Engineer

Port

the number that picks a service on a host

What it is

A port is a 16-bit number in the header of every TCP and UDP segment that tells the receiving machine which of its programs the data belongs to. The address gets a packet to the right host; the port gets it to the right process once it is there. Engineers say port number when they want to be precise, and ports in the plural usually means the set a server listens on. The word is unrelated to a physical port on a switch, or to porting software to another platform.

Because the field is 16 bits wide there are 65,536 values, 0 to 65535, and TCP and UDP each have their own set: TCP port 53 and UDP port 53 are two separate things that DNS happens to use both of. The TCP specification, RFC 9293, defines a socket as an internet address joined to a port, and a connection as a pair of sockets. That makes four values: client address, client port, server address, server port. It also explains why one server port can carry thousands of conversations at once. Each conversation differs in at least one of the other three values, usually the client port.

RFC 6335 divides the range in three. Ports 0–1023 are System Ports, 1024–49151 are User Ports, and 49152–65535 are Dynamic Ports, which IANA never assigns. IANA keeps the registry that maps a service name to a number: 22 for SSH, 53 for DNS, 443 for HTTPS, 5432 for PostgreSQL. A registration is a convention that clients rely on, and any program can listen on any free number, which is how SSH ends up on a port nobody would guess.

The client side of a connection usually gets a port it never chose. When a program connects out, the operating system picks one from its ephemeral range, and Linux uses 32768–60999 by default rather than the IANA dynamic range. Asking to bind port 0 means the same thing: give me any free one.

Use the concept in a decision

Put “What is a port?” to work in a real decision. We can help relate the concept to the system you are building or buying. Wi-Fi, firewalls and remote access set up so staff can work from anywhere and only the right people get in.

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. Managed 30 Ubuntu Linux VPS instances — 30 Linux servers managed, with recovery planned for

Managed 30 Ubuntu Linux VPS instances Get in touch →

Why it matters

The decision behind every port is exposure: which programs answer from outside, and on which address. A port with no listener earns a refusal. A port with a listener is a piece of software that anyone who can reach the address can talk to, bugs included. The cheapest service to secure is one listening on 127.0.0.1, because packets from other machines cannot reach the loopback address, whatever the firewall says.

So the first question for each service is whether it needs to be reachable from outside at all. A database that only the application on the same machine talks to belongs on loopback. A tool only administrators use can stay on loopback and be reached through a jump host or a VPN. What remains, typically 80 and 443 for a website and one port for SSH, is the public surface. Keeping it short is a cost decision as much as a security one, because each public listener needs patching, logging and a firewall rule that matches it.

The second question is the number. Standard ports cost nothing in configuration: browsers, clients and restrictive networks expect 443 and pass it. Moving SSH off 22 cuts the noise from scanners that try only the default, which makes logs easier to read, but a scan of the full range finds it anyway, and every tool that refers to the port has to learn the new number. That last part is the one teams forget.

The third is privilege and sharing. On Linux a process needs root or the CAP_NET_BIND_SERVICE capability to bind a port below 1024, unless ip_unprivileged_port_start has been lowered from its default of 1024. And normally one program owns a given address and port, so two services that both want 443 need something in front to split the traffic, usually a reverse proxy that reads the host name the client asked for.

What it looks like in practice

When the company set up its own git forge in July 2026, the forge's SSH listener was bound to loopback at port 23231, and anyone outside reaches it by way of a jump host, so the count of public ports stayed where it was. A multiplexer that would serve HTTPS and git over SSH from one public port, by looking at the first bytes of each connection, is written and switched off. A listener with no users would still have to be defended.

In August 2026 a hardening review found that SSH had moved to port 1986 during the host's first hardening, while the intrusion-banning jail went on watching 22. Each of the jail's bans blocked a port that had no listener, so the protection had never once applied. The fix was small. What stayed was a rule: every security control is paired with an assertion that exercises it.

The host has 464 MB of memory and no dashboard, so its questions are answered by seven read-only reporting roles that render the live machine to Markdown. The facts report lists the listening sockets, which is the port inventory the kernel actually holds rather than the one the configuration describes. Among the real things the reports turned up was a listening port that no declaration mentioned.

Where it goes wrong

A service meant for the machine itself answers the whole internet. Many programs listen on every address by default, shown as `0.0.0.0`, `[::]` or `*` in the socket list, and a database or admin panel meant to be local is then one firewall mistake away from exposure. Set the bind address explicitly to `127.0.0.1` or `::1` and confirm it in the local address column:

```
ss -tlnp
ss -ulnp
```

A protection keeps watching a port the service has left. The service moved to a new number, and the firewall rule, the banning jail, the monitoring probe or a client configuration still names the old one. Every one of those tools keeps working correctly against a port where no daemon listens. Keep the number in one variable that every consumer reads, and add a check that compares the listening socket with what each control is watching.

Outgoing connections start to fail under load with "Cannot assign requested address". Every outgoing connection takes a local port from the ephemeral range, and a closed connection keeps its four values reserved in TIME_WAIT for a while afterwards. With Linux's default range of 32768–60999, one client address can hold 28,232 connections to the same server address and port, open or recently closed. A proxy or worker that opens a fresh connection per request to one backend runs through them. Reuse connections through keep-alive or a pool first; widening the range or adding source addresses only buys headroom.

A service started as an ordinary user stops with "Permission denied" on port 80. Binding below 1024 needs privilege. Give the service the CAP_NET_BIND_SERVICE capability through its unit file, put it on a high port behind a reverse proxy, or, on a host where every local user is trusted, lower ip_unprivileged_port_start. Running the whole service as root to get one port is the worst of the trades.

Startup stops with "Address already in use". Another process holds the port, often an old copy of the same service that is still shutting down, or a second service with the same default (8080 is a popular one). `ss -tlnp` names the owner. Decide which service owns the number and give each one an explicit port in its configuration instead of relying on defaults.

Further reading

- RFC 9293, section 4: Glossary — the TCP specification's own definitions of port, socket and connection.
- RFC 6335: IANA Procedures for the Management of the Service Name and Transport Protocol Port Number Registry — the three port ranges and how numbers in them are assigned.
- Service Name and Transport Protocol Port Number Registry — IANA's list of which service is registered on which number.
- RFC 768: User Datagram Protocol — the three-page UDP specification, source and destination ports included.
- RFC 6056: Recommendations for Transport-Protocol Port Randomization — how a client should pick its ephemeral ports so they are hard to guess.
- IP Sysctl — Linux kernel settings, among them ip_local_port_range and ip_unprivileged_port_start.

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 port?” 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/port/