

A markdown file summarizing my current approaches for creating 3-2-1 compliant backups of my local and cloud infrastructure including Linux desktop, VMs, and cloud-hosted apps.

Edit

Manage topics

6 commits

1 branch

0 packages

0 releases

1 contributor

Branch: master


New pull request

Create new file


Upload files

Find file

Clone or download


 danielrosehilljlm Version in Readme header for greater visibility

Latest commit 33ab8f6 6 days ago

 images

Master backup strategy repository homepage Readme text update

6 days ago

 README.md

Version in Readme header for greater visibility

6 days ago

README.md

Master Backup Strategy (V1.2)

By: Daniel Rosehill (github@danielrosehill.co.il)

Version control: V1.2 (Updated: 01/05/20)

This "master" backup strategy summarizes the overall backup strategy that I currently use to back up my local and cloud data in compliance with the 3-2-1 backup approach:

Objective: 3-2-1 Compliant Backups



UBUNTU DESKTOP BACKUPS 3-2-1



PRIMARY SSD



BACKUP SSD 1

CLONEZILLA
ONSITE / MONTHLY



BACKUP SSD 2

TIMESHIFT
ONSITE/DAILY



EXTERNAL SSD

CLONEZILLA
ONSITE/MONTHLY



AWS S3

CLI
ONSITE/YEARLY

DANIEL ROSEHILL

- 3 extant copies of all critical data
- 2 backup copies -> Copies on different storage media
- 1 copy stored off-site

1: Local Backups

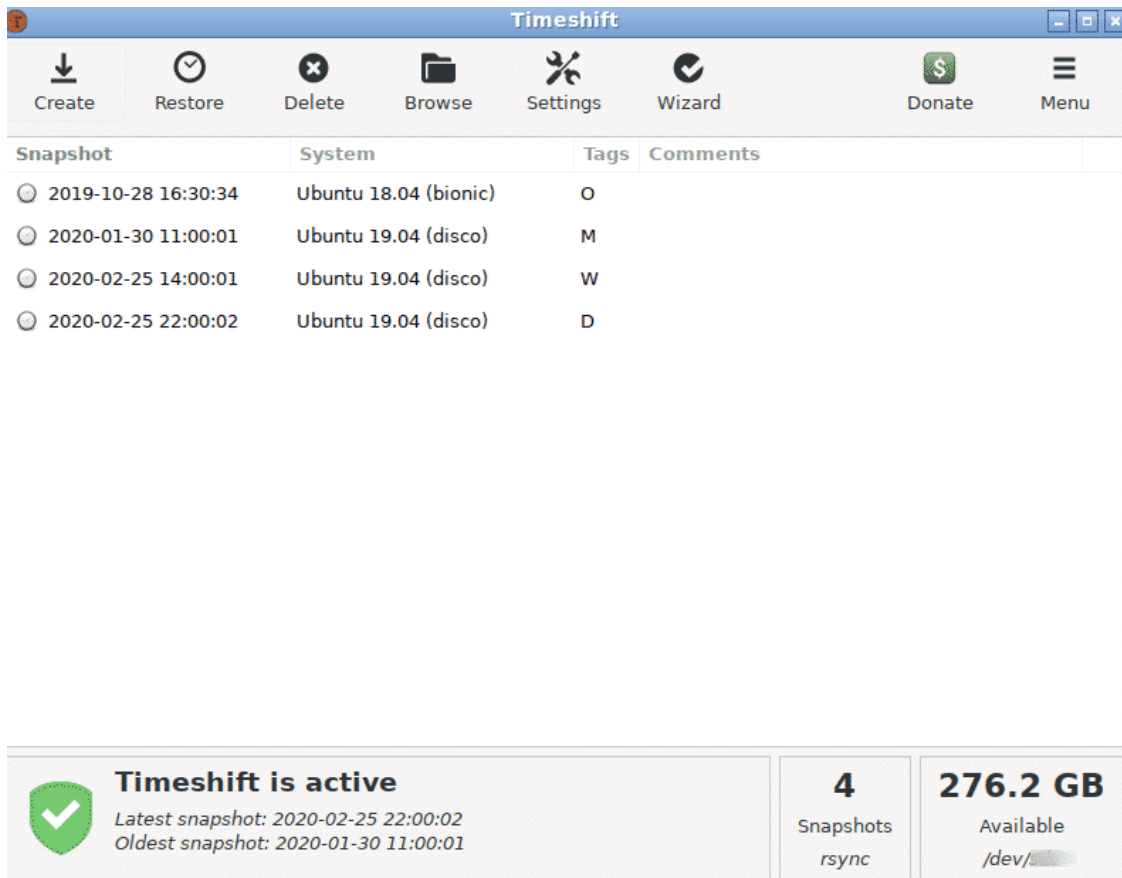
I summarized my approach for backing up my local (Linux) desktop [on LinuxHint.com](https://linuxhint.com).

