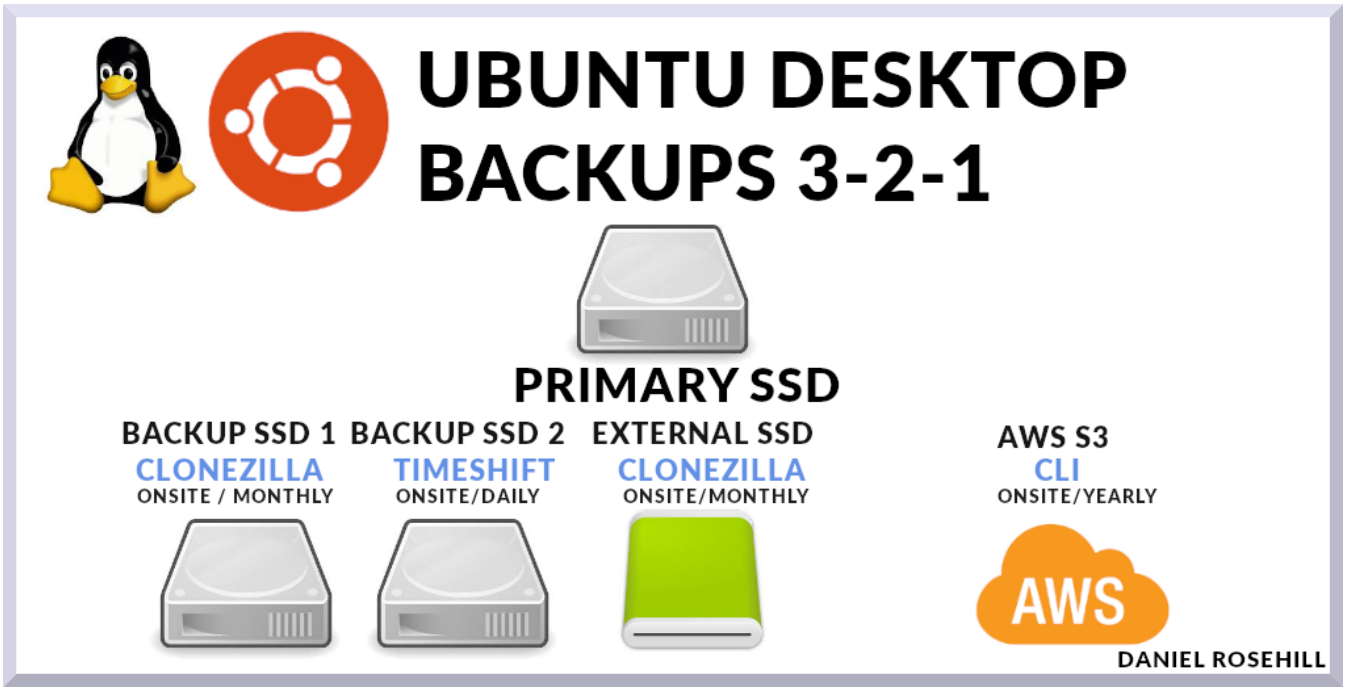
My current Ubuntu (desktop) backup strategy

Daniel Rosehill
Jan 6 · 6 min read



A schematic of my Ubuntu Desktop backup strategy, which has the primary data source (the SSD) backup up to four locations, three on-site and one off-site.

About a year and a half ago, two very unfortunate incidents hampered my productivity and occurred within the space of one month:

- Just after migrating web host — while I was temporarily bereft of hosting backups — **my shared hosting account was targeted by a virulent form of Russian Wordpress malware called Baba Yaga** (yes, that's its real name, and WordFence have put together an excellent white paper dissecting it here). This resulted in two

mostly sleepless weeks attempting to debug a network of 20 infected websites down to the MySQL level. Many, unfortunately,had to be reconstructed manually. Very little writing or actual work got done during that period.

