Engineer

Host

a computer attached to a network

What it is

A host is a computer at the end of a network path: the machine traffic is addressed to and comes from, as opposed to the routers and switches that only pass it along. Engineers also say *host machine*, call the name it answers to its *hostname*, and in virtualisation speak of the physical host that carries guest machines. To *host* something, as in web hosting, is to run it on such a machine.

The internet's own definition dates from October 1989. RFC 1122, the requirements document for internet hosts, calls a host computer, or simply host, the ultimate consumer of communication services, a machine that generally runs application programs on behalf of its users. It corresponds to what the OSI model called an end system. The networks between hosts are joined by gateways, which today are called routers. The same document covers everything from single-purpose hosts to full-service ones, and calls a host with more than one network interface multihomed.

That line between host and router is drawn by behaviour, not by hardware. RFC 1122 says that any host "that forwards datagrams generated by another host is acting as a gateway", requires a switch that turns the gateway function off, and requires that switch to default to off. A host with two network interfaces still does not start forwarding on its own. A laptop, a phone, a printer, a rented virtual machine and a container with an address of its own are all hosts.

NIST's glossary keeps two definitions side by side. One counts any hardware device with a way onto a network. The other, from guidance on securing web servers, says that a server is a host to its clients and that a client station is a host too, because it is "a source of information to the network", unlike a router or switch, which directs traffic. That is the useful contrast with the word server. A

server is a role a program plays, answering requests. A host is the machine, whether it answers anyone or not. One host can run many servers, and a laptop that runs none is still a host.

A host collects names. The companion document, RFC 1123, set the rules for host names: the first character may be a digit as well as a letter, host software must handle names up to 63 characters and should handle names up to 255, and wherever a user can type a host's name, typing its address in dotted-decimal form should work as well. On top of that, the DNS can point any number of names at one address, and HTTP lets a client say which of them it wants in the `Host` field of each request, so a single host can answer for many sites.

Use the concept in a decision

Put "What is a host?" 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 host is the unit that operations count by. Inventories list hosts, monitoring watches hosts, firewall rules and access lists name them, and cloud providers bill them by the hour or month. A plan that says "add a host" is concrete in a way that "add capacity" is not: it means one more machine to create, name, patch, back up and pay for.

It is also the unit of compromise. Whoever controls a host controls every server program on it, every key stored on its disk and every connection it can make. What a host holds sets the cost of losing it, which is why careful operators decide per host which secrets, which data and which network paths it gets, and give it no more.

And it is where names meet. The same machine is known by the name in the inventory, the name it calls itself, the DNS names pointing at its address and the names in the requests it receives. When those agree, a log line, an alert and a support ticket lead to the same box. When they drift, a person paged at night spends the first minutes of the incident finding the machine.

What it looks like in practice

This site's operations work treats the host as the thing to build from code and give its own secrets. When the platform needed a second machine at a different provider, the second host is created, or re-adopted, from the repository. The first machine had been made by hand and given its firewall afterwards, which leaves a window where a fresh host with a default password policy is reachable from the internet. The role that creates the second one adopts an existing machine rather than duplicating it, so running it again is safe, and its own file states the instance type, the region, the specification and the monthly cost in euros, because it is the one role in the repository that spends money.

Going from one host to two then exposed every value written on the assumption that there would only ever be one. Each operational secret became a map keyed by host. The page records why sharing was wrong in two ways. A dead-man's switch shared by two hosts alerts less than two switches, not more: either host still checking in keeps it green while the other is dead. And sharing a backup passphrase couples the hosts' exposure. Separate restic repositories can have different master keys even when they use the same passphrase, but someone who obtains that passphrase and can read both repositories' encrypted data and key files can unlock both. The existing host was named in all four maps with its repository and both switches left empty, which the page calls the truthful state.

From a visitor's side the word means something smaller: a name the browser contacts. The security page invites a reader to open the browser's network panel and count the hosts, and says there is one, and it is the site's own.

Where it goes wrong

A host starts forwarding traffic it was meant only to receive. Container networking and VPN software switch IP forwarding on, and from then on the host routes between its interfaces. Docker's documentation, for one, says it enables both settings below when it starts and then drops forwarded packets unless a rule accepts them. Where forwarding is on without such a rule, firewall rules written for traffic addressed to the host leave traffic passing through it alone, and a network behind it becomes reachable by a path nobody drew. Read the switches, and write forwarding rules on purpose wherever they say `1`:

```
sysctl net.ipv4.ip_forward net.ipv6.conf.all.forwarding
```

The names disagree. The inventory says one thing, the machine calls itself whatever the provider's image chose, DNS has a third name and the TLS certificate a fourth. Every alert then carries the name nobody uses. Take the machine's own hostname from the same source as the inventory and the DNS records, and set it when the host is created rather than when someone notices.

A server answers whatever host name it is sent. The `Host` field comes from the client, and a server that answers unknown names with a default site will build links, redirects and cached pages from a name an attacker chose. RFC 9110 warns that the field is a frequent target for malware trying to "poison a shared cache or redirect a request to an unintended server". Answer only the names the host actually serves, refuse the rest, and build absolute links from configuration rather than from the request.

Cloned hosts share an identity. A machine copied from an image carries whatever the image held: SSH host keys, tokens, sometimes a backup passphrase. Clients can then no longer tell the copies apart, and one leak opens every copy. Generate host keys on first boot, keep secrets out of images, and give each host its own entry wherever a secret is configured.

Further reading

- host — NIST CSRC Glossary — both definitions, including the one that sets hosts against routers and switches.
- RFC 1122: Requirements for Internet Hosts, Communication Layers — the 1989 definition of a host, multihoming, and forwarding switched off by default.
- RFC 1123: Requirements for Internet Hosts, Application and Support — the rules for host names and their lengths, in section 2.1.
- RFC 9110, HTTP Semantics: Host and :authority — how one server tells many host names apart, and why the field is a target.

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 a host?” 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/host/