yewkeo, 123RF

Exploring Grsync, Back In Time, and Time Drive

STAYING SAFE

The best time to worry about a backup solution is before you need it.

Read on for a roundup of some easy backup tools. **BY DMITRI POPOV**

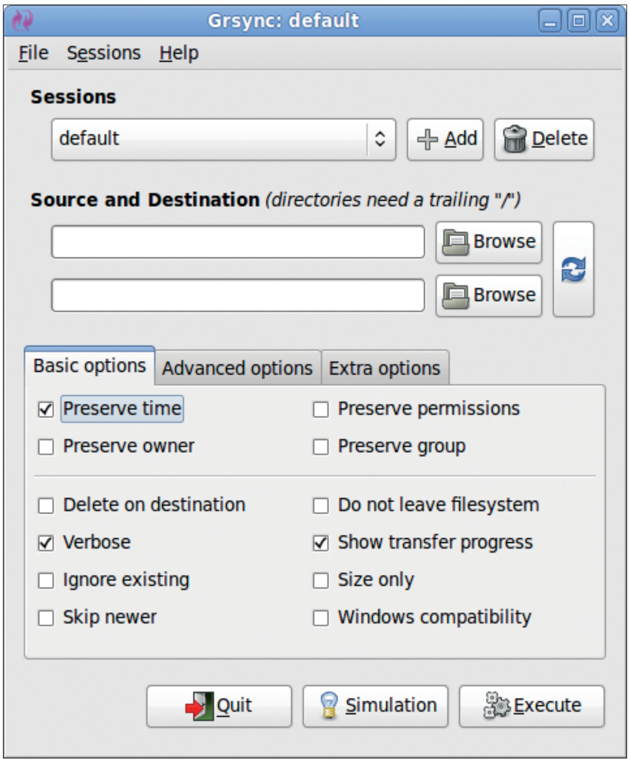
If you want to keep your data safe, you cannot avoid installing and configuring a backup application. Choosing the right tool for the job is crucial if you want the backup process to be painless and foolproof. In the previous issue, Kristian Kissling described the SBackup desktop backup utility. In this article, I'll look at a few easy alternatives: Grsync, Back In Time, and Time Drive.

Grsync

Grsync [1] is not exactly a standalone backup application; it is a graphical interface to the powerful command-line backup utility called rsync. With rsync, you can create advanced backup scripts, but if fiddling with parameters and switches is not your cup of tea, then Grsync might be the tool for you. Thanks to its user-friendly interface (Figure 1), you can create sophisticated backup profiles by selecting and ticking the desired options and checkboxes. Grsync is available in Ubuntu's software repository, so installing it is a matter of running the *sudo apt-get install grsync* command in the Terminal or opening the Ubuntu Software Center and clicking *Install* after searching on *Grsync*. Then launch

Grsync by choosing *Applications* | *System Tools* | *Grsync*.

Grsync can handle multiple backup jobs, which can come in handy when you need to run separate backup operations. For example, you might want to create separate backup sessions for backing up your documents to a remote server every hour and saving your photos to a removable storage device twice a day. To create a new backup session, press the *Add* button in the *Sessions* section and specify the source and destination directories.



Sessions

default

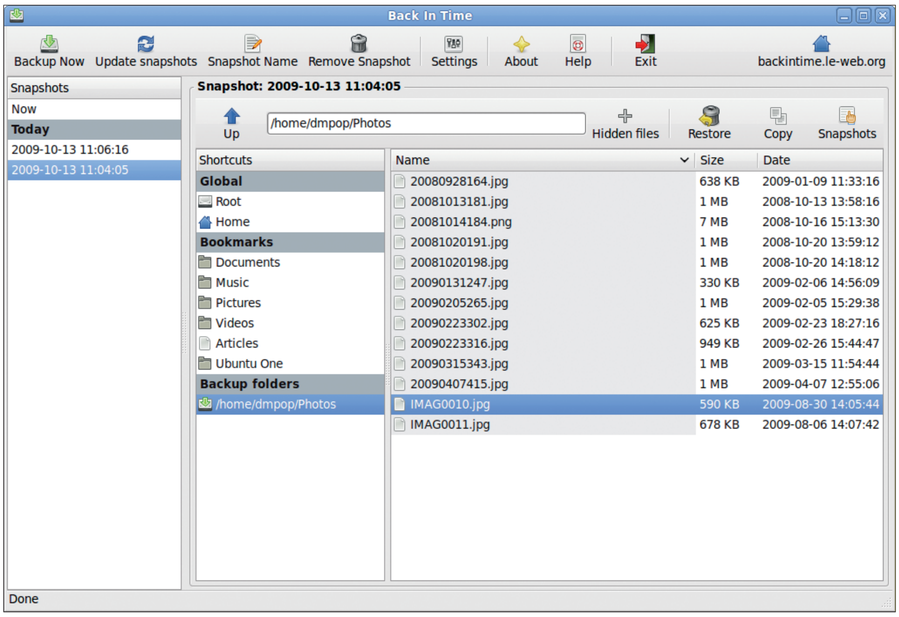
Source and Destination (directories need a trailing "/")

