Network Commands Cheatsheet

A field reference for diagnosing, capturing, and analysing network traffic — covering ip, ss, dig, nmap, tcpdump, wireshark/tshark, and more.

Interface & Routing (iproute2)

`ip a`	Show all interfaces and IPs
`ip a show eth0`	Single interface
`ip link set eth0 up/down`	Toggle interface
`ip addr add 10.0.0.5/24 dev eth0`	Add IP
`ip addr del 10.0.0.5/24 dev eth0`	Remove IP
`ip r`	Routing table
`ip r get 8.8.8.8`	Which interface/route reaches this IP?
`ip route add default via 10.0.0.1`	Set default gateway
`ip neigh`	ARP table (modern)
`arp -an`	ARP table (legacy)
`ip -s link`	Per-interface counters (rx/tx, drops, errors)
`ethtool eth0`	Link speed/duplex/driver

Sockets & Connections

`ss -tulpn`	Listening TCP/UDP with PIDs
`ss -tn state established`	Active TCP connections
`ss -tn dst :443`	All connections to port 443
`ss -p sport :22`	Connections from your SSH port
`ss -s`	Socket summary
`lsof -i :8080`	Process owning port 8080
`lsof -i tcp:80 -sTCP:ESTABLISHED`	Established to port 80
`netstat -rn`	Routing table (legacy)
`netstat -i`	Interface stats (legacy)
`fuser 80/tcp`	PID using TCP port 80

Reachability & Path

`ping -c 4 host`	Send 4 ICMP echoes
`ping -i 0.2 -c 50 host`	0.2s interval, 50 packets
`ping -M do -s 1472 host`	Path MTU discovery
`traceroute host`	Hop-by-hop path
`traceroute -T -p 443 host`	TCP traceroute on port 443

`mtr host`	Live combined ping + traceroute
`mtr -rwzbc 100 host`	One-shot 100-packet report
`hping3 -S -p 80 host`	Crafted TCP SYN probe

DNS

`dig +short example.com`	Just the answer IP(s)
`dig example.com ANY`	All record types
`dig +trace example.com`	Walk from root → authoritative
`dig @1.1.1.1 example.com`	Query specific resolver
`dig -x 1.2.3.4`	Reverse DNS lookup
`dig MX example.com +short`	Mail server records
`dig TXT example.com`	TXT records (SPF, DMARC, etc.)
`host -t SOA example.com`	Authoritative server
`nslookup example.com`	Classic interactive lookup
`drill / kdig`	Modern dig replacements

HTTP Probing

`curl -I url`	HEAD only — see headers
`curl -v url`	Verbose request and response
`curl -L url`	Follow redirects
`curl -o file url`	Save to file
`curl -X POST -d 'k=v' url`	POST form data
`curl -H "X-Token: abc" url`	Custom header
`curl -w "@fmt.txt" -o /dev/null url`	Detailed timing breakdown
`curl --resolve host:443:1.2.3.4 url`	Override DNS for one request
`wget -c url`	Download with resume support
`httpie / xh`	Friendlier modern alternatives

Port Scanning (nmap)

`nmap -sn 10.0.0.0/24`	Ping sweep — find live hosts
`nmap -sS -p- target`	Full TCP SYN scan all 65k ports
`nmap -sV -A target`	Version + OS + scripts
`nmap -sU -p 53,123,161 target`	UDP scan common ports
`nmap -p 80,443 --script http-headers t`	Run NSE script
`nmap -F target`	Top 100 ports (fast)
`nmap --top-ports 1000 target`	Most common 1000
`nmap -oA scan target`	Save in all output formats
`nmap -Pn target`	Skip ICMP host discovery
`nmap -T4 target`	Aggressive timing

Packet Capture (tcpdump)

`tcpdump -i any`	Capture all interfaces
`tcpdump -i eth0 -nn`	No name resolution
`tcpdump -w out.pcap`	Write to file
`tcpdump -r in.pcap`	Read from file
`tcpdump host 1.2.3.4`	Filter by host
`tcpdump port 53`	Filter by port
`tcpdump src 10.0.0.5`	Source IP filter
`tcpdump tcp port 80 and host x`	Compound filter
`tcpdump -A port 80`	Print ASCII payload
`tcpdump -X port 80`	Hex + ASCII
`tcpdump -c 100 ...`	Stop after 100 packets

`tcpdump -G 60 -W 24 -w "%H.pcap"`	Hourly rotation, keep 24

tshark (Wireshark CLI)

`tshark -i any`	Live capture
`tshark -r in.pcap -Y "http"`	Display filter on file
`tshark -r in.pcap -T fields -e ip.src`	Extract single field
`tshark -z conv,tcp -r in.pcap`	TCP conversation summary
`tshark -z io,stat,1 -r in.pcap`	I/O stats every 1 second
`wireshark -r in.pcap`	Open in GUI

Throughput & Latency Testing

`iperf3 -s`	Run iperf server
`iperf3 -c host -t 30`	TCP test against server, 30s
`iperf3 -c host -u -b 100M`	UDP test at 100 Mbps
`iperf3 -c host -P 8`	Parallel streams
`netcat / nc -lvp 4444`	Listen on port 4444
`nc -vz host 1-1024`	TCP port range scan
`socat TCP-LISTEN:8080,fork TCP:host:80`	Generic relay

Firewall Quick Reference

`iptables -L -n -v`	List rules with counters
`iptables -A INPUT -p tcp --dport 22 -j ACCEPT`	Allow SSH in
`iptables -P INPUT DROP`	Default-deny incoming
`iptables-save > rules.v4`	Persist rules
`nft list ruleset`	Show nftables rules (modern)
`ufw allow 80,443/tcp`	UFW shortcut
`firewall-cmd --add-port=8080/tcp --permanent`	firewalld syntax