

[Home](#) / [Server Administration](#) / Back up your Pi to your Google drive

## Raspberry Pi Networking

Connect and control multiple Raspberry Pi's. Sign up now



# Back up your Pi to your Google drive

Being able to back up data to the cloud is very useful. It means that even if your Raspberry Pi dies or your SD card gets corrupted, your data is still safe. It also means that you can access your data from any where in the world.

If you have a Google drive account, you can use the grive program to sync a folder on your Pi with your Google drive.

## Install grive

Start by making sure that your Pi's software is up to date, install some additional packages, and get the code using git:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install git cmake build-essential libgcrypt11-dev libjson0-dev
libcurl4-openssl-dev libexpat1-dev libboost-filesystem-dev libboost-program-
options-dev binutils-dev libboost-test-dev libqt4-dev libyajl-dev
```

Your message for visitors here [Learn more](#) [Close](#)

```
$ git clone git://github.com/Grive/grive.git
```

Modify /home/pi/grive/libgrive/src/drive/State.cc to resolve some compiler problems. Lines that use the Add() method need to contain a cast to the correct size of integer. The expression "(boost::uint64\_t)" must be added to lines 251, 252 and 256. The updated code should look like this:

```
void State::Write( const fs::path& filename ) const
{
```

```

    Json last_sync ;
    last_sync.Add( "sec", Json((boost::uint64_t)m_last_sync.Sec() ) );
    last_sync.Add( "nsec", Json((boost::uint64_t)m_last_sync.NanoSec() ) );

    Json result ;
    result.Add( "last_sync", last_sync ) ;
    result.Add( "change_stamp", Json((boost::uint64_t)m_cstamp) ) ;

    std::ofstream fs( filename.string().c_str() ) ;
    fs << result ;
}

```

Now you need to configure the make files by running cmake (note the '.' after the cmake command) and compile the source code by running make:

```

$ cd ./grive
$ cmake .
$ make

```

You need to create a directory where you can store files that you want to be synchronized with your Google drive, and copy the grive executable to that directory.

```

$ mkdir ~/google_drive
$ cp ./grive/grive ~/google_drive

```

The first time you execute grive, you need to use the -a option in order to authenticate with Google. You must be logged into your Google account from your Pi for this to work. If you're logged into your Google account on a PC or laptop, log out, and then log in again using a browser on your Pi. Once you've done this, change to your Google drive directory and run grive.

```

$ cd ../google_drive/
$ ./grive -a

```

A link will be printed in the terminal window. Copy the link and paste it in a browser. A page on google.com will appear with a long code that you need to enter in the terminal window. Pasting the code didn't work for me, so you may have to type it in. The authentication process will then take place, and your Pi will sync for the first time. **Before you do this, check to see how much space you've used up on your Google drive, and check to see if there's enough space on your Pi's SD card.**

## Running a backup to the cloud

If you've got this far, you should now be able to sync files and folders in `/home/pi/google_drive` with your Google drive. The next step is to backup the data on your Pi and upload it to your Google drive.

The following bash script creates a tar archive containing my home directory. I've made sure not to include the Google drive directory in this archive, otherwise previous archives would be included in the backup. The second line compresses the archive into a `.tar.gz` file. The file is moved to the google drive directory before running grive to upload the new archive to your Google drive on the internet. I've used the date command to embed the current date in the file name.

```
#!/bin/bash

tar -crvf backup_$(date +%y.%m.%d).tar /home/pi --exclude="/home/pi/google_drive"
gzip backup_$(date +%y.%m.%d).tar
mv backup_$(date +%y.%m.%d).tar.gz ./google_drive
cd ./google_drive
./grive
cd ..
```

Save this code as `grive_backup.sh`, and make it executable using the `chmod +x` command.

When you run the script, it may appear to hang while `gzip` is running. This may take a few minutes depending on how much data there is in your home directory.

## Running a backup script as cron job

This script can easily be automated by creating an entry in the crontab file. Linux uses the crontab file to schedule tasks. Entries are made in the format

```
minute hour day_of_month month day_of_week username command
```

Use this command to open crontab:

```
$ sudo leafpad /etc/crontab
```

...and then add this line:

```
00 03 * * * pi /home/pi/grive_backup.sh
```

This specifies that you want to run `grive_backup.sh` at 3.00am every day, and it should be run as user `pi`. Linux should automatically detect that your crontab file has been modified, so there's no need to reboot.

Your SD card will fill up after a few days or weeks, so it's best to remove old backups. You can do this by adding the following line to the script above:

```
rm backup_$(date --date="7 days ago" +%y.%m.%d).tar.gz
```

This time the date command is being used to create a date from 7 days ago, so a back up file from 7 days ago will be deleted.

[9 Comments](#)

## More from this category:

### [Server Administration](#)

- [Connect your Raspberry Pi to a USB hard disk](#)
- [Change your Raspberry Pi's hostname](#)
- [Share your Raspberry Pi's files and folders across a network](#)
- [Back up your Pi to your Google drive](#)
- [Get your Raspberry Pi web site on-line](#)
- [Run a script on start up](#)

## Comments

9 Comments

Raspberry Web Server

 Login ▾

 Recommend

 Share

Sort by Best ▾



Join the discussion...



**Dominic Afonso** • 4 months ago

Thanks for the great walk through. I had to run the cmake . and make commands a couple of times as the make command resulted in some errors at first, so I just tried again. With it all up and running I now have a daily backup of an sql database syncd with my google drive. on the downside, i now have the entire contents of my google drive on my pi. is there a way to set a single folder on google drive as the sync location?

1 ^ | ▾ • Reply • Share >



**James Prankard** • 5 months ago

Thanks for this.

Got this running on my PI with a local git server and gitlist browser. And now it backs all the git repositories to google drive for me :)

1 ^ | ▾ • Reply • Share >



**Anders Jackson** • 2 years ago

Greate post on a important subject, backup.

You might want to consider to save one backup from each month in you archive on Google so you can retrieve older data if you don't miss the removed data within one week. You can manually remove those monthly backups when running low on space so you only have one yearly backup of those old data.

Encrypt those backups could also be useful, but that is another story.:-)

You might also want to make a .tar.gz (or .tgz) file directly from tar. You can do that by adding the switch -z to the tar command. Use something like this:

```
tar -czrvf backup.tgz /home/pi --exclude="..."
```

If you want to know more about tar command, you can use the man command. Don't forget the -k switch to man.

1 ^ | v • Reply • Share >



**sanjibm** • 3 months ago

Very useful post, thanks very much

^ | v • Reply • Share >



**Luke Taylor** • 4 months ago

Can this be done on an external flash drive?

^ | v • Reply • Share >



**Luke Taylor** → Luke Taylor • 4 months ago

I don't have enough space on my pi, and I'd like to have a flash drive that functions as a google drive folder on my pi

^ | v • Reply • Share >



**Steinar** • a year ago

I guess that the correct should be -cvf and not crvf? can we combine c and r?

i get this error

tar: You may not specify more than one '-Acdrux' or '--test-label' option

^ | v • Reply • Share >



**PyPlate** Mod → Steinar • a year ago

Well spotted, yes that should be -cvf. I'll update the page when I get a chance.

^ | v • Reply • Share >



**PyPlate** Mod • 2 years ago

You're right, using the -z option would have been easier than using the gzip command on a separate line.

^ | v • Reply • Share >

#### ALSO ON RASPBERRY WEB SERVER

WHAT'S THIS?

### Raspberry Pi web server - Building an SQLite temperature logger

36 comments • 2 years ago



**Peter Subkowski** — Hey, thanks for your site, it really taught me a lot. However, one thing I can't figure out. When I reboot the Pi, ...

### Connect your Raspberry Pi to a USB hard disk

20 comments • 2 years ago



**douglourey** — I use LABEL in the fstab file to automatically mount my external hard disk drive and usb flash drives because I find it ...

### Building a web user interface for the temperature monitor

50 comments • 2 years ago



**baja** — for having local time just add after now, localtime: datetime('now', 'localtime')

### Raspberry Pi web server - Get your Raspberry Pi web site on-line

6 comments • 2 years ago



**PyPlate** — You need to set the server name in each virtual host file (see <http://linuxwebservers.net/apa...>) The name ...

# Raspberry Pi Case / Box

Best-selling protective case/ box for model A and B by SB Components



Follow me



## Home

[Home](#)

[Specs](#)

[Set Up](#)

[GPIO](#)

## Linux basics

[What is Linux?](#)

[Files and directories in Linux](#)

[Users and Permissions](#)

## Bash

[Bash basics](#)

## Raspberry Pi Cluster

[Raspberry Pi Cluster](#)

[Raspberry Pi Server Cluster Tests](#)

[Improving cluster performance by tuning Apache](#)

[On Site Optimization](#)

[Adding more nodes to the cluster](#)

[Comparing the performance of Nginx and Apache web servers](#)

## HTML and CSS

HTML Basics

Making a complete HTML page

CSS - Cascading Style Sheets

## CGI scripting

Writing CGI scripts on a Raspberry Pi

Writing CGI scripts in Python

Using Google charts in Python CGI scripts

Sending data to an HTTP server - get and post methods

Web Forms with Python

Using Python to set, retrieve and clear cookies

Setting up Nginx and uWSGI for CGI scripting

## Raspberry Pi Temperature Logger

Building an SQLite temperature logger

Building a web user interface for the temperature monitor

## SQL Databases

Set up an SQLite database on a Raspberry Pi

Accessing an SQLite database with Python

Using MySQL on a Raspberry Pi

## Server Administration

Connect your Raspberry Pi to a USB hard disk

Change your Raspberry Pi's hostname

Share your Raspberry Pi's files and folders across a network

Back up your Pi to your Google drive

Get your Raspberry Pi web site on-line

Run a script on start up

## GPIO

Raspberry Pi GPIO

Controlling GPIO with wiringPi

Connecting your Pi's GPIO to a breadboard

Detecting a button press through GPIO

Using Interrupt Driven GPIO

Use GPIO to turn on an LED

Using Python to control GPIO pins

Connecting a temperature sensor to GPIO

## Pi Face

Pi Face

Building a remote control vehicle using a Raspberry Pi

## Pyplate

Install Pyplate on a Raspberry Pi with Apache



Copyright © 2014 Steve Breuning

Raspberry Pi is a trademark of the [Raspberry Pi Foundation](#). This site is not affiliated with the Raspberry Pi Foundation.

This site is powered by [PyPlate CMS](#).