- **An attempt to upgrade from an LTS version of Ubuntu to the latest release bricked my system** — and over-zealous use of the fcsk command then bricked the SSD it had been running on. I was down for another two days or so as I was forced to built my desktop back from scratch on a new storage medium.

Yes, I realize that backups is a really boring subject that we'd all rather not have to think about. But losing your desktop and/or cloud data is painful!

This painful experience — which cost far more in lost time than the price of a new SSD — taught me to take backups *very* seriously. And I set as an objective for myself that that unplanned reinstallation of Ubuntu would be the very last one I ever did.

Thankfully, my system has been up ever since, which puts me at about 18 months of uptime.

My backup strategy is a work in progress, but — in case anybody finds it of use or of interest — this is what has served me in the period since.

The 3–2–1 Backup Rule

The overarching objective for my backup strategy (besides, keeping my system up and stable) is now to adhere to the 3–2–1 backup rule.

This states that:

- You should keep three copies of your data. Which I actually find a little misleading because that means ***your primary data source plus two backups.***
- **The two copies should be kept on different devices or storage media.** For instance: an external hard drive and an AWS bucket.
- **At least one of these copies should be stored offsite.**

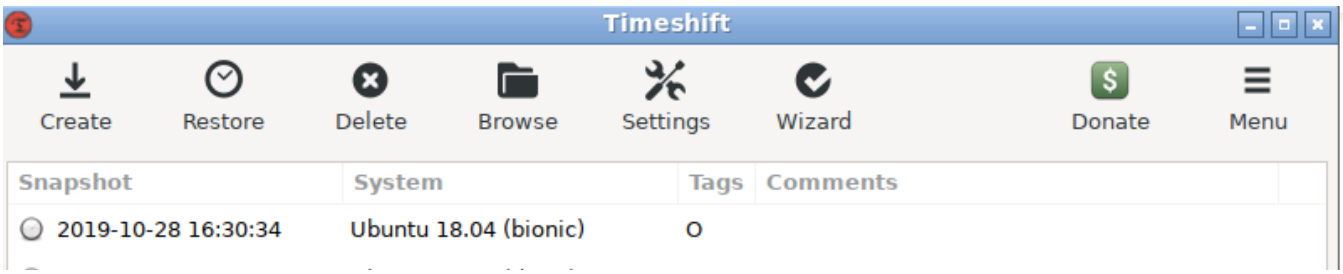
I aim to ensure that this process exists for all my data sources.

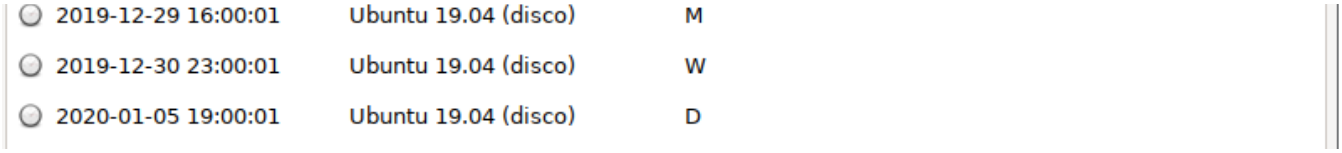
Currently these include:

- **My desktop**, which consists of my Linux drive (Ubuntu + LXQt) and my very infrequently accessed Windows drive. **Backing up these as whole drives obviously captures all virtual machines (VMs) stored on the drives** — but not drives which are temporarily mounted such as my primary cloud storage. Obviously, if you have various OSes on separate partitions rather than separate drives the process is a little more simple.
- **My cloud storage:** which consists of my primary cloud storage account.
- Data from the **various cloud services** that I use.
- My **web hosting.**

At the moment, only my desktop meets the strict criteria, so the others (namely, my cloud infrastructure) are themselves a work in progress.

Desktop Backup 1: Timeshift SSD (on-site, daily)





My current Timeshift backups, consisting of an on-demand backup, a daily backup, a weekly backup, and a monthly backup. Timeshift can restore images as a CLI if your system won't boot. It's saved me several times already. Link to the Github project here.

