

Only you can see this message

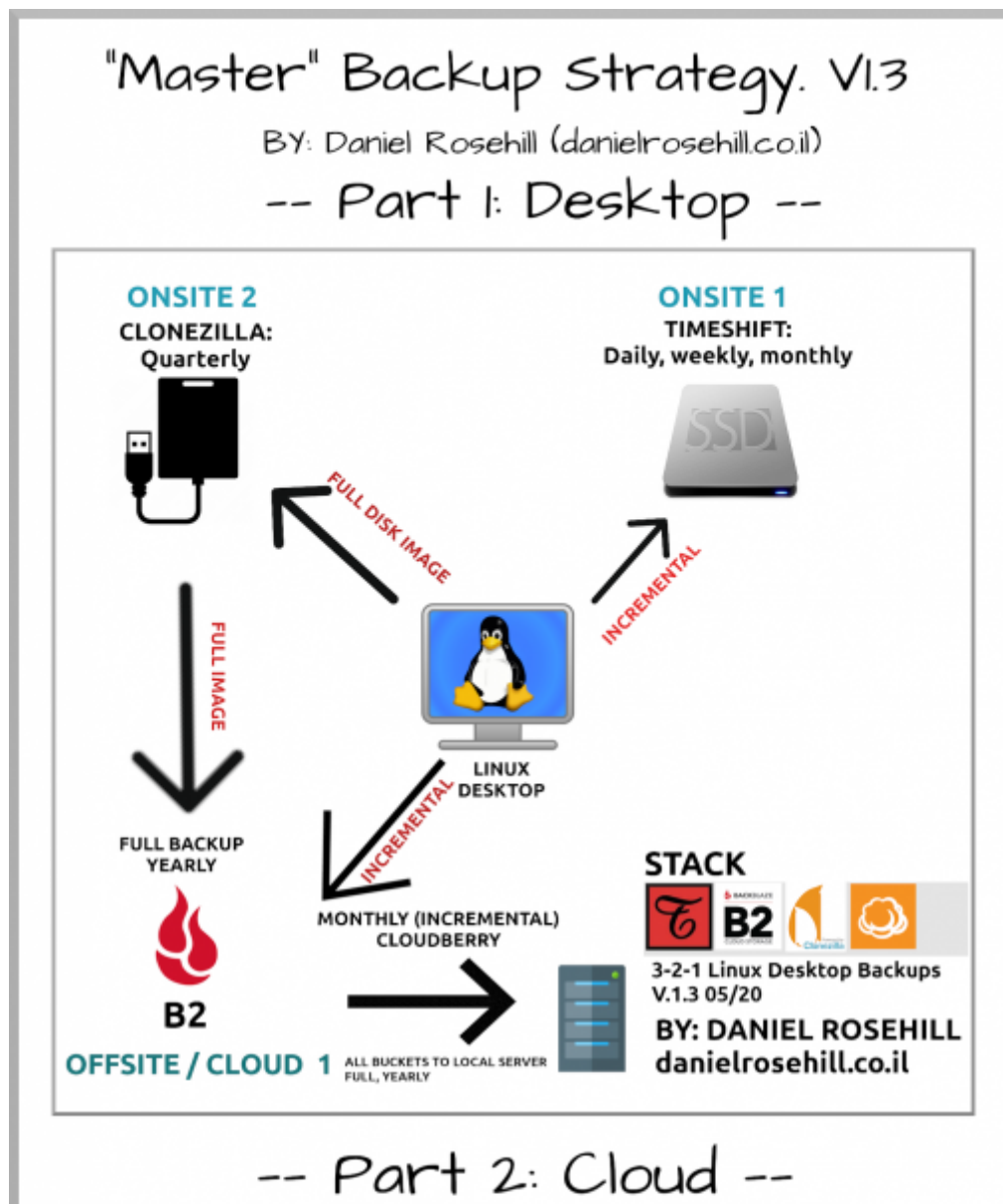
This story's distribution setting is on. [Learn more](#)

