Engineer

Data storage

keeping data, or the media that hold it, for later reading

What it is

Data storage, usually shortened to storage, is the layer of a system where data lives between uses, on media that hold it through a reboot and a power cut. The NIST glossary gives the word two senses, both taken from SP 800-88r2: the retrievable retention of data, and the hardware that data may be entered onto and retrieved from.

Every kind rests on a physical medium: a spinning hard disk, a solid-state drive, an SD card, a tape. Software sees one of three layers above it:

- **Block storage** presents a device as a run of fixed-size blocks addressed by number. A disk partition, a RAID array and a cloud virtual disk all look like this to the kernel, and a filesystem or a database engine decides what the blocks mean.
- **File storage** lays a filesystem over those blocks: names, directories, sizes, owners and permissions, and a file can be changed a few bytes at a time. Shared over NFS or SMB, one tree serves many machines.
- **Object storage** keeps whole objects, each a run of bytes plus metadata, under a name inside a container, and is reached through an HTTP API rather than a mount. Google Cloud Storage stores data as objects in containers called buckets, every bucket belongs to a project, and each object holds a file of any format as data that cannot be changed once written.

The three part ways over what a change costs. A block device rewrites one block in place. A filesystem edits part of a file and moves it with a rename. An object store replaces the whole object: Google's objects page rules out appends and truncation, and a replacement is atomic, so readers get the old version until the new upload completes. Depending on the tool, a rename can mean a copy under the new name and a delete of the original; Cloud Storage also supports atomic

object renames within a bucket. In a bucket with a flat namespace, folders are a picture the tools draw from slashes in object names; the string `folder1/` is part of the name, and the bucket stores no folder at all.

The same file at each layer looks like this:

```
# block: the kernel's view of a disk as numbered blocks
lsblk /dev/sda

# file: a filesystem mounted over those blocks, with names and directories
df -h /srv

# object: a whole object sent to a bucket over an API
gcloud storage cp readings.csv.gz gs://example-bucket/2024/01/
```

Use the concept in a decision

Put “What is data storage?” to work in a real decision. We can help relate the concept to the system you are building or buying. Backups that are tested rather than assumed, and a written plan for the morning a server does not come back on.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Automated 100 critical data backups using Barman — 100 critical backups automated across the databases

Automated 100 critical data backups using Barman Get in touch →

Why it matters

The medium decides how data is lost. A hard disk, an SSD and an SD card each fail in their own way, and a recovery method that suits one is the wrong tool for the next. Redundancy inside one box, such as a mirrored pair or a RAID array, keeps data readable after a single disk dies, but it writes every deletion to every disk as faithfully as it writes the data.

Durability and availability are separate promises. Durability is the chance that stored data survives; availability is the chance it can be read at a given moment. Cloud Storage states 99.999999999% annual durability for every storage class, while the typical monthly availability it gives runs from 99.9% to above 99.99%, depending on class and location. A storage class is metadata on every object that sets its availability and pricing model, and new objects inherit their bucket’s default. The colder classes trade a lower price for keeping data against slightly

lower availability, a minimum storage duration of 30, 90 or 365 days and a charge for reading the data back; Google aims them at data read once a month or less, at most once a quarter, and less than once a year.

Location is part of the data. A copy on the same disk, the same machine or the same cloud account shares that thing's fate, and a copy elsewhere does not. Location also sets speed and cost: Google notes that keeping data in the same region as the compute that uses it maximises performance for data-intensive work and can reduce network charges.

How software reaches storage shapes the rest. Restic, a backup program, defines its repository as nothing more than a directory holding a set of subdirectories and files, and that same directory can sit on a local disk, on a server reached over SFTP, or in an object store such as Google Cloud Storage or Amazon S3. The bytes stay the same while the latency, the failure modes and the person who controls the account all change.

What it looks like in practice

From the bottom of the stack up, the founder's earlier work touched the physical medium that fails and the object store that moves whole files. The company's own backup implementation sets out automation for a copy at another provider as protection against losing a host; activating that storage and verifying a live restore are separate steps.

The medium: data recovery ran across pretty much every common storage type, from SD cards and spinning HDDs to SSDs, RAID arrays, external drives and Mac systems. People brought in storage that had failed or been damaged, holding photos, business files, the only copy of something that mattered. Each medium fails and gives up its data differently: an SSD is a different problem from a platter drive, a RAID array is a different problem again, and a Mac filesystem has quirks of its own. Knowing the right approach for the specific medium mattered as much as any single technique, and clients got critical data back from devices they had already written off as gone.

The object store: in six ETL/ELT pipelines feeding Google BigQuery, a Google Cloud Storage bucket stood between the source and the warehouse. Curated datasets and tables were exported to intermediate formats, compressed with

gzip and transferred to the bucket, and from there the bq CLI and Python scripts loaded them into BigQuery. Whole compressed files went in and whole files came out, the pattern object storage is built for.

The copy elsewhere is part of the company's backup and restore-test implementation. An Ansible role installs restic scripts and systemd units for a database dump where configured, an encrypted, deduplicated snapshot to storage at another provider over SFTP, retention pruning and an integrity check. A separate monthly workflow restores a known file to a scratch location and compares its bytes, testing the read-back path rather than just the write. Backup and restore have separate success signals, and the timers require a configured repository. The implementation record dated 2026-10-27 says those timers were disabled without a repository and records no completed live restore. It documents the automation and what must be activated; it does not prove that the company can already lose the host and recover its data.

Where it goes wrong

Most storage trouble comes from a copy, a check or a cost that looked fine until the day it was needed.

A copy shares a fate with the original. A snapshot in the same account, a backup on the same disk, a replica in the same building: each survives the small failures and goes down with the large one. List the single events that could take out both, and move one copy until the list is short.

Redundancy gets counted as a backup. A mirror copies every write to every disk, deletions and corruption included. It answers a dead disk; a separate copy with its own history answers a mistake.

An object store is treated as a filesystem. Code written for files expects cheap renames and directory listings. Some tools implement object renames as a copy plus a delete, even though Cloud Storage also supports atomic renames within a bucket. Google notes that deeply nested, directory-like structures in a flat namespace lack the listing performance of a native filesystem. Lay out object names for the way they will be read, and keep work that edits files in place on file or block storage.

The check reads the index and skips the data. By default, restic's `check` command verifies the repository's structure and leaves the pack files that hold the backed-up data unread, because reading them means downloading every one. `--read-data` reads them all, and `--read-data-subset=1/5` through `5/5` covers the repository in five runs. A check that only lists files proves that names exist, and little about the bytes behind them.

A cold class holds hot data. A file read daily from a class priced for monthly reads pays a charge on every read, so match the class to the read pattern the data actually has.

Further reading

- NIST glossary: storage — the two senses of the word as NIST SP 800-88r2 defines them.
- About Cloud Storage objects — object data and metadata, naming rules, immutability and the flat namespace behind simulated folders.
- Cloud Storage: Storage classes — durability, availability, minimum storage durations and retrieval fees for each class.
- restic: Preparing a new repository — what a repository is, and the local, SFTP and object-store backends it can live on.
- restic: Working with repositories — checking a repository's structure and reading back the data it holds.
- Cloud Storage: Move objects — the atomic rename operation within one bucket, distinct from tools that simulate a move by copying and deleting.

Glossary

Choose for your actual constraints

Bring the deployment path, the operational risks and the people on call. We can scope a change with checks, a recovery route and clear ownership.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is data storage?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/storage/