https://github.com/danielrosehilljlm/Master_Backup_Strategy

1/5

The components are as follows:

1. Timeshift backup



I take the following restore points:

- Daily (x1)
- Weekly (x1)
- Monthly

These restore points are saved onto a *dedicated* 480 GB SSD within my desktop. [Timeshift](#) has time and again proven indispensable in rolling the system back to a point in time before changes — typically updates — rendered it unstable or prove some key system like package management.

Timeshift can be used to restore the system over a CLI.

2. Clonezilla



Because Timeshift requires that GRUB be intact, it cannot be relied upon to restore a completely bricked Linux system.

Therefore I also use [Clonezilla](#) to take disk <-> disk images.

I use another dedicated SSD for this purpose — although there is no reason one couldn't use a very generously sized HDD and create separate partitions for the Clonezilla and Timeshift backups (although this creates more redundancy on storage media).

I run Clonezilla as often as I remember. Approximately once every 3 months. I have yet to have to rely upon this for restore.

Notes:

- Because full system backups capture all virtual machines (VMs) nested within the home directory (if you're using VMWare Workstation player at /home/\$user/vmware) there is no need to create separate backups for VMs — although for convenience's sake (to be able to restore a VM without having to restore the overlying system), I take periodic backups of my Windows VM too.
- Because Clonezilla backup images of a whole disk are heavy, I have only ever uploaded backed up to S3 once. In the event of a disk failure and replacement, restoring from local media would make much more sense.
- This is obviously desktop-centric. A concerned traveller could bring an external SSD with him/her while travelling with a Clonezilla image of the SSD in order to replace it.

2: Cloud Backups: Major

I divide my cloud backup approach into two parts:

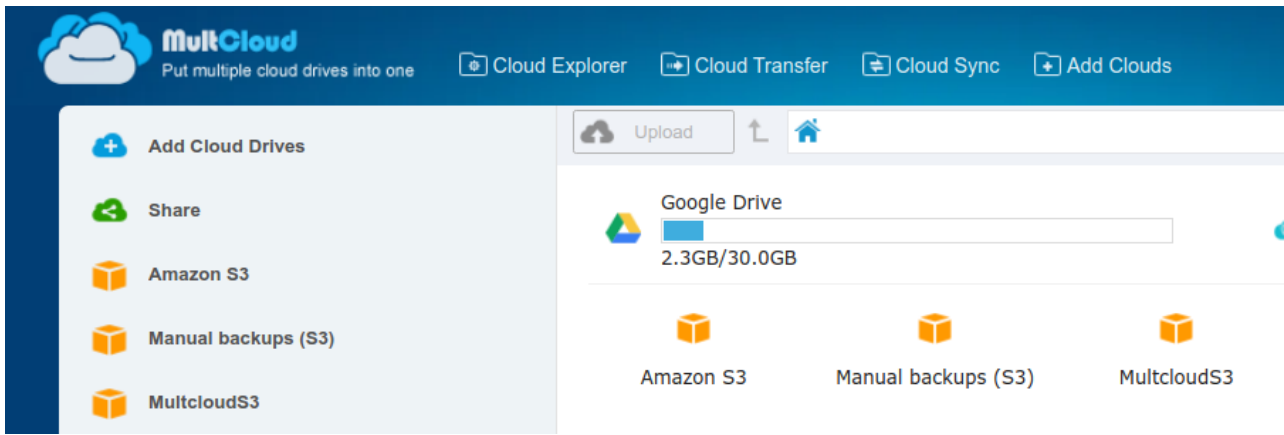
- Major cloud buckets
- Minor cloud buckets and SaaS services

The major cloud buckets are so-called because they are heavy and contain a lot of data. For me, these are:

- Cpanels
- Gsuite
- Primary cloud storage device

And these are handled as follows:

