rdiff-backup

If you care about your data, you must back it up. The best backup scheme is one that is always on, and that happens without human interaction.

There are many tools for backing up systems, some with centralised management, others with fancy user interfaces, and others that are designed to be simple, easy to configure, and easy to use. The rdiff-backup tool falls into the latter category.

In this article, you will learn to:

1. Install rdiff-backup
2. rdiff-backup –exclude-globbing-filename $HOME/.excludes $HOME /path/to/drive
3. Automate rdiff-backup with cron
4. Restore from a backup

Requirements

To backup your machine properly, you should have:

- A client machine you want to backup, capable of running rdiff-backup
- A server to receive backup data, capable of running rdiff-backup
- Passwordless SSH access to your server

If you do not have a server that you can use as a backup server, you can just backup to an external or networked hard drive. It's not off-site, so it's not ideal, but it's better than not backing up at all.

Install

The rdiff-backup project is hosted at http://rdiff-backup.nongnu.org/ but the easiest way to install it is from SlackBuilds.org

You must install rdiff-backup on *both* the client computer being backed up and the server to which it is sending backup files. This means you are preforming *two* installs of rdiff-backup.

The server to which you backup does not need to also be running Slackware, as long as rdiff-backup is available for it, it will work as an rdiff-backup server.

To install rdiff-backup:

1. Install its dependency, librsync, from https://slackbuilds.org/result/?search=librsync
2. Install rdiff-backup from https://slackbuilds.org/result/?search=rdiff-backup

Configure

Unlike many popular UNIX backup applications, rdiff-backup requires no configuration. It is ready to use immediately after installation.

Backing up your data to a remote server

With rdiff-backup installed on both the client machine and the destination server, and assuming that your server is using SSH keys for access, start your first backup:

```
$ rdiff-backup -v 8 $HOME username@your-server.com::/path/to/backup/dir
```

- **username** on the server must be a valid user name for that server (this might be different from your local user name).
- **dir** on the server must already exist, and it should be a safe place meant to contain all of your user's backup files.

The initial backup takes longer than subsequent backups, because it is generating backups for your entire home directory. In future backups, only diffs are generated.

Backing up to a local drive

If you do not have access to any off-site server for your backups, then you should still back up using an external hard drive.

Your hard drive must be formatted such that you have read and write permissions. Ideally, it should be formatted with a good open source file system, such as EXT4, JFS, or UDF.

To start your first backup, attach your hard drive and mount it. Create a destination directory on the hard drive to hold your backup files.

Start your backup:

```
$ rdiff-backup -v 8 $HOME /path/to/backup/dir
```

The initial backup takes longer than subsequent backups, because it is generating backups for your entire home directory. In future backups, only diffs are generated.

Backing up a single file

Sometimes you might want to just backup a single file:

```
$ rdiff-backup --include $HOME/foo.txt --exclude '**' $HOME username@your-
server.com::/path/to/backup/dir
```

Excludes

There are often things in your home directory that aren't important enough to back up. You can (and arguably should) exclude these files from being backed up.

Create a file to contain a list of your excluded locations. Open the file in a text editor and list each directory or file you want excluded from the rdiff-backup process. Such a file is easy to create: you can just echo the path of a directory or file into, for instance, **~/.excludes**. If the file does not exist, the file will be created for you, and the entry added.

```
$ echo '$HOME/.local/share/Trash' >> ~/.excludes
```

As you find more files that you want to exclude, use the same command.

$ echo '$HOME/downloads' » ~/.excludes $ echo '$HOME/tmp' » ~/.excludes

Add the file containing your excludes to your rdiff-backup command. For instance:

```
rdiff-backup --exclude-globbing-filelist \
$HOME/.excludes \
$HOME \
username@your-server.com::/path/to/dir \
```

Automating backups

Although you can backup manually with the rdiff-backup command, it's important to automate backups to buffer against human forgetfulness. Automate rdiff-backup with cron.

For example, to run a backup once a day at 23:00:

```
0 23 * * * rdiff-backup --exclude-globbing-filelist $HOME/.excludes $HOME
username@your-server.com::/path/to/dir
```

To run backups every other hour only during 9:00 to 17:00 (business hours):

```
0 9,11,13,15,17 * * * rdiff-backup --exclude-globbing-filelist
$HOME/.excludes $HOME username@your-server.com::/path/to/dir
```

And so on. If you are unclear, there are several good examples online, probably for the exact schedule you are planning.

Recovering lost data

You should periodically verify your backups to make sure that what you *think* is happening is actually happening. The easiest way to do this is to recover a file.

Create a test file called ~/back.up

```
$ date >> ~/back.up
```

Let the file get backed up by your backup script.

Once your script has run, remove the file and attempt to restore it.

Restore the most-recently backed-up version of a file:

```
$ rdiff-backup --restore-as-of now username@your-
server.com::/path/to/dir/back.up $HOME/back.up
```

If you are using a hard drive for backups, use the path to the hard drive:

```
$ rdiff-backup --restore-as-of now /path/to/dir/back.up $HOME/back.up
```

Continue to check the health of your backups periodically. For instance, to recover a file from 3 days ago:

```
$ rdiff-backup --restore-as-of 3D /path/to/dir/back.up $HOME/back.up
```

Other acceptable time formats include 21m34s (21 minutes and 34 seconds ago) and full dates like 2017-01-11.

For more options, see the TIME FORMATS section of the rdiff-backup man page.

Incremental recovery

If you know even more detail about your file, you can use rdiff-backup to restore directly from an incremental backup file.

Increment files are stored in your backup destination (your server or backup drive) in the location **rdiff-backup-data/increments**

Incremental backups hold previous versions of changed files. You can specify one directly to get an exact iteration of a file:

```
$ rdiff-backup /path/to/dir/rdiff-backup-
data/increments/craft.2017-10-31T18:06:06-01:00.diff.gz $HOME
```

Sources

* Originally written by User klaatu

howtos, template