# The Subnet Cheat Sheet

*A complete IPv4 and IPv6 subnetting reference for engineers.*

## 01 Core concepts

*The vocabulary you need*

| | |
|---|---|
| **IP Address** | A 32-bit number written as four 8-bit octets in dotted decimal (192.168.1.10). Each octet ranges 0-255. |
| **Subnet Mask** | A 32-bit pattern where 1s mark the network portion and 0s mark the host portion. /24 = 255.255.255.0. |
| **CIDR Notation** | IP followed by a slash and the prefix length: 10.0.0.0/24 means 24 network bits, 8 host bits. |
| **Network Address** | First address in a subnet (all host bits zero). Identifies the subnet itself - not assignable. |
| **Broadcast Address** | Last address in a subnet (all host bits one). Reaches all hosts on the subnet - not assignable. |
| **Usable Hosts** | Total addresses minus reserved IPs. RFC standard: subtract 2. AWS / Azure: subtract 5. GCP: subtract 4. |
| **Wildcard Mask** | The bitwise NOT of a subnet mask. Used in Cisco ACLs. /24 inverts to wildcard 0.0.0.255. |
| **VLSM** | Variable-Length Subnet Masking - using different prefix sizes within one parent CIDR. Allocate largest first. |
| **Supernet** | A larger CIDR that aggregates multiple smaller adjacent subnets. Same as route summarization in BGP. |

## 02 RFC 1918 private ranges

*Not routable on the public internet - use freely internally*

| CIDR | Subnet Mask | Total Addresses | Common Use |
|---|---|---|---|
| 10.0.0.0/8 | 255.0.0.0 | 16,777,216 | Large enterprises, cloud VPCs |
| 172.16.0.0/12 | 255.240.0.0 | 1,048,576 | Mid-sized networks, Docker default |
| 192.168.0.0/16 | 255.255.0.0 | 65,536 | Home & small office (/24 typical) |

## 03 Complete IPv4 prefix reference

*/0 through /32 with masks, wildcards, and host counts*

| CIDR | Subnet Mask | Wildcard Mask | Total | Usable Hosts |
|---|---|---|---|---|
| /0 | 0.0.0.0 | 255.255.255.255 | 4,294,967,296 | 4,294,967,294 |
| /1 | 128.0.0.0 | 127.255.255.255 | 2,147,483,648 | 2,147,483,646 |
| /2 | 192.0.0.0 | 63.255.255.255 | 1,073,741,824 | 1,073,741,822 |
| /3 | 224.0.0.0 | 31.255.255.255 | 536,870,912 | 536,870,910 |
| /4 | 240.0.0.0 | 15.255.255.255 | 268,435,456 | 268,435,454 |
| /5 | 248.0.0.0 | 7.255.255.255 | 134,217,728 | 134,217,726 |
| /6 | 252.0.0.0 | 3.255.255.255 | 67,108,864 | 67,108,862 |
| /7 | 254.0.0.0 | 1.255.255.255 | 33,554,432 | 33,554,430 |
| /8 | 255.0.0.0 | 0.255.255.255 | 16,777,216 | 16,777,214 |
| /9 | 255.128.0.0 | 0.127.255.255 | 8,388,608 | 8,388,606 |
| /10 | 255.192.0.0 | 0.63.255.255 | 4,194,304 | 4,194,302 |
| /11 | 255.224.0.0 | 0.31.255.255 | 2,097,152 | 2,097,150 |
| /12 | 255.240.0.0 | 0.15.255.255 | 1,048,576 | 1,048,574 |
| /13 | 255.248.0.0 | 0.7.255.255 | 524,288 | 524,286 |
| /14 | 255.252.0.0 | 0.3.255.255 | 262,144 | 262,142 |
| /15 | 255.254.0.0 | 0.1.255.255 | 131,072 | 131,070 |
| /16 | 255.255.0.0 | 0.0.255.255 | 65,536 | 65,534 |
| /17 | 255.255.128.0 | 0.0.127.255 | 32,768 | 32,766 |
| /18 | 255.255.192.0 | 0.0.63.255 | 16,384 | 16,382 |
| /19 | 255.255.224.0 | 0.0.31.255 | 8,192 | 8,190 |
| /20 | 255.255.240.0 | 0.0.15.255 | 4,096 | 4,094 |
| /21 | 255.255.248.0 | 0.0.7.255 | 2,048 | 2,046 |
| /22 | 255.255.252.0 | 0.0.3.255 | 1,024 | 1,022 |
| /23 | 255.255.254.0 | 0.0.1.255 | 512 | 510 |
| /24 | 255.255.255.0 | 0.0.0.255 | 256 | 254 |
| /25 | 255.255.255.128 | 0.0.0.127 | 128 | 126 |
| /26 | 255.255.255.192 | 0.0.0.63 | 64 | 62 |
| /27 | 255.255.255.224 | 0.0.0.31 | 32 | 30 |
| /28 | 255.255.255.240 | 0.0.0.15 | 16 | 14 |
| /29 | 255.255.255.248 | 0.0.0.7 | 8 | 6 |
| /30 | 255.255.255.252 | 0.0.0.3 | 4 | 2 |
| /31 | 255.255.255.254 | 0.0.0.1 | 2 | 2 |
| /32 | 255.255.255.255 | 0.0.0.0 | 1 | 1 |

**How to read this table**

Usable = Total - 2 (one each for network and broadcast). For cloud providers, see the next page. /31 (point-to-point per RFC 3021) and /32 (host route) are special cases.

