SS Inspect sockets

START HERE

`ss -tulpn`
Listening TCP/UDP + process

`ss -tnp state established`
Connections with process

`ss -tn`
Established TCP connections

`ss -tn dst <ip>`
Connections to a host

`ss -ltn 'sport = :<port>'`
Who listens on a port?

`ss -ti`
TCP internals (rtt, cwnd)

`ss -s`
Socket totals

`ss -lx`
Listening Unix sockets

Socket type -t -u -x -4 -6 → **`state / -l / -a`** → **State filter** state established → **`expression`** → **Address filter** sport = :443 → **`-n -p -e -i`** → **Output** columns + extras

CONCEPTS

With no options ss shows only connected sockets. Add -l for listening ones, -a for both.

Process info (-p) only shows sockets you own unless you run ss as root.

SELECT SOCKETS

`ss -t`
TCP connections

`ss -u`
UDP sockets

`ss -x`
Unix domain sockets

`ss -l`
Only listening sockets

`ss -a`
Listening and connected

`ss -4 / ss -6`
Only IPv4 / IPv6

`ss -w`
Raw sockets

`ss -0`
Packet sockets

`ss -A <type>,<type>`
Pick socket types by name
Example: ss -A tcp,udp. Same selection as -t -u.

`ss -f <family>`
Only one address family
inet, inet6, link, unix, netlink, vsock, tipc, xdp.

`ss --vsock`
VM sockets

OUTPUT CONTROL

`ss -n`
Numeric ports and addresses

`ss -p`
Show owning process

`ss -r`
Resolve host names

`ss -H`
No header line

`ss -O`
One line per socket

OUTPUT CONTROL (cont.)

`ss -Q`
Hide Recv-Q and Send-Q

`ss -e`
Extended info (uid, inode)

`ss -o`
Show timers (keepalive, timewait)

`ss -m`
Socket memory usage

`ss -i`
TCP internals (rtt, cwnd, retrans)

`ss -T`
Show threads, not just processes

`ss -Z`
SELinux context of the process

`ss --cgroup`
Show cgroup of each socket

`ss -s`
Summary of socket counts

FILTER BY STATE

`ss -t state established`
Only established

`ss -t state listening`
Only listening TCP

`ss -t state time-wait`
Sockets in TIME-WAIT

`ss -t state close-wait`
Peer closed, app has not
Many CLOSE-WAIT sockets usually mean the application forgets to close them.

`ss -t state syn-recv`
Half-open incoming handshakes

`ss -t state connected`
Any non-listening state

`ss -t exclude <state>`
Everything except a state
Example: ss -t exclude time-wait.

`ss -t state bucket`
syn-recv and time-wait

FILTER BY ADDRESS

`ss -tn 'sport = :<port>'`
Local port equals

FILTER BY ADDRESS (cont.)

`ss -tn 'dport = :<port>'`
Remote port equals

`ss -tn dst <ip>`
Remote address

`ss -tn src <ip>`
Local address

`ss -tn dst <net>/<mask>`
Remote subnet

`ss -tn dst <ip>:<port>`
Remote host and port

`ss -tn 'sport > :1024'`
Port comparison
Operators: = != > >= < <=, or eq ne gt ge lt le. Quote them in the shell.

`ss -tn '( sport = :80 or sport = :443 )'`
Combine with or / and
Parentheses need quoting or escaping; 'not' negates.

`ss -tn 'dport = :ssh'`
Service name instead of number

`ss -x src <path>`
Unix socket by path

`ss -tn dev <iface>`
Sockets bound to an interface

`ss -tn fwmark <mark>`
Match packet firewall mark

`ss -tn autobound`
Auto-bound local port

ADVANCED

`ss -E`
Stream sockets as they are destroyed

`ss -K <filter>`
Force-close matching sockets
Needs root and a kernel built with CONFIG_INET_DIAG_DESTROY.

`ss -N <netns>`
Inspect a named network namespace

`ss -B`
TCP sockets bound but inactive

`ss --inet-sockopt`
Show inet socket options

`ss -b`
BPF filters on sockets

ADVANCED (cont.)

`ss -D <file>`
Dump raw TCP data to a file

`ss -F <file>`
Read filter from a file

READING THE TABLE

Columns

Column	Meaning
`Netid`	Socket type: tcp, udp, u_str
`State`	LISTEN, ESTAB, TIME-WAIT...
`Recv-Q`	Bytes not yet read by the app
`Send-Q`	Bytes not yet acked by the peer
`Local Address:Port`	This end of the socket
`Peer Address:Port`	Remote end; * if none
`Process`	users:((name,pid,fd)) with -p

On a LISTEN socket

Column	Means
`Recv-Q`	Connections waiting for accept()
`Send-Q`	Configured backlog size

Local 0.0.0.0 or [::] means all interfaces; 127.0.0.1 is reachable only from this host.

Recv-Q close to Send-Q on a LISTEN socket means the accept queue is full.

RECIPES

What is using port 8080?

```
sudo ss -ltnp 'sport = :8080'
# no output means nothing is listening
```

Count connections per state

```
ss -tanH | awk '{print $1}' | sort | uniq -c | sort -rn
```

Top remote hosts by connections

```
ss -tnH state established | awk '{print $4}' \
  | cut -d: -f1 | sort | uniq -c | sort -rn | head
```

Watch listeners live

```
watch -n 1 'ss -tulpn'
# slow connection: ss -ti dst <ip>
```