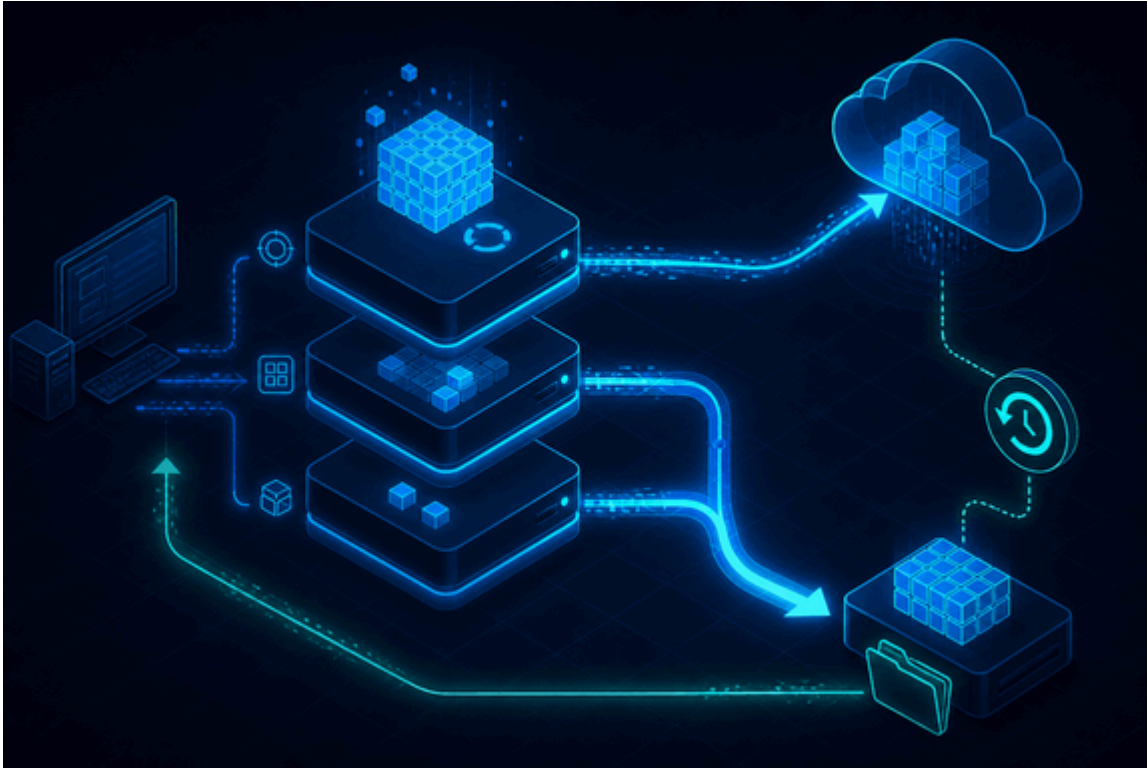


Backup Strategies: Full, Differential, Incremental — and Actually Getting Data Back

Wed, Aug 26, 6:00 PM EDT · <https://scottslab.io/posts/backup-strategies-full-differential-incremental>



TL;DR

Three backup types trade space against restore effort. Full backups copy everything and are the simplest to restore. Differential backups capture all changes since the last full, so restoring means the full plus one differential. Incremental backups capture only the changes since the last backup of any kind: smallest to write, but restoring means replaying the full plus every incremental in order. The most common reason you restore isn't a disaster, it's human or technical error. Keep at least one copy offsite, and remember a backup you've never restored from is a hope, not a backup.

There are three primary backup types, and the whole choice is a trade between how much space they eat and how much work a restore takes.

A full backup is a complete copy of all the data. It's the simplest to restore because everything's in one place, and the most expensive in time and space to make. Snapshots are a hardware-platform variant of the same idea. A differential backup captures all changes since the last full backup; it grows each day as changes accumulate, but restoring is easy: the last full plus one differential. An incremental backup captures only the changes since the last backup of any kind, full or incremental.

It's the smallest and fastest to write, but the restore is the most work, because you replay the full plus every incremental since, in order.

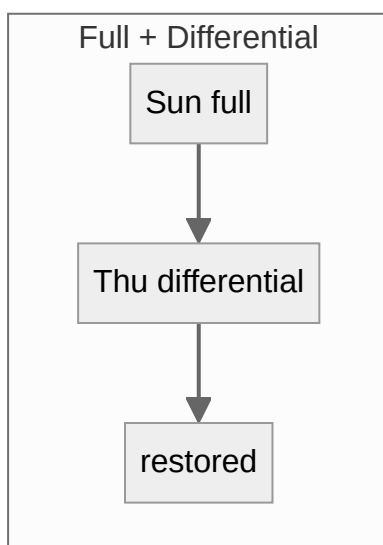
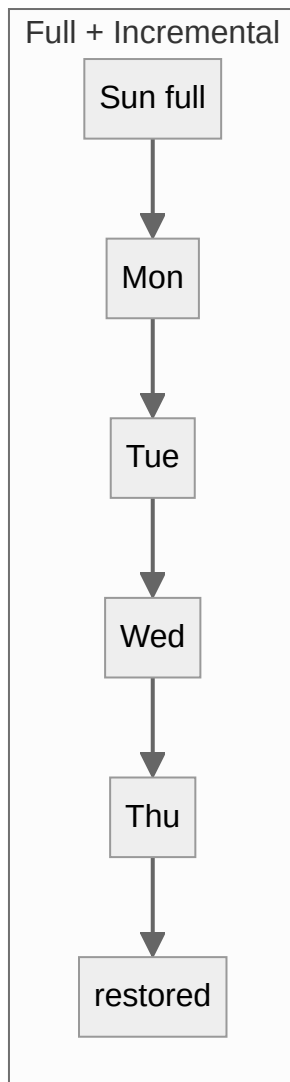
Full everything · simplest restore · most space

Differential changes since last FULL · restore = full + 1

Incremental changes since last backup · restore = full + all, in order
--

Backup Strategies: Full, Differential, Incremental — and Actually Getting Data Back

Make it concrete with a Sunday-full schedule. With differentials, to recover on Friday you restore Sunday's full, then Thursday's differential. Two steps, done. With incrementals, you restore Sunday's full, then apply Monday, Tuesday, Wednesday, and Thursday's incrementals in order. Five steps, and if any one is missing or corrupt, the chain breaks. That's the trade in one example: differential spends storage for a faster, more robust restore; incremental saves storage at the cost of a longer, more fragile one.



Backup Strategies: Full, Differential, Incremental — and Actually Getting Data Back

A few things the metrics never tell you. The most common reason you'll actually restore is boring: human or technical error, someone deleting the wrong thing. Not a hurricane. Optimize for the

routine restore, not just the apocalypse. Keep at least one copy offsite (cloud counts); if you're cloud-only, consider two different providers, because "all my backups are in one account" is a single point of failure wearing a redundancy costume. Online backups restore instantly but cost more; offline backups are cheaper but need someone to go get them. Two patterns worth knowing. The non-persistence approach backs up only each server's unique data and rebuilds the OS from infrastructure-as-code, so you never back up what you can regenerate. Live boot media, a bootable USB, lets you pull data off a machine's storage without trusting, or even starting, its own OS.

The one rule under all of it: a backup you have never restored from is not a backup, it's a guess. Test the restore, or you don't have one.

Written by Scott S. — Contact: scott@scottschlangen.com — scottslab.io — <https://scottslab.io/posts/backup-strategies-full-differential-incremental>