Engineer

Cloud

computing rented by the hour from somebody else

What it is

Renting computers you will never see is the shortest honest description of the cloud, known in full as cloud computing: a provider owns the buildings, the power and the hardware, and the customer switches servers, storage and networks on and off through a web console or an API, paying for what was used. Every “cloud” workload still runs on a physical machine in a data centre; someone else bought it and rents slices of it to many customers at once.

A precise definition comes from the US National Institute of Standards and Technology. Its Special Publication 800-145, written by Peter Mell and Timothy Grance and published in September 2011, describes cloud computing as a model for “on-demand network access to a shared pool of configurable computing resources” that can be provisioned and released with minimal management effort. It names five characteristics. On-demand self-service means a new server without talking to anybody. Broad network access means the service is reached over the network from ordinary devices. Resource pooling means customers share the provider’s hardware and generally choose a country or a data centre, never a rack. Rapid elasticity means capacity grows and shrinks with demand. Measured service means usage is metered, which is what makes a bill by the hour or the gigabyte possible.

A dedicated server rented by the year, changed by ticket and billed at a flat rate meets few of these tests. That is hosting, often a good choice, but a different contract with different economics.

The same document sorts cloud services into three models. With infrastructure as a service (IaaS) the customer rents virtual machines, disks and networks, and patches the operating system on them. With platform as a service (PaaS) the customer deploys code onto the provider’s runtime. With software as a service

(SaaS) the customer uses a finished application, such as web mail. It adds four deployment models: public, open to anyone; private, provisioned for one organisation and possibly on its own premises; community, shared by organisations with common concerns; and hybrid, two or more of these bound together so data and applications can move between them. The best-known public clouds are large platforms such as AWS, Azure and Google Cloud, and beside them smaller providers sell plain virtual machines at simpler prices.

Use the concept in a decision

Put “What is the cloud?” to work in a real decision. We can help relate the concept to the system you are building or buying. Servers sized for what you actually use, moved into the cloud or back out of it, with the monthly bill worked out first.

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. Ran 20 Docker apps on Google Cloud and AWS — 20 containerised applications deployed and watched

Ran 20 Docker apps on Google Cloud and AWS Get in touch →

Why it matters

The cloud turns a purchase into a running cost, and that is most of the decision in one line. Buying servers means paying up front for capacity sized to a guess about next year’s busiest day. Renting means paying every month for what is switched on, with the right to change the answer next week. For a new product or a seasonal business, that flexibility is worth a premium. For a workload that has run at the same size for three years, the premium keeps being charged long after anyone used the flexibility.

The second decision is how far up the stack to rent. A virtual machine is close to a server a team already understands, and it moves between providers with little more than a copy of the disk. A managed database, a serverless function or a proprietary data warehouse takes work off the team, and each one is also a piece of the application written against one company’s interface. The useful habit is to know what leaving each managed service would cost, and to accept that cost on purpose rather than discover it during a price rise.

The third is responsibility. Renting hardware hands over the building, the power and the hypervisor; who can log in, which ports are open and whether the data has a copy elsewhere stay with the customer. The large providers each publish a shared responsibility model that draws this line, and the accounts, the configuration and the data sit on the customer’s side of every version.

What it looks like in practice

For one international client the subject was the bill. Their AWS setup had been over-provisioned and had collected services they no longer used, so the monthly cost had drifted far out of proportion to the business. The work began with an audit of what was really used, removed unnecessary services, adjusted the remaining capacity and moved workloads away from AWS. Those are the infrastructure changes described by the engagement record; it does not establish a like-for-like cost ratio, a productivity result or a disruption-free migration.

The company's own public-services host sits at the other end of the scale. It is a single-core cloud instance with 512 MB of memory and 10 GB of disk, about 464 MB of it usable, serving TLS for seven domains, a git forge, a Tor onion service, Gemini and Gopher servers and intrusion banning. Backup automation is implemented separately; its configuration does not establish active encrypted backups. In a cloud console the easy answer to a list that long is a bigger instance type, and the onion service went on without one: over a recorded day of samples, the instance's lowest reading of available memory was about 310 MB of 464, its swap use 2.6 per cent and its single core 99.7 per cent idle. The machine costs less per month than a lunch, and mail is deliberately kept off it because a mail server needs headroom this one has already spent.

Renting also means deciding how recovery extends beyond the provider's copy. The company's backup and restore-test implementation supports a database dump where configured, then an encrypted, deduplicated restic snapshot over SFTP to storage at another provider, retention pruning and an integrity check. Backup and restore have separate success signals. A separate monthly workflow brings a known file back to a scratch location and compares its bytes, testing the return trip rather than just the upload. Timers depend on a configured repository: the implementation record dated 2026-10-27 says they were disabled without one and records no completed live restore. That is an implemented recovery design with activation prerequisites, not proof of a successful recovery or of the current runtime state.

Where it goes wrong

Clouds fail less often than the habits around them, and the common problems are slow, quiet and expensive.

The bill grows while the product stays the same size. Resources accrete: a test server left running after the test, a disk still billed after its machine was deleted, snapshots kept forever, a load balancer in front of nothing. Moving data out of a provider, or between its regions, is metered too, so shipping logs and backups across those boundaries pays on every copy. Tag every resource with an owner and a purpose, set a budget alert, and read the bill line by line once a month.

Capacity is sized for a day that never comes. Machine sizes are picked once, at launch, from a guess about the worst case, and the guess outlives the people who made it. Measure what a workload uses across a normal week and a busy one, size to that with a margin, and let the autoscaler shrink as readily as it grows.

The provider's snapshot is treated as the backup. It lives in the same account, behind the same credentials, at the same company as the machine it protects, so a stolen login or a suspended account takes the server and its only copy together. A real backup sits under different credentials, ideally at a different provider, encrypted with a key that provider never holds, and is restored on a schedule.

Security stays where the defaults left it. The usual openings are a firewall rule admitting SSH from the whole internet, a long-lived access key in a developer's dotfiles, an administrator account used for everyday work and a storage bucket made public for one demo. Grant the least access that does the job, prefer short-lived credentials, require a second factor on every human login, and keep the configuration in code so that a change to it gets reviewed.

Leaving turns out to need a rewrite. Every proprietary queue, function runtime and database dialect ties code to one provider a little more firmly. Keep the core on open interfaces where that is cheap, such as PostgreSQL, containers and plain virtual machines, and write down the exit cost of each proprietary service when it is adopted.

Further reading

- The NIST Definition of Cloud Computing — the short definition with its five characteristics, three service models and four deployment models.
- Guidelines on Security and Privacy in Public Cloud Computing — what an organisation should weigh before moving data, applications and infrastructure to a public provider.

- NIST Cloud Computing Reference Architecture — the architecture and taxonomy NIST built on its definition, naming the parties to a cloud arrangement.
- FinOps Framework — the FinOps Foundation's model for making cloud spending a shared practice of engineering and finance.

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 the cloud?" 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/cloud/