CIDR and Subnet Mask Cheatsheet

By Dejan Panovski • Updated on Nov 2, 2026

IPv4 CIDR prefix to subnet mask table from /0 to /32 with address and host counts, subnet boundaries, reserved ranges, IPv6 prefix sizes, and CIDR in Linux commands.

A CIDR prefix such as `/24` tells you how many leading bits of an IP address identify the network. This cheatsheet maps every IPv4 prefix to its subnet mask and host count, and covers subnet boundaries, reserved ranges, common IPv6 prefix sizes, and the places CIDR shows up in Linux commands.

CIDR Basics

An IPv4 address has 32 bits. The prefix counts the bits that belong to the network. Host counts assume conventional unicast subnets and subtract only network and broadcast addresses; reserved ranges and provider reservations can reduce availability.

`192.168.10.0/24`	Network `192.168.10.0` with a 24-bit prefix
`Prefix length`	Leading network bits, from `/0` to `/32`
`Host bits`	`32 - prefix`; a `/24` leaves 8
`Total addresses`	`2^(32 - prefix)`; a `/27` has 32
`Usable hosts`	Total minus 2 (network and broadcast) for `/0` to `/30`
`Block size`	256 minus the mask octet where the prefix ends; for `/27`, `256 - 224 = 32`
`Wildcard mask`	`255.255.255.255` minus the mask; a `/26` is `0.0.0.63`
`Subnets from a split`	`2^(new prefix - old prefix)`; a `/24` holds four `/26` blocks

Prefixes /25 to /32

The prefix ends in the last octet. These sizes cover small LANs, point-to-point links, and host routes.

`/25 255.255.255.128`	128 addresses, 126 usable hosts
`/26 255.255.255.192`	64 addresses, 62 usable hosts
`/27 255.255.255.224`	32 addresses, 30 usable hosts
`/28 255.255.255.240`	16 addresses, 14 usable hosts
`/29 255.255.255.248`	8 addresses, 6 usable hosts
`/30 255.255.255.252`	4 addresses, 2 usable hosts
`/31 255.255.255.254`	2 addresses, both usable on point-to-point links (RFC 3021)
`/32 255.255.255.255`	1 address, a single host route

Prefixes /17 to /24

The prefix ends in the third octet. Each step to a shorter prefix doubles the block.

`/17 255.255.128.0`	32,768 addresses, 32,766 usable hosts
`/18 255.255.192.0`	16,384 addresses, 16,382 usable hosts
`/19 255.255.224.0`	8,192 addresses, 8,190 usable hosts
`/20 255.255.240.0`	4,096 addresses, 4,094 usable hosts
`/21 255.255.248.0`	2,048 addresses, 2,046 usable hosts
`/22 255.255.252.0`	1,024 addresses, 1,022 usable hosts
`/23 255.255.254.0`	512 addresses, 510 usable hosts
`/24 255.255.255.0`	256 addresses, 254 usable hosts

Prefixes /9 to /16

The prefix ends in the second octet. A `/16` holds 256 `/24` networks.

`/9 255.128.0.0`	8,388,608 addresses, 8,388,606 usable hosts
`/10 255.192.0.0`	4,194,304 addresses, 4,194,302 usable hosts
`/11 255.224.0.0`	2,097,152 addresses, 2,097,150 usable hosts
`/12 255.240.0.0`	1,048,576 addresses, 1,048,574 usable hosts
`/13 255.248.0.0`	524,288 addresses, 524,286 usable hosts
`/14 255.252.0.0`	262,144 addresses, 262,142 usable hosts
`/15 255.254.0.0`	131,072 addresses, 131,070 usable hosts
`/16 255.255.0.0`	65,536 addresses, 65,534 usable hosts

Prefixes /0 to /8

The prefix ends in the first octet. A `/0` matches every IPv4 address.

`/0 0.0.0.0`	4,294,967,296 addresses, every IPv4 address (the default route)
`/1 128.0.0.0`	2,147,483,648 addresses, 2,147,483,646 usable hosts
`/2 192.0.0.0`	1,073,741,824 addresses, 1,073,741,822 usable hosts
`/3 224.0.0.0`	536,870,912 addresses, 536,870,910 usable hosts
`/4 240.0.0.0`	268,435,456 addresses, 268,435,454 usable hosts
`/5 248.0.0.0`	134,217,728 addresses, 134,217,726 usable hosts
`/6 252.0.0.0`	67,108,864 addresses, 67,108,862 usable hosts
`/7 254.0.0.0`	33,554,432 addresses, 33,554,430 usable hosts
`/8 255.0.0.0`	16,777,216 addresses, 16,777,214 usable hosts

Subnet Boundaries in a /24

A network address must start on one of these values in the last octet.

`/25`	Block of 128, 2 subnets: `.0`, `.128`
`/26`	Block of 64, 4 subnets: `.0`, `.64`, `.128`, `.192`
`/27`	Block of 32, 8 subnets: `.0`, `.32`, `.64` … `.224`
`/28`	Block of 16, 16 subnets: `.0`, `.16`, `.32` … `.240`
`/29`	Block of 8, 32 subnets: `.0`, `.8`, `.16` … `.248`
`/30`	Block of 4, 64 subnets: `.0`, `.4`, `.8` … `.252`

Find the Network for an Address

Round the octet where the prefix ends down to a multiple of the block size, then set all following octets to zero. A `/20` has a block of 16 in the third octet, so 45 rounds down to 32.