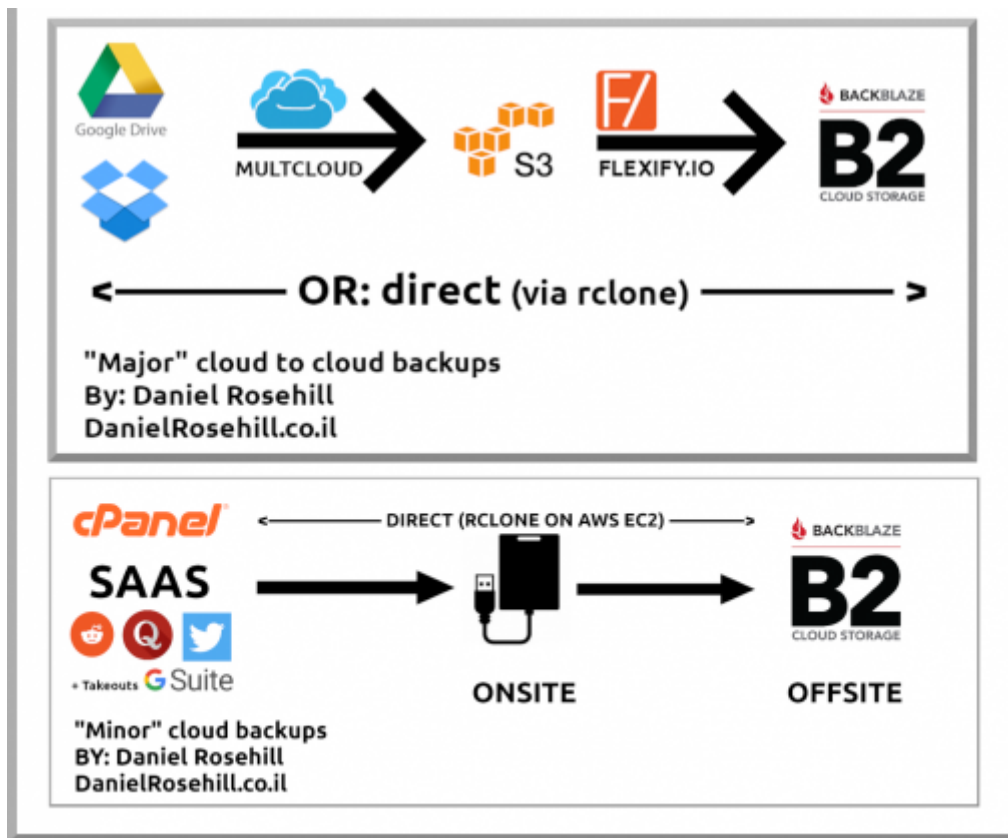
Linux Desktop and Cloud Backup: A Summary of My Master Backup Strategy And Documentation (V1.3)



Daniel Rosehill

May 10 · 5 min read





After having spent the best part of a week playing around with B2 and S3 buckets, syncing gigabytes of data across the cloud, and making sure that my home Linux setup was as well backed up as it has ever been — I'm back to, you know, actually going about daily life and generating the next pools of data which, in turn, will soon be backed up.

This intensive week of thinking about backups (and taking them) is a process I have been going through about once a year for the past several of them: I review how I'm backing up data to date and (this is important) I try to make it better.

But this year there's been one additional twist to my usual *modus operandi*: in addition to actually figuring out how to make things better, I've decided to document my findings here, on YouTube, and on Github.

Do I think that anybody cares how I back up my web hosting? Potentially a few fellow enthusiasts but certainly not most people.

