# SS Inspect 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

**ss -f** *<family>*
Only one address family

## 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

**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 -s**
Summary of socket counts

## FILTER BY STATE

**ss -t state established**
Only established

**ss -t state listening**
Only listening TCP

## FILTER BY STATE (cont.)

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

**ss -t state close-wait**
Peer closed, app has not

**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

## FILTER BY ADDRESS

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

**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

## FILTER BY ADDRESS (cont.)

**ss -tn '( sport = :80 or sport = :443 )'**
Combine with or / and

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

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

## 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 |

## READING THE TABLE (cont.)

### 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.

## 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>
```