sanfoundry.com

15+ "ss" Command Usage Examples in Linux

by linuxcmd2

15-19 minutes

This tutorial explains Linux “ss” command, options and its usage with examples.

ss – socket statistics

Description :

ss is used to dump socket statistics. It allows showing information similar to netstat. It can display more TCP and state informations than other tools.

Usage :

ss [options] [FILTER]

Options :

When no option is used ss displays a list of open non-listening TCP sockets that have established connection.

-h, –help

Show summary of options.

-V, –version

Output version information.

-n, –numeric

Do not try to resolve service names.

-r, –resolve

Try to resolve numeric address/ports.

-a, –all

Display both listening and non-listening (for TCP this means established connections) sockets.

-l, –listening

Display only listening sockets (these are omitted by default).

-o, –options

Show timer information.

-e, –extended

Show detailed socket information

-m, –memory

Show socket memory usage.

-p, –processes

Show process using socket.

-i, –info

Show internal TCP information.

-s, –summary

Print summary statistics. This option does not parse socket lists obtaining summary from various sources. It is useful when amount of sockets is so huge that parsing /proc/net/tcp is painful.

-Z, –context

As the -p option but also shows process security context.

-z, –contexts

As the -Z option but also shows the socket context. The socket context is taken from the associated inode and is not the actual socket context held by the kernel. Sockets are typically labeled with the context of the creating process, however the context shown will reflect any policy role, type and/or range transition rules applied, and is therefore a useful reference.

-b, –bpf

Show socket BPF filters (only administrators are allowed to get these information).

-4, –ipv4

Display only IP version 4 sockets (alias for -f inet).

-6, –ipv6

Display only IP version 6 sockets (alias for -f inet6).

-0, –packet

Display PACKET sockets (alias for -f link).

-t, –tcp

Display TCP sockets.

-u, –udp

Display UDP sockets.

-d, –dccp

Display DCCP sockets.

-w, –raw

Display RAW sockets.

-x, –unix

Display Unix domain sockets (alias for -f unix).

-f FAMILY, –family=FAMILY

Display sockets of type FAMILY. Currently the following families are supported: unix, inet, inet6, link, netlink.

-A QUERY, –query=QUERY, –socket=QUERY

List of socket tables to dump, separated by commas. The following identifiers are understood: all, inet, tcp, udp, raw, unix, packet, netlink, unix_dgram, unix_stream, unix_seqpacket, packet_raw, packet_dgram.

-D FILE, –diag=FILE

Do not display anything, just dump raw information about TCP sockets to FILE after applying filters. If FILE is – stdout is used.

-F FILE, –filter=FILE

Read filter information from FILE. Each line of FILE is interpreted like single command line option. If FILE is – stdin is used.

FILTER := [state TCP-STATE] [EXPRESSION] Please take a look at the official documentation (Debian package iproute-doc) for details regarding filters.

Examples :

1. List all connections

```
$ ss | less
Netid  State      Recv-Q Send-Q   Local
Address:Port        Peer Address:Port
u_str  ESTAB      0      0                    *
207499                * 207500
u_str  ESTAB      0      0      @/tmp/dbus-
HulwP2Cqbm 207393                 * 207392
u_str  ESTAB      0      0      @/tmp/.X11-unix/X0
206529                * 206528
u_str  ESTAB      0      0                    *
206446                * 206447
u_str  ESTAB      0      0      @/tmp/dbus-
HulwP2Cqbm 205775                 * 205774
u_str  ESTAB      0      0      @/tmp/dbus-
HulwP2Cqbm 205578                 * 205577
u_str  ESTAB      0      0      @/tmp/dbus-
HulwP2Cqbm 207082                 * 207081
u_str  ESTAB      0      0      @/dbus-vfs-daemon
/socket-eEA5oIcY 228375                 * 0
u_str  ESTAB      0      0                    *
```

```
206971                * 206972
u_str  ESTAB      0      0                     *
205301                * 205302
u_str  ESTAB      0      0      @/tmp/dbus-
HulwP2Cqbm 206668              * 206667
u_str  ESTAB      0      0      @/dbus-vfs-daemon
/socket-rCip3gc7 205882              * 205881
u_str  ESTAB      0      0                     *
205170                * 205171
u_str  ESTAB      0      0                     *
7967                  * 7968
....
```