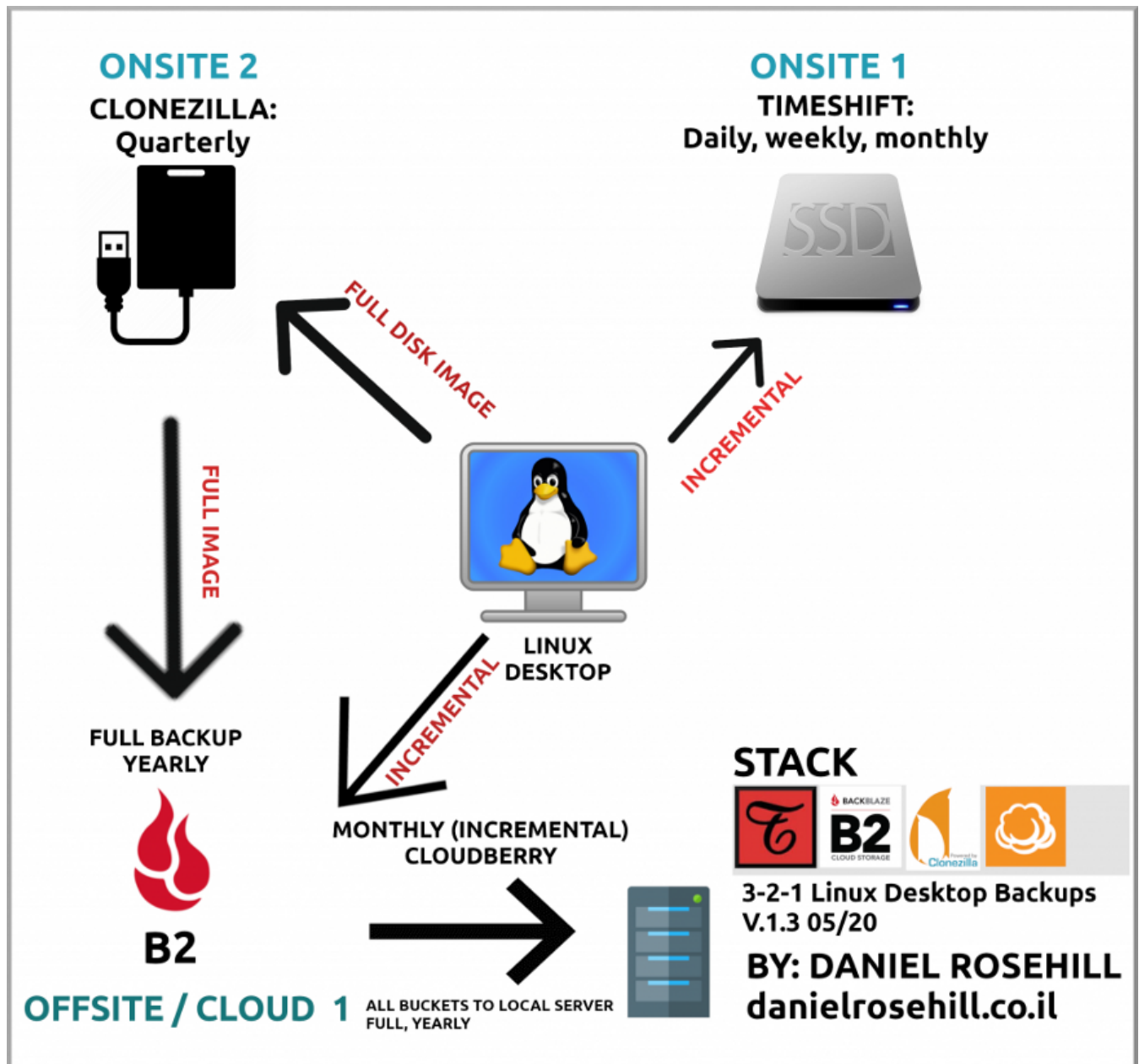
Rather, the purpose of creating all this material is really to document the journey for myself. Because in order to get better, I need to keep a log of what I am doing now.

So although this Github repository does an adequate job of explaining my current approach: to tie everything together, here is V1.3 of my Master Backup Strategy.

See 'Master Backup Strategy' on Github

. . .

Part 1: Linux Desktop



The first and most important part of my backup approach is to make sure that my desktop data is protected on several levels.