## 04 Cloud provider reserved IPs

*Real usable hosts per subnet by provider*

| Provider | Reserved | Minimum | What gets reserved |
|---|---|---|---|
| AWS VPC | 5 | /28 | Network - VPC router - DNS - future use - broadcast |
| Azure VNet | 5 | /29 | Network - default gateway - 2x DNS - broadcast |
| GCP | 4 | /29 | Network - gateway - second-to-last - broadcast |
| OCI | 3 | /30 | Network - gateway - broadcast |
| Kubernetes | 2 | /24 typical | Network - broadcast (per-node CIDR separate) |
| Standard (RFC) | 2 | /30 | Network - broadcast (on-prem / textbook) |

## 05 Usable hosts by prefix x provider

*Quick capacity-planning lookup*

| Provider | /24 | /25 | /26 | /27 | /28 | /29 |
|---|---|---|---|---|---|---|
| Standard | 254 | 126 | 62 | 30 | 14 | 6 |
| AWS | 251 | 123 | 59 | 27 | 11 | 3 |
| Azure | 251 | 123 | 59 | 27 | 11 | 3 |
| GCP | 252 | 124 | 60 | 28 | 12 | 4 |
| OCI | 253 | 125 | 61 | 29 | 13 | 5 |
| Kubernetes | 254 | 126 | 62 | 30 | 14 | 6 |

**Why this matters**

A /28 on textbook math gives you 14 hosts. On AWS, you get 11. On Azure, also 11. On GCP, 12. Capacity plans built on the wrong number cause deployments to fail with *InsufficientFreeAddressesInSubnet* - usually at the worst possible moment.

## 06 Quick formulas

*The math you reach for daily*

| | |
|---|---|
| Total addresses in /n | 2^(32 - n) |
| Usable hosts (RFC standard) | 2^(32 - n) - 2 |
| Usable hosts (AWS / Azure) | 2^(32 - n) - 5 |
| Usable hosts (GCP) | 2^(32 - n) - 4 |
| Network address | IP AND subnet_mask |
| Broadcast address | network OR wildcard_mask |
| Wildcard mask | NOT subnet_mask |
| Smallest prefix for N hosts | 32 - ceil(log2(N + reserved)) |
| Magic number method | 256 - relevant_mask_octet |
| CIDR overlap check | a.start <= b.end AND b.start <= a.end |

## 07 VLSM worked example

*Parent 10.0.0.0/22 - five departments - largest-first*

| Department | Need | Prefix | Allocated CIDR | Usable |
|---|---|---|---|---|
| Engineering | 250 hosts | /24 | 10.0.0.0/24 | 254 |
| Sales | 100 hosts | /25 | 10.0.1.0/25 | 126 |
| Operations | 50 hosts | /26 | 10.0.1.128/26 | 62 |
| DMZ | 25 hosts | /27 | 10.0.1.192/27 | 30 |
| Management | 10 hosts | /28 | 10.0.1.224/28 | 14 |

## 08 Cloud-specific gotchas

*Things that bite in production*

| Provider | Tip |
|---|---|
| AWS | Enable prefix delegation on VPC CNI for denser EKS pod packing. |
| AWS | ALB needs at least 8 IPs per AZ subnet - don't use a /28. |
| Azure | Some Azure services require dedicated subnets (App Gateway, Bastion). |
| GCP | GCP VPC is global by default; subnets are per-region. Plan accordingly. |
| OCI | Smallest reserved overhead (3 IPs) - efficient for small allocations. |
| K8s | Pod CIDR overlapping with VPC range breaks pod-to-internet routing. |

## 09 Common pitfalls

*Things that break in production*

- ! Forgetting cloud reserved IPs - a /28 has 14 usable on paper, only 11 on AWS.
- ! Subnets that don't start on a boundary (network must be a multiple of subnet size).
- ! Allocating smallest-first in VLSM - causes fragmentation, leaves gaps too small to fill.
- ! Two VPCs both using 10.0.0.0/16 - they cannot peer.
- ! No headroom - /24 with 250 hosts today, no room to grow tomorrow. Use the 2x rule.
- ! EKS pod CIDR overlapping with VPC range - pods can't reach the internet.
- ! Putting load balancers in a /28 - AWS ALB needs 8 IPs per AZ.
- ! Cisco ACL using subnet mask instead of wildcard mask.
- ! Forgetting that /31 is point-to-point (both addresses usable) per RFC 3021.
- ! Underestimating Kubernetes Service CIDR - default /12 is plenty, /20 is still big.

## 10 IPv6 quick reference

*The bare minimum to be dangerous*

| | |
|---|---|
| Address size | 128 bits, written as 8 groups of 4 hex digits |
| Standard subnet | /64 (required for SLAAC) |
| Customer site | /56 - contains 256 /64s |
| Org allocation | /48 - contains 65,536 /64s |
| Link-local | fe80::/10 - every interface auto-generates one |
| Unique local | fc00::/7 (use fd00::/8) - IPv6 RFC 1918 equivalent |
| Global unicast | 2000::/3 - your ISP assigns from this space |
| No broadcast | Multicast (ff00::/8) replaces it |
| Loopback | ::1 - same as 127.0.0.1 in IPv4 |
| Unspecified | :: - the "any" address |