My first line of defense against day-to-day system stability issues is Timeshift.

I keep a dedicated 1TB SSD in my desktop just for holding restore points. I personally recommend this versus partitioning a single backup drive for various purposes. Why? Storage is cheap and it adds a little more redundancy.

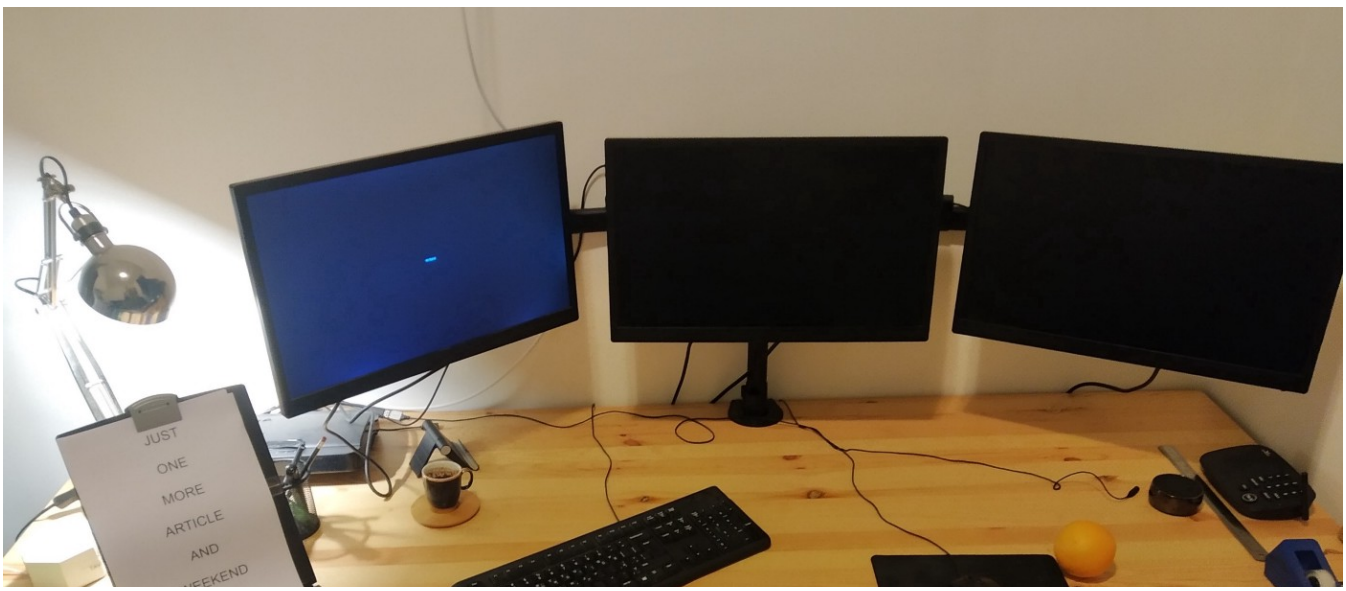
I typically keep;