2. Filter out tcp connections

```
$ ss -aA tcp
State      Recv-Q Send-Q      Local Address:Port
Peer Address:Port
LISTEN     0      5               127.0.1.1:domain
*:*
LISTEN     0      128             127.0.0.1:ipp
*:*
CLOSE-WAIT 1      0          192.168.42.250:58390
103.245.222.184:http
TIME-WAIT  0      0          192.168.10.148:56833
74.125.236.99:http
CLOSE-WAIT 1      0          192.168.10.140:35766
103.245.222.184:http
CLOSE-WAIT 1      0          192.168.42.250:58392
103.245.222.184:http
TIME-WAIT  0      0          192.168.10.148:49839
```

```
23.57.219.27:http
ESTAB      0      0          192.168.10.148:53060
173.194.36.41:https
CLOSE-WAIT 1      0          192.168.10.140:35765
103.245.222.184:http
TIME-WAIT  0      0          192.168.10.148:47000
74.125.28.100:http
CLOSE-WAIT 1      0          192.168.42.250:58391
103.245.222.184:http
TIME-WAIT  0      0          192.168.10.148:38878
173.194.36.46:http
CLOSE-WAIT 1      0          192.168.10.140:35763
103.245.222.184:http
CLOSE-WAIT 1      0          192.168.10.140:35764
103.245.222.184:http
CLOSE-WAIT 1      0          192.168.42.250:58389
103.245.222.184:http
LISTEN     0      128                   ::1:ipp
:::*
CLOSE-WAIT 1      0                     ::1:55327
::1:ipp
```

OR

```
$ ss -at
State      Recv-Q Send-Q      Local Address:Port
Peer Address:Port
LISTEN     0      5               127.0.1.1:domain
*:*
LISTEN     0      128             127.0.0.1:ipp
*:*
```

```
CLOSE-WAIT 1      0          192.168.42.250:58390
103.245.222.184:http
TIME-WAIT  0      0          192.168.10.148:56833
74.125.236.99:http
CLOSE-WAIT 1      0          192.168.10.140:35766
103.245.222.184:http
CLOSE-WAIT 1      0          192.168.42.250:58392
103.245.222.184:http
TIME-WAIT  0      0          192.168.10.148:49839
23.57.219.27:http
ESTAB      0      0          192.168.10.148:53060
173.194.36.41:https
CLOSE-WAIT 1      0          192.168.10.140:35765
103.245.222.184:http
TIME-WAIT  0      0          192.168.10.148:47000
74.125.28.100:http
CLOSE-WAIT 1      0          192.168.42.250:58391
103.245.222.184:http
TIME-WAIT  0      0          192.168.10.148:38878
173.194.36.46:http
CLOSE-WAIT 1      0          192.168.10.140:35763
103.245.222.184:http
CLOSE-WAIT 1      0          192.168.10.140:35764
103.245.222.184:http
CLOSE-WAIT 1      0          192.168.42.250:58389
103.245.222.184:http
LISTEN     0      128                   ::1:ipp
:::*
CLOSE-WAIT 1      0                     ::1:55327
::1:ipp
```

3. Filter out udp connections

```
$ ss -aA udp
State      Recv-Q Send-Q      Local Address:Port
Peer Address:Port
UNCONN     0      0                   *:58718
*:*
UNCONN     0      0           127.0.1.1:domain
*:*
UNCONN     0      0                   *:bootpc
*:*
UNCONN     0      0                   *:mdns
*:*
UNCONN     0      0                   *:27412
*:*
UNCONN     0      0                  :::62912
:::*
UNCONN     0      0                  :::mdns
:::*
UNCONN     0      0                  :::46372
:::*
```

OR

```
$ ss -au
State      Recv-Q Send-Q      Local Address:Port
Peer Address:Port
UNCONN     0      0                   *:58718
*:*
UNCONN     0      0           127.0.1.1:domain
*:*
UNCONN     0      0                   *:bootpc
```

```
*:*
UNCONN     0      0                        *:mdns
*:*
UNCONN     0      0                        *:27412
*:*
UNCONN     0      0                       :::62912
:::*
UNCONN     0      0                       :::mdns
:::*
UNCONN     0      0                       :::46372
:::*
```

4. Do not resolve hostname

To get the output faster, use the “n” option to prevent ss from resolving ip addresses to hostnames. But this will prevent resolution of port numbers as well.

```
$ ss -nt
State      Recv-Q Send-Q        Local Address:Port
Peer Address:Port
CLOSE-WAIT 1      0
192.168.42.250:58390      103.245.222.184:80
ESTAB      0      0
192.168.10.148:56390       63.245.216.132:443
CLOSE-WAIT 1      0
192.168.10.140:35766      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58392      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35765      103.245.222.184:80
CLOSE-WAIT 1      0
```

```
192.168.42.250:58391      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35763      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35764      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58389      103.245.222.184:80
CLOSE-WAIT 1      0
::1:55327                  ::1:631
```

5. Show only listening sockets

```
$ ss -lnt
State      Recv-Q Send-Q        Local Address:Port
Peer Address:Port
LISTEN     0      5                 127.0.1.1:53
*:*
LISTEN     0      128               127.0.0.1:631
*:*
LISTEN     0      128                     ::1:631
:::*
```

The above command lists out all “listening” “tcp” connections.