☑ Preserve time
□ Preserve permissions
□ Preserve owner
□ Preserve group
□ Delete on destination
□ Do not leave filesystem
☑ Verbose
☑ Show transfer progress
□ Ignore existing
□ Size only
□ Skip newer
□ Windows compatibility

Figure 1: Grsync provides a user-friendly interface to rsync.

The three tabs at the bottom of the window – *Basic options*, *Advanced options*, and *Extra options* – let you configure the backup session by specifying the desired settings. Which options you want to enable depends, of course, on your needs, but you might want to tick a few key checkboxes in most situations. For example, to enable detailed output, which can help you monitor and troubleshoot the backup operation, tick the *Verbose* checkbox. If you want the backup operation to delete files in the destination directory that don't exist any longer in the source directory, enabling the *Delete on destination* option will do the trick. The *Windows compatibility* option makes rsync play nice with FAT-formatted devices, which can be useful if you want to back up files to a USB stick or a FAT partition.

The *Advanced options* section offers even more ways to fine-tune the backup session. By default, rsync compares files in the source and destination directories by size and file modification time. However, you can force rsync to perform the comparison with the use of checksum values by ticking the *Always checksum* checkbox. As you might guess, the *Compress file data* option forces rsync to compress files during the transfer, which can be useful when the session uses a directory in a locally mounted remote file system as its destination.



Snapshots
Now
Today
2009-10-13 11:06:16
2009-10-13 11:04:05

Snapshot: 2009-10-13 11:04:05

/home/dmpop/Photos

Name	Size	Date
20080928164.jpg	638 KB	2009-01-09 11:33:16
20081013181.jpg	1 MB	2008-10-13 13:58:16
20081014184.png	7 MB	2008-10-16 15:13:30
20081020191.jpg	1 MB	2008-10-20 13:59:12
20081020198.jpg	1 MB	2008-10-20 14:18:12
20090131247.jpg	330 KB	2009-02-06 14:56:09
20090205265.jpg	1 MB	2009-02-05 15:29:38
20090223302.jpg	625 KB	2009-02-23 18:27:16
20090223316.jpg	949 KB	2009-02-26 15:44:47
20090315343.jpg	1 MB	2009-03-15 11:54:44
20090407415.jpg	1 MB	2009-04-07 12:55:06
IMAG0010.jpg	590 KB	2009-08-30 14:05:44
IMAG0011.jpg	678 KB	2009-08-06 14:07:42

Figure 2: Back In Time uses snapshots to keep your data safe.

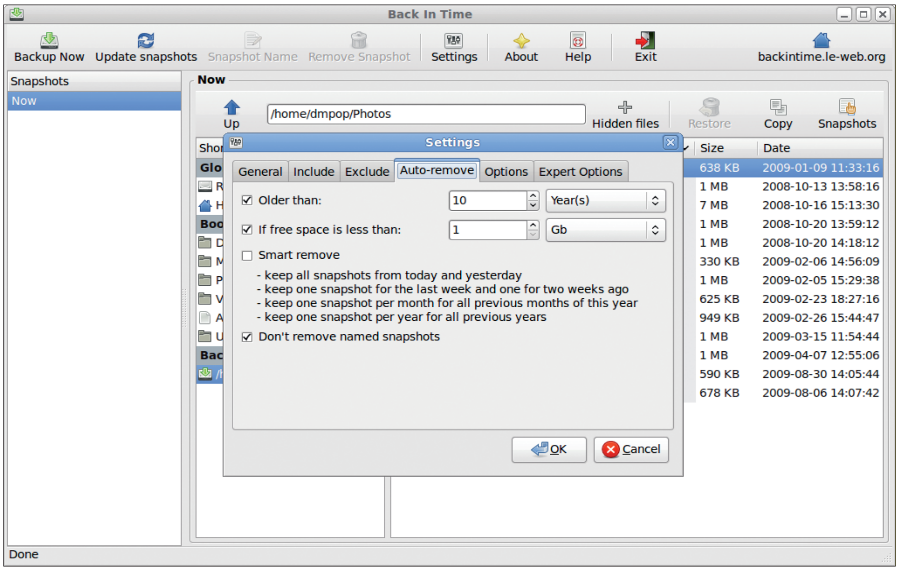
Finally, the *Extra options* section allows you to specify scripts that Grsync should run before and after the backup operation. For example, Grsync can run a script that mounts a remote file system containing the destination directory at the start and then run the command to shut down the system when the backup is finished.

