Mar  3 05:39:28 2017)
ElapsedTime 592.08 (9 minutes 52.08 seconds)
SourceFiles 3
SourceFileSize 2048000004 (1.91 GB)
NewFiles 3
NewFileSize 2048000004 (1.91 GB)
DeletedFiles 0
ChangedFiles 0
ChangedFileSize 0 (0 bytes)
ChangedDeltaSize 0 (0 bytes)
DeltaEntries 3
RawDeltaSize 2048000000 (1.91 GB)
TotalDestinationSizeChange 2053341932 (1.91 GB)
Errors 0
-----

root@Jail:/stuff #
```

User can login to B2 account and manually create the bucket beforehand, or they can just enter a new bucket name in the command and it will automatically create the bucket and send all files up to it.

## Creating Cron jobs to schedule syncing of data to the B2 Cloud

Below we will show how to create a cronjob. Cronjobs run commands in the background and will allow you to sync the contents of a desired directory to the B2 cloud without having to manually start the process every time.

Steps:

1. We need to create a script which the Cron daemon will call. This script will either have the rclone or duplicity command you want to run. To create the file, you'll need to run:  
`vim /root/scripts/rclone-data.sh`  
You can choose whatever directory you want to store your scripts in, we chose /root/scripts in this example. You can also use any other text editor in linux such as nano, vi, or vim.
2. Below is an example of a script using rclone to sync the /mnt/data directory.

```
#!/bin/bash
#This is an example script for the Cron daemon to call to sync the /mnt/data directory to the B2 cloud
rclone sync /mnt/data remote:centos-backup
```

Make sure that this scrip is executable. This is done by running:

```
chmod +x /path/to/script
```

3. To create our cronjob, run the following command to get into the editor:  
`crontab -e` This is bring you into the vim text editor, type 'i' to insert text.
4. Cron uses the following syntax for scheduling:

```
1 2 3 4 5 /path/to/script where:
```

1 = Minutes (0-59)

2 = Hour (0-23)

3 = Day (1-31)

4 = Month (1-12)

5 = Day of the week (0-6 ----- Sunday =0)

If you want the Cronjob to run every Friday night at 8:00PM it would look as follows:

```
00 20 * * 5 /path/to/script
```

5. Below is an example of a Cronjob:

```
35 10 30 06 * /root/scripts/rclone-data.sh
```

This calls the rclone-data.sh script at 10:35AM on June 30<sup>th</sup>.

Once done entering your text, hit 'Esc' then type ':wq' and hit enter to save.