6. Print process name and pid

```
# ss -ltp
State      Recv-Q Send-Q      Local Address:Port
Peer Address:Port
LISTEN     0      5             127.0.1.1:domain
*:*        users:(("dnsmasq",1199,5))
LISTEN     0      128           127.0.0.1:ipp
*:*        users:(("cupsd",793,10))
```

```
LISTEN     0      128                    ::1:ipp
:::*        users:(("cupsd",793,9))
```

7. Print summary statistics

```
$ ss -s
Total: 648 (kernel 0)
TCP:   12 (estab 0, closed 0, orphaned 0, synrecv
0, timewait 0/0), ports 0

Transport Total     IP        IPv6
*         0         -         -
RAW       0         0         0
UDP       8         5         3
TCP       12        10        2
INET      20        15        5
FRAG      0         0         0
```

8. Display only IPv4 or IPv6 socket connections

To display only IPv4 socket connections use the ‘-f inet’ or ‘-4’ option.

```
$ ss -tl -f inet
State      Recv-Q Send-Q      Local Address:Port
Peer Address:Port
LISTEN     0      5               127.0.1.1:domain
*:*
LISTEN     0      128             127.0.0.1:ipp
*:*
```

To display only IPv6 connections use the ‘-f inet6’ or ‘-6’ option.

```
$ ss -tl6
```

```
State      Recv-Q Send-Q      Local Address:Port
Peer Address:Port
LISTEN     0      128                   ::1:ipp
:::*
```

9. To display all Ipv4 tcp sockets that are in “connected” state.

```
$ ss -t4 state established
Recv-Q Send-Q        Local Address:Port
Peer Address:Port
0      0               192.168.1.2:54436
165.193.246.23:https
0      0               192.168.1.2:43386
173.194.72.125:xmpp-client
0      0               192.168.1.2:38355
199.59.150.46:https
0      0               192.168.1.2:56198
108.160.162.37:http
```

10. To display all Ipv4 tcp sockets that are in “time-wait” state.

```
$ ss -t4 state time-wait
Recv-Q Send-Q        Local Address:Port
Peer Address:Port
0      0               192.168.1.2:42261
199.59.150.39:https
0      0                 127.0.0.1:43541
127.0.0.1:2633
```

Note: The state can be either of the following

1. established

2. syn-sent

3. syn-recv

4. fin-wait-1
5. fin-wait-2
6. time-wait
7. closed
8. close-wait
9. last-ack
10. closing
11. all – All of the above states
12. connected – All the states except for listen and closed
13. synchronized – All the connected states except for syn-sent
14. bucket – Show states, which are maintained as minisockets, i.e. time-wait and syn-recv.
15. big – Opposite to bucket state.

11. Display all socket connections with source or destination port of ssh.

```
$ ss -at '( dport = :ssh or sport = :ssh )'
State      Recv-Q Send-Q    Local Address:Port
Peer Address:Port
LISTEN     0      128                    *:ssh
*:*
LISTEN     0      128                   :::ssh
:::*
```

12. Display Sockets with destination port 443 or 80

```
$ ss -nt '( dst :443 or dst :80 )'
State      Recv-Q Send-Q        Local Address:Port
Peer Address:Port
CLOSE-WAIT 1      0
192.168.42.250:58390      103.245.222.184:80
```

```
CLOSE-WAIT 1      0
192.168.10.140:35766      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58392      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35765      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58391      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35763      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35764      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58389      103.245.222.184:80
```

13. Filter by address and port

```
$ ss -nt dst 103.245.222.184:80
State      Recv-Q Send-Q        Local Address:Port
Peer Address:Port
CLOSE-WAIT 1      0
192.168.42.250:58390      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35766      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58392      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35765      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58391      103.245.222.184:80
CLOSE-WAIT 1      0
```

```
192.168.10.140:35763      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35764      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58389      103.245.222.184:80
```

14. Filtering by ports only

```
$ ss -nt dport = :80
State      Recv-Q Send-Q        Local Address:Port
Peer Address:Port
CLOSE-WAIT 1      0
192.168.42.250:58390      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35766      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58392      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35765      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58391      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35763      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.10.140:35764      103.245.222.184:80
CLOSE-WAIT 1      0
192.168.42.250:58389      103.245.222.184:80
```

15. Display sockets with remote ports less than 100

16. Display sockets with port numbers greater than 25

```
# sudo ss -nt sport gt :1024
```

17. Display sockets with connections to remote port 80

```
# sudo ss -nt state connected dport = :80
```

Sanfoundry Global Education & Learning Series – 1000 Linux Tutorials.