Once you're satisfied with the settings, your backup is ready to go, but don't press the *Execute* button just yet. To make sure that everything works as it should, Grsync allows you to perform a dry run by pressing the *Simulate* button. After pressing the button, take a close look at the output generated by Grsync. If everything looks good, you can press *Execute* to perform the backup. Restoring data with Grsync is pretty straightforward. In fact, to perform a restore, you just make Grsync run the very same session in reverse. To do this, simply press the button that switches source and destination directories and hit *Execute*. Grsync also allows you to export and import sessions via the *Sessions* | *Export* and *Sessions* | *Import* commands, which means you can use this feature to share your backup sessions with other users, as well as to transfer sessions between different machines.

Back In Time

Back In Time [2] works in a rather different manner from traditional backup applications like Grsync. Instead of keeping an exact copy of a source directory in a destination, Back In Time creates multiple snapshots of one or several directories, and each snapshot stores the entire contents of the directories. Although it might sound like a waste of resources, this approach offers a more flexible way to keep your data safe.

If you use Grsync to perform backups on a weekly basis, you can only get the version of any file that is one week old. But what if you need to get a version of the file from, say, 10 days ago? Here is



Now

/home/dmpop/Photos

Settings – Auto-remove

☑ Older than: 10 Year(s)

☑ If free space is less than: 1 Gb

☐ Smart remove

- keep all snapshots from today and yesterday
- keep one snapshot for the last week and one for two weeks ago
- keep one snapshot per month for all previous months of this year
- keep one snapshot per year for all previous years

☑ Don't remove named snapshots

Size	Date
638 KB	2009-01-09 11:33:16
1 MB	2008-10-13 13:58:16
7 MB	2008-10-16 15:13:30
1 MB	2008-10-20 13:59:12
1 MB	2008-10-20 14:18:12
330 KB	2009-02-06 14:56:09
1 MB	2009-02-05 15:29:38
625 KB	2009-02-23 18:27:16
949 KB	2009-02-26 15:44:47
1 MB	2009-03-15 11:54:44
1 MB	2009-04-07 12:55:06
590 KB	2009-08-30 14:05:44
678 KB	2009-08-06 14:07:42

Figure 3: You can specify the way Back In Time manages old snapshots.

where Back In Time comes into the picture. It saves every version of the file in a separate snapshot, so you can locate and recover whatever version you need (Figure 2). You can think of this feature as an Undo command in your word processor or image editor. At any time, you can “undo” changes made to a document or a photo by recovering a specific version of the file from a snapshot.

Although you can download a *.deb* package from the project’s website or install via Ubuntu’s Software Center, a better way to install Back In Time on your system is through the project’s repository. This ensures that you always have the latest version of the application. To begin, add the project’s repository to the *sources.list* file by opening it in your favorite text editor (e.g., Gedit) with the following command:

```
sudo gedit /etc/apt/sources.list
```

Then add the following line at the end of the file:

```
deb http://le-web.org/repository ↩
  stable main
```

Save and close the file, open the terminal, and run the following commands:

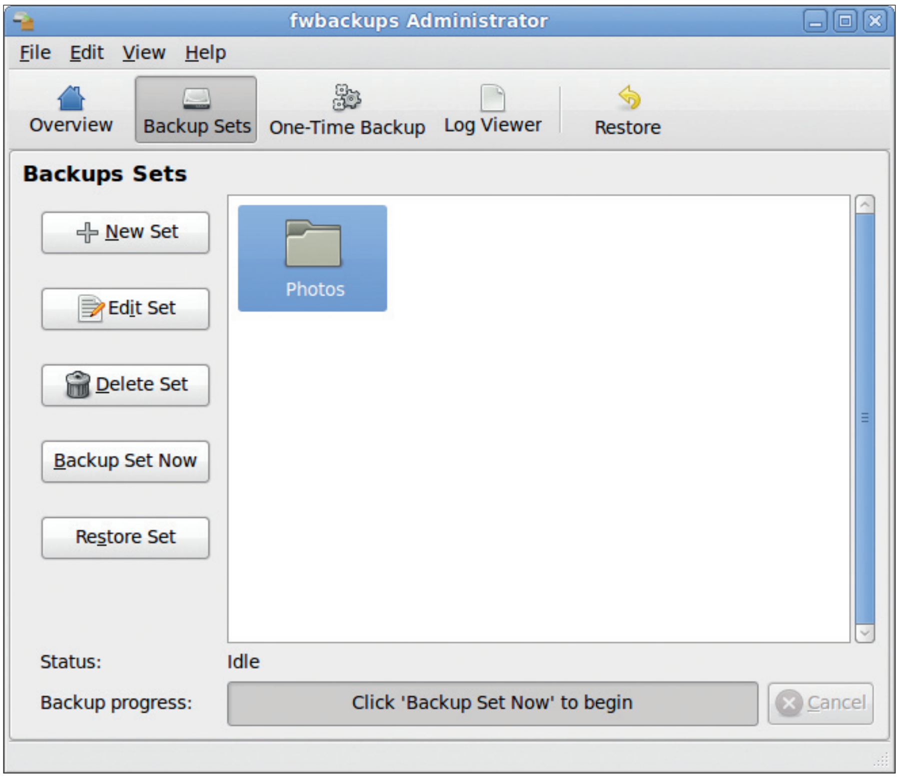
```
wget http://le-web.org/repository/↩
  le-web.key
sudo apt-key add le-web.key
sudo apt-get update
sudo apt-get install ↩
  backintime-common backintime-gnome
```

If you are using Kubuntu, then replace the last command with this one:

```
sudo apt-get install ↩
  backintime-common backintime-kde4
```

Once the Back In Time application is installed, you can launch it by choosing *Applications* | *System Tools* | *Back In Time*. On the first run, the application will open the Settings dialog window (Figure 3), where you can configure the backup job. In the *General* section, choose the directory for storing snapshots and enable optional scheduling so the application can take snapshots at specified time intervals. Of course, you can leave the scheduling feature disabled if you prefer to create snapshots when you need them. The *Include* section lets you add the directories you want to back up. If you want to exclude certain files from the backup, you can do so by specifying a pattern in the *Exclude* section. For example, if you don’t want to include PNG graphics files, you can add the following pattern: **.png* or **.PNG*.

Because each snapshot stores the contents of all included directories, you need all the hard disk space you can get. For example, if the overall size of all directories included in the backup job is 5GB, you can store only 30 snapshots on a 150GB hard disk. To avoid running out of disk space, you must configure how Back In Time manages older snapshots, which can be done in the *Auto-remove* section. Here, you can set up Back In Time to delete snapshots older than a specified age (e.g., older than two weeks or three months). Alternatively, you can enable the feature that removes old snapshots when the free space on the hard disk reaches a certain limit. Also, you can let Back In Time handle the way it removes old snapshots by ticking the



Status: Idle

Backup progress: Click 'Backup Set Now' to begin

Figure 4: The fwbackups Administrator start window.

Smart remove checkbox. And, to be on the safe side, you can enable the *Don't remove named snapshots* feature, so Back In Time skips snapshots with custom names.

Operating Back In Time is as easy as it gets. If you enabled scheduling, the application will quietly create snapshots at the predefined intervals. Otherwise, you can create a snapshot manually by pressing the *Backup Now* button. The list of all snapshots will appear in the *Snapshots* pane of Back In Time's main window. To view the contents of a snapshot, select it in the list, and you will see the list of files in the right-hand pane (see Figure 2). To restore a specific file, select it in the desired snapshot and hit the *Restore* button, or you can rename the snapshot with the *Snapshot Name* button or delete any existing snapshots with the *Remove Snapshot* button.

fwbackups

If neither Grsync nor Back In Time cut it for you, try fwbackups [3]. This backup application strikes a perfect balance between feature set and ease of use.