Firstly, I take daily, weekly, and monthly snapshots via Timeshift onto an internal SSD. There are, however, two deficiencies with this approach:

- If my desktop *and* the SSD it contained were destroyed in a power surge this onsite backup would be useless.
- Timeshift backups are not that low-level: they're taken aboard a live system.

So, I also take Clonezilla disk-to-image backups onto a local SSD every 3 months or so.

Although I love Timeshift — and it has been all that has been required to “save” my system to date — I trust Clonezilla more . It's a low-level tool. It doesn't run aboard a live system. And it's very, very robust.

To cover the required offsite component, I set up a desktop bucket on Backblaze B2.

Like with the local approach, I've applied two layers of protection to make sure that I have strong protection offsite as well as on my local network:

- I use Cloudberry to take frequent incremental backups up to the cloud.
- I use rclone to take less frequent copies of the Clonezilla images and push them to the cloud.

SUMMARIES:

- My current Ubuntu (desktop) backup strategy

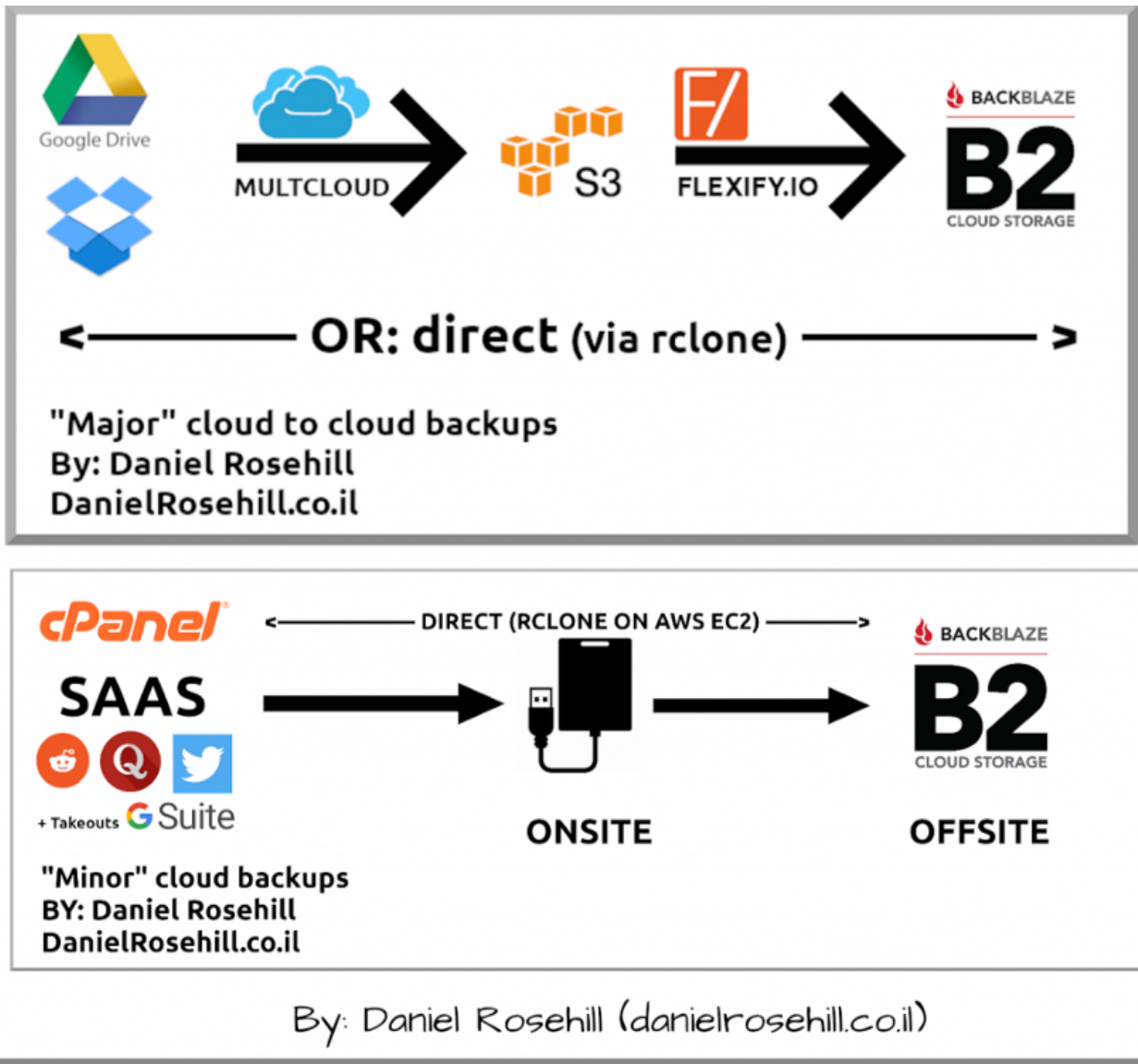
TOOLS:

- Timeshift
- Clonezilla
- Cloudberry
- rclone (CLI)

. . .

Part 2: The Cloud

-- The Cloud: Major and Minor --



Mopping up all the stuff in the cloud is trickier because we're not talking about one data source but rather several discrete pools of data aboard proprietary filesystems.

So I start with the big stuff — Gsuite and web hosting — and work my way down to the smaller data sources in SaaS applications.

- **The only backup I really need to use as a traditional backup (recent snapshots etc) is my cloud storage**, which is primarily on Google Drive. I use a cloud storage tool to make sure that there are always a couple of snapshots of that in S3.
- **However, simply syncing Drive doesn't cover all the other stuff that I put into Google's cloud.** To name but a few, there are: contacts, bookmarks, YouTube videos, Google Photos — among many others. Because I want to capture and back up *all* of my data through this process I initiate a Google Takeout and then use an AWS EC2 instance to upload this to B2 cloud to cloud (where I can avail of a practicable upload speed). I documented that recently here. Also: because cloud

storage is unlikely to fail and this is a bit tedious I only go through this process about once a year (just Google Drive, by contrast, is moved across the clouds once a week).

For web hosting: I back up Cpanels using the built in backup tool. I can again use EC2 to pull and push these up to B2 quickly. And I also keep a local copy.

The messiest part of this is backing up data contained in all the various SaaS services that I use. I documented those sources on Github recently and the various processes for getting data out of them (inconsistent!).

This process is manual — and if there were a tool that extracted and centralized this backup at the press of a button I would happily use it!

Instead I set calendar reminders and — approximately once a quarter — bundle all my cloud backups into a B2 bucket.

SUMMARIES:

- Using Cloudberry to upload Clonezilla disk images to cloud object storage
- Backing up pCloud to B2 using EC2 and rclone
- How to: Backup Your Web Hosting (According to the 3–2–1 Principle)
- How to Backup Gsuite Takeouts to Backblaze B2 (Using rclone + EC2)
- Cloud backup approaches (GitHub)

TOOLS:

- FilezillaPro
- Cloudberry
- AWS EC2

. . .

The Final Touch

There is one thing left to do.

Data is being aggregated offsite on B2.

But not all of that data has been duplicated to an onsite backup source yet (e.g. the SaaS backups that I pushed directly from one cloud to another).

For that reason once a year all this data is pulled down from cloud object storage to a local backup server, thus ensuring that everything is backed up twice: on another source and on an onsite/offsite medium.

. . .

Please Let Me Know Your Feedback

I've started my version control at 1.3 because — realistically — this backup strategy has already gone through several iterations, including the last one to move cloud object storage from S3 to B2.

This backup strategy is far from perfect and I am sure that there are many things that I haven't thought of — or which could be done better.

So if you have any ideas for improvement kindly drop me a message!

. . .

Originally published at <https://www.danielrosehill.co.il>

[Backups](#) [Technology](#) [Linux](#) [Cloud Computing](#) [Data](#)

[About](#) [Help](#) [Legal](#)

Get the Medium app