`192.168.10.34/27`	Network `192.168.10.32`, broadcast `192.168.10.63`, hosts `.33` to `.62`
`172.16.45.10/20`	Network `172.16.32.0`, broadcast `172.16.47.255`
`10.20.30.40/12`	Network `10.16.0.0`, broadcast `10.31.255.255`

Split and Combine Networks

Splitting borrows host bits. Combining (route aggregation) needs contiguous blocks that start on the larger prefix boundary.

`Split 192.168.10.0/24 in two`	`/25`: `.0` and `.128`
`Split 192.168.10.0/24 in four`	`/26`: `.0`, `.64`, `.128`, `.192`
`Split 10.0.0.0/16 into /24 blocks`	256 networks, `10.0.0.0/24` to `10.0.255.0/24`
`Combine two /24 blocks`	One `/23` when the first third octet is even
`Combine 192.168.0.0/24 to 192.168.3.0/24`	`192.168.0.0/22`
`Combine 192.168.1.0/24 to 192.168.4.0/24`	No single `/22`; summarizes as `192.168.1.0/24`, `192.168.2.0/23`, `192.168.4.0/24`
`Check a /22 start`	The third octet is a multiple of 4

Private and Reserved IPv4 Ranges

Ranges you will meet in configs, routing tables, and documentation.

`10.0.0.0/8`	Private network (RFC 1918)
`172.16.0.0/12`	Private network, `172.16.0.0` to `172.31.255.255` (RFC 1918)
`192.168.0.0/16`	Private network (RFC 1918)
`100.64.0.0/10`	Carrier-grade NAT shared space (RFC 6598)
`127.0.0.0/8`	Loopback
`169.254.0.0/16`	Link-local; cloud metadata services listen on `169.254.169.254`
`0.0.0.0/0`	Every address; the default route
`224.0.0.0/4`	Multicast
`192.0.2.0/24, 198.51.100.0/24, 203.0.113.0/24`	Documentation examples (RFC 5737)
`255.255.255.255/32`	Limited broadcast

IPv6 Prefixes

IPv6 addresses have 128 bits and no broadcast address.

`/128`	Single address
`/127`	Point-to-point link (RFC 6164)
`/64`	Standard subnet; SLAAC needs a `/64`
`/56`	Common delegation to a home or small site; 256 `/64` subnets
`/48`	Common delegation to a site; 65,536 `/64` subnets
`::/0`	Default route
`::1/128`	Loopback
`fe80::/10`	Link-local
`fc00::/7`	Unique local addresses (`fd00::/8` in practice)
`ff00::/8`	Multicast
`2001:db8::/32`	Documentation examples

CIDR in Linux Commands

Commands and rules that take or print a prefix.

`ip -brief address`	Show each interface address with its prefix
`sudo ip addr add 192.168.10.34/27 dev eth0`	Assign an address with a `/27` prefix
`ip route show`	Show routes; the connected route is the network, `192.168.10.32/27`
`sudo ip route add 10.20.0.0/16 via 192.168.10.33`	Route a block through a gateway
`sudo ufw allow from 192.168.10.0/24 to any port 22`	Allow SSH from one subnet
`sudo nmap -sn 192.168.10.0/24`	Find live hosts in a subnet
`ipcalc 192.168.10.34/27`	Print the network, broadcast, and host range (`ipcalc` package)
`python3 -c "import ipaddress; print(ipaddress.ip_network('192.168.10.34/27', strict=False))"`	Print the network for an address; prints `192.168.10.32/27`

Troubleshooting

Quick checks for common CIDR mistakes.

`Error: Invalid prefix for given prefix length.`	`ip route add` needs the network address; use `192.168.10.32/27`, not `192.168.10.34/27`
`Error: Nexthop has invalid gateway.`	The gateway is outside every connected subnet; check the prefix on the interface
`WARN: Rule changed after normalization`	`ufw` cleared the host bits; `192.168.10.34/27` became `192.168.10.32/27`
`ValueError: ... has host bits set`	Python `ipaddress` is strict; pass `strict=False` or use the network address
`Address added without a prefix`	`ip addr add 10.0.0.5 dev eth0` assigns a `/32` and adds no connected route
`Two hosts on one LAN cannot reach each other`	Compare prefixes; with a mismatched mask, one host sends the traffic to its gateway

Related Guides

Use these guides for full addressing and firewall workflows.

CIDR Notation and Subnetting Explained	How prefixes, masks, and host counts work
Subnet Calculator	Calculate ranges and split a block in the browser
IPv4 vs IPv6	Differences between the two protocols
ip Command in Linux	Manage addresses and routes
How to Find Your IP Address in Linux	Check interface addresses and prefixes
ufw Command in Linux	Allow and deny traffic by subnet
IP Command Cheatsheet	`ip` addresses, routes, and links at a glance