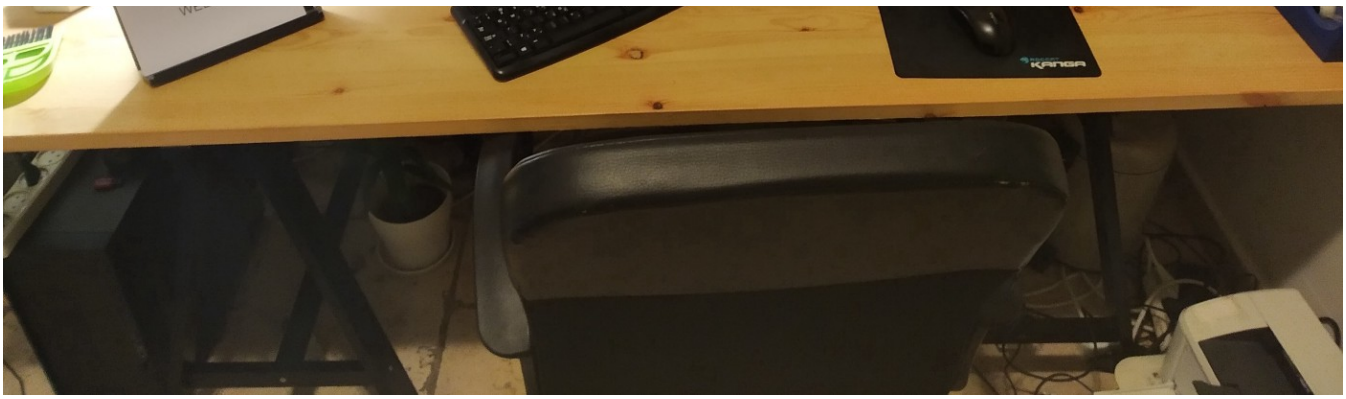
- **Two day rolling daily backups.**
- **A weekly backup.**
- **A monthly backup.**

If a set of upgrades corrupts GRUB, for instance, or the latest NVIDIA driver upgrade won't work with my graphics card so that I can't drive a monitor, I can usually roll back to a restore point taken if not this morning then yesterday.

So far, this has all that has been required to keep my system running at its best order ever.

Desktop Backup 2: Clonezilla SSD (on-site, monthly)





The desktop

Because — as a relatively soft backup engine that runs aboard a live system — I don't want to pull all my eggs in Timeshift's basket, **I also manually take backups using Clonezilla. I run this manually once a month.**

I run disk image to disk image duplicates onto another dedicated SSD (I've currently maxed out my motherboard's SATA ports!).

Desktop Backup 3: Backup, Ready-to-Run SSD (on-site, every six months)

Again using Clonezilla, every six months I format and overwrite *another* SSD again with my primary.

I keep this SSD in a hard drive case in my office — but on a shelf.

Because this SSD is not kept attached to my computer it is not vulnerable to power surges.

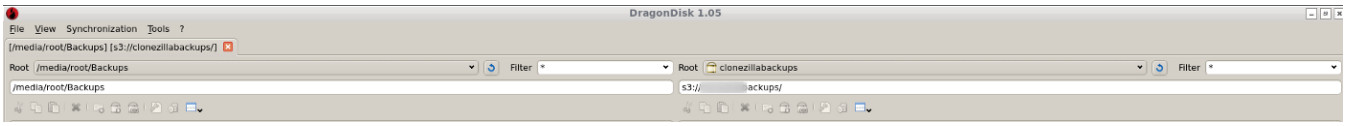
In the unlikely event that a power surge were to fry my entire desktop and all its components (including the two other on-site backups drives) this would be my quickest restore point.

In the even more unlikely even that my worldly belongings, including my computer are robbed, flooded, or go up in a fiery furnace, backup 4 will be my only restore point (of course, if you store this backup at an offsite location such as your office, you would have two points of redundancy).

Alternative approach: use Clonezilla to back up an *image* of the primary (like for Desktop Backup 2) to a simple external SSD. *Disadvantage:* slightly longer recovery

time in the catastrophic my-computer-was-fried-by-lightning-I-need-to-buy-a-new-one-now eventuality.

Desktop Backup 4: AWS S3 (off-site, yearly)



DragonDisk, an old school but reliable client for backing up to S3

Of course, for this system to be truly complete, **I need to take an off-site backup for a system disk image.**