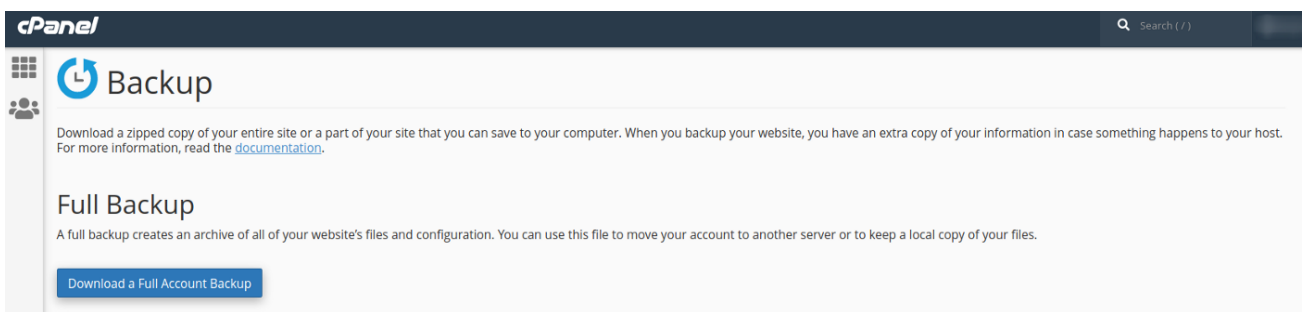
Google Drive (not: Gsuite) + cloud storage --> S3: Via Multcloud (Ongoing, Automated)



I use [Multcloud](#) to automatically sync GDrive and main cloud storage to S3 once a week.

Because the likelihood of any major cloud provider losing one's data is infinitesimally small, I would feel comfortable running these backup syncs less regularly than that.

Cpanels (Every 3 months, Manual)



Every 3 months I manually backup and upload to S3 the websites that I manage.

There's not much more to do than running a full account backup and then uploading and overwriting those files to S3. I haven't managed to get cloud-to-cloud running for these.

GSuite (Every 3 months, Manual)

Every 3 months I run a Gsuite Takeout.

Beacues I'm already capturing Google Drive I don't include that in the archive.

I put this up to S3 and keep a copy on a local hard drive too.

3: Cloud Backups: Minor

Account

Privacy

Ads

Communications

How others see your profile and network information

How others see your LinkedIn activity

How LinkedIn uses your data

Job seeking preferences

Blocking and hiding

Mentions or tags by others

Choose whether other members can mention or tag you

No

How LinkedIn uses your data

Manage your data and activity

Review the data that you've provided, and make changes if you'd like

Getting a copy of your data

See your options for accessing a copy of your account data, connections, and more

Change

Manage who can discover your profile from your email address

Choose who can discover your profile if they are not connected to you but have your email address

Change


Everyone

Manage who can discover your profile from your phone number

Choose who can discover your profile if they have your phone number

Change

Nobody



In [this Github repository](#) I have documented backing up what I call "minor" cloud services and provided instructions for all the providers that I am familiar with — although there are many, many, more.

Every 3 to 6 months I will manually run through this checklist and upload the backups to S3.

4: Pull S3 --> Local (Manual, Every 6 Months)

In order to keep a second copy of the cloud media that is onsite (S3 is clearly another cloud) about once every six months I then pull down my S3 buckets and copy them locally.

Specifically I want to download:

- My hosting backups
- The minor cloud service backups
- The GDrive and Cloud storage backups

I usually save the Gsuite backup not including Gdrive directly onto the local

5: Contact Information

I have certainly invested quite a bit of time in devising this strategy. I am sure that it could be approved upon. However, for the moment it works:

- Since instituting my local backups I have had a stable Linux system for about two years. No more upgrades bricking the system. In fact, Timeshift has all that's been required when an update has corrupted my system.
- I feel much, much more confident putting everything I do on the cloud once I know I have a system for getting a copy. There are things that some people are worried about when using cloud computing that don't worry me. And conversely, I feel a strong need to retain a copy I can control and access of all my cloud data in one place. Others trust their providers' own backup strategies and do not feel the need. We can agree to respect one another's preferences I hope!
- If you would like to get in touch to suggest any improvements please email github at danielrosehill dot co dot il. Thank you for checking out this repository!