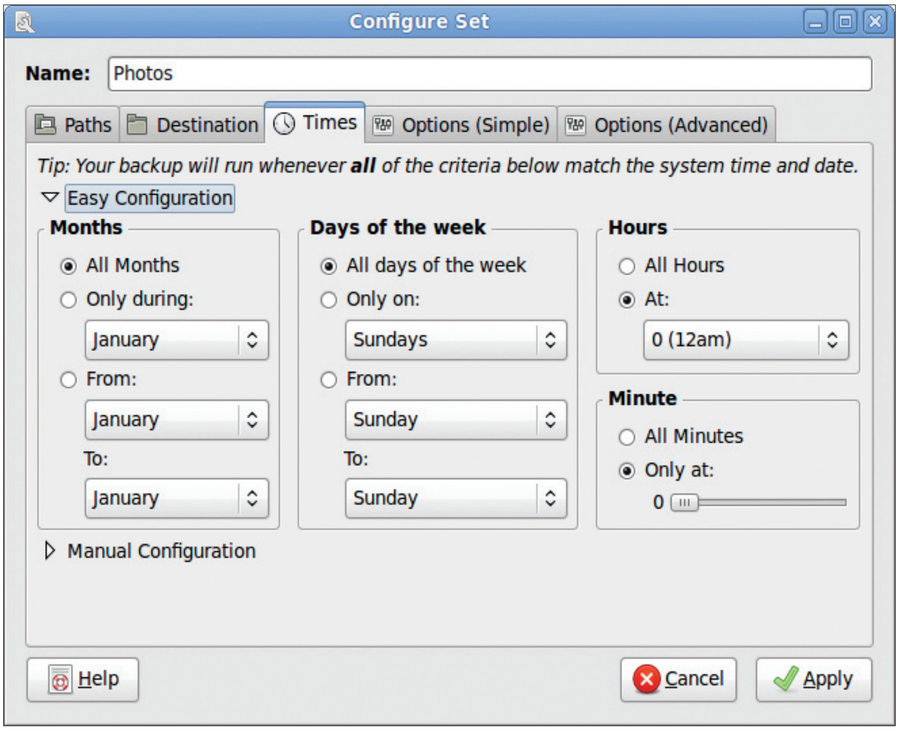
Because fwbackups is not available as a *.deb* package, it must be compiled manually. Before you do this, however, you should install several packages that fwbackups needs to function properly. This includes gettext, Python, PyGTK, pycrypto, and paramiko, as well as the optional libnotify package. To install all these packages, enter:

```
sudo apt-get install gettext python ➘
   python-gtk2 python-crypto ➘
   python-paramiko
```

To be able to compile software on your Ubuntu system, you must also install the build-essential package with *sudo apt-get install build-essential*. Once you've done that, download the latest .*tar.bz2* archive from the project's website and unpack it. In the terminal, switch to the resulting directory and run the following commands:

```
./configure
make
sudo make install
```

That's it. Now you can launch the application from the command line with *fwbackups*.



Configure Set

Name: Photos

Paths | Destination | Times | Options (Simple) | Options (Advanced)

*Tip: Your backup will run whenever **all** of the criteria below match the system time and date.*

▽ Easy Configuration

Months
● All Months
○ Only during: January
○ From: January
To: January

Days of the week
● All days of the week
○ Only on: Sundays
○ From: Sunday
To: Sunday

Hours
○ All Hours
● At: 0 (12am)

Minute
○ All Minutes
● Only at: 0

▷ Manual Configuration

Figure 5: fwbackups makes it easy to specify a backup schedule.

The next thing to do is create a new backup set. To begin, press the *Backup Sets* button (Figure 4) in the main toolbar and hit *New Set*. In the *Paths* tab of the Configure Set window, add the directories and individual files you want to back up and switch to the *Destination* tab. Here you have to specify the path to a directory for storing backups. Besides local folders, fwbackups can connect to remote directories via SSH. This can come in handy if you want to back up your files on a server or another Linux machine on the network.

The *Times* tab (Figure 5) lets you specify a schedule for the backup set. To specify a schedule, use the *Easy Configuration* panel and select the appropriate options, or you can do this by hand in the *Manual Configuration* panel. The latter requires a working knowledge of crontab syntax. In the *Options (Simple)* tab, you can configure other useful settings. For example, you can specify the number of old backups the applications should keep. Here, you can also enable the archiving feature, which saves backups as tar, gzip, or bzip archives.

As in Grsync, you can specify the commands you want fwbackups to run before and after the backup operation, which can be done in the *Options (Advanced)* section. Creating a backup set makes sense if you want to back up your data regularly. But fwbackups also allows you to run an ad hoc backup operation from the *One-Time Backup* button (see Figure 4).

Restoring data with fwbackups is not difficult either. Hit the *Restore* button then choose the target directory, source type, and a backup set from the appropriate drop-down lists. If you configured fwbackups to keep multiple backups, you can select the one you want from the *Date* drop-down list. Pressing the *Start Restore* button restores the data.

Final Word

A good backup application can save your bacon one day, so do yourself a favor and give the described tools a try. And don't put it off until your hard drive starts making strange noises – it might be too late by then. Although it might take a few tries to figure out the best backup configuration for your particular setup, once you've done it, you won't have to worry about losing your precious data. ■

INFO

[1] Grsync: *http://www.opbyte.it/grsync*

[2] Back In Time: *http://backintime.le-web.org*

[3] fwbackups: *http://www.diffingo.com/oss/fwbackups*