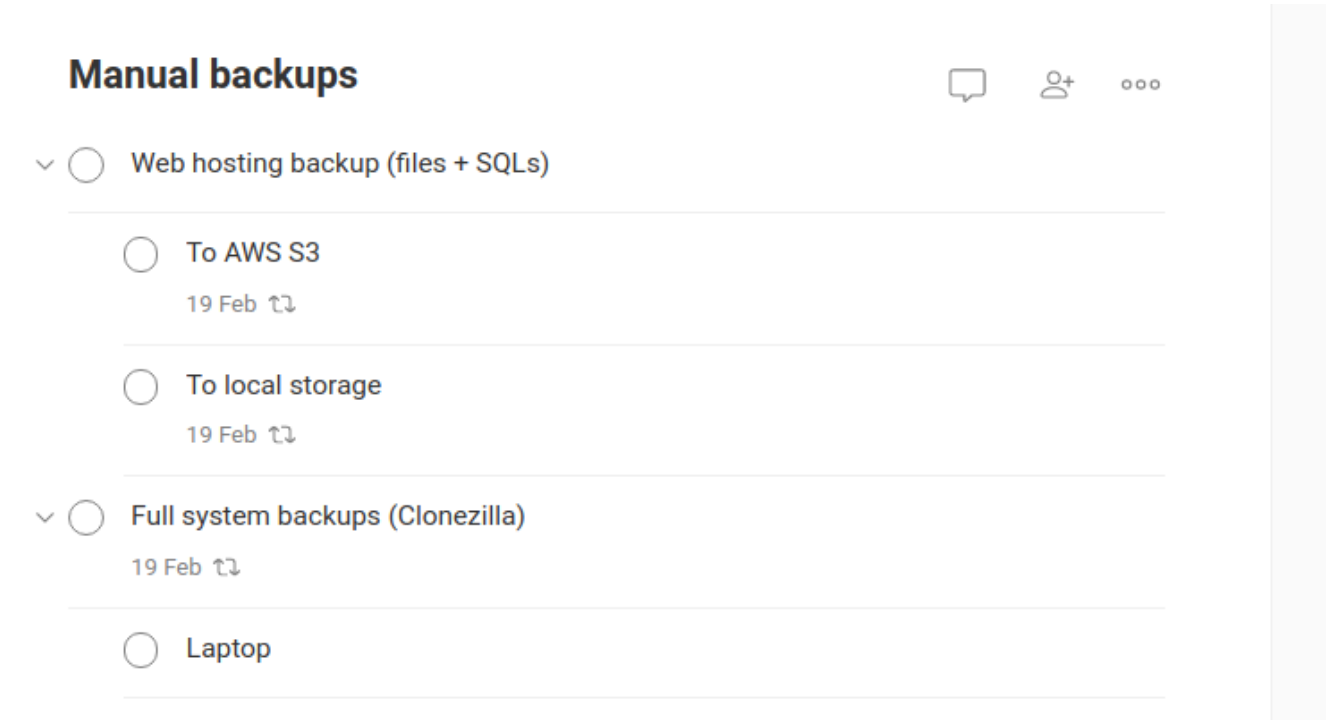
For this purpose, and given the data size involved, S3 (and S3 Glacier) is the obvious cloud-hosted option.

Uploading a file of this size would be very difficult from my home internet network, which has a typical upload speed of about 2 Mbps.

For that reason, I pushed my last image to the cloud from a company with a business-grade uplink.

This year, I will have to find another creative solution.

Reminders by Todoist



Of course, as I haven't yet figured out a way to automate all the bits and pieces in this backup strategy, I have, unfortunately, to rely upon Todoist reminders that I set in a dedicated project.

Cloud services and hosting are backed up to AWS which I figure is enough for the time being.

At some point, I want to buy a physical server to pull down all my buckets once a year.

This has been on my to-do list for a while but has been out-prioritized by other tech expenses.

But, let's be realistic about it — AWS isn't going anywhere anytime soon.

That's it for now.

Yes, it's a little complicated. Yes, it's a little time consuming.

But it's better and less time consuming than having to reinstall my desktop from scratch twice a year.

Hope you enjoyed — and if you have a better backup strategy, or have flaws to point out in mind, please let me know in the comments — or drop me an email.

Happy backing up!

Cloud Computing Backups Ubuntu Linux Desktop Computing