SS Command in Linux with Useful Examples

By Bobbin Zachariah | Updated September 24, 2021 | Commands

The ss tool is a CLI command used to display information about the network socket in Linux. The ss stands for socket statistics. It is a similar tool to netstat, which can display more information such as TCP and state information.

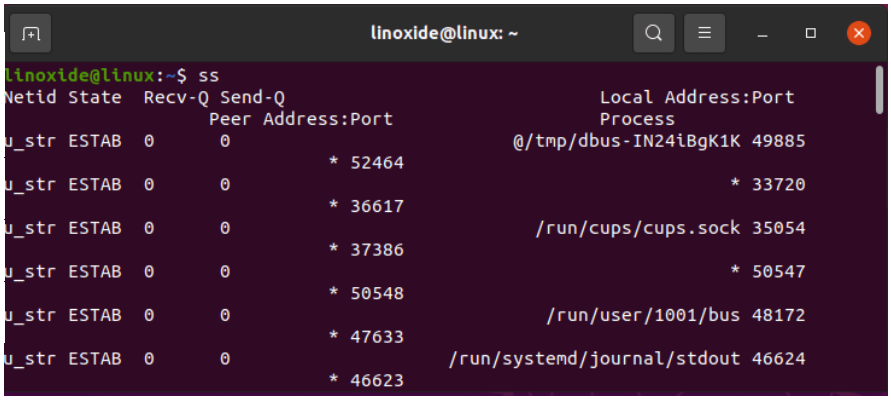
The ss tool comes with the iproute2 package. It can display stats for PACKET, TCP, UDP, DCCP, RAW, and Unix domain sockets.

In this tutorial, we learn **ss command** in Linux with useful examples.

1. List network connection

The ss command without any options list all open non-listening sockets (e.g. TCP/UNIX/UDP) that have established connection.

```
$ ss
```



1. Netid: It displays the types of sockets.
2. State: It displays the state of a socket if it is Established (ESTAB), Unconnected (UNCONN), or, Listening (LISTEN).
3. Recv-Q: It displays the number of received packets in the queue.
4. Send-Q: It displays the number of sent packets in the queue.
5. Local address:port : It displays the address of local machine and port.
6. Peer address:port : It displays the address of remote machine and port.

You can obtain more detailed information by using the ss command in conjunction with options. You can also select multiple options at the same time.

ss command syntax:

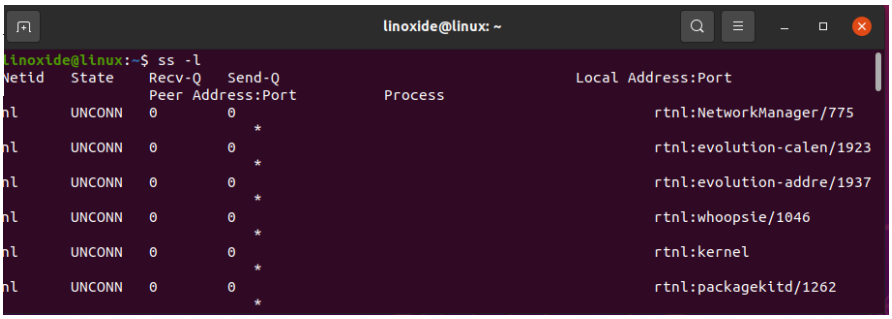
```
$ ss [Option]
```

```
$ ss [Option1] [Option2] [Option3]
```

2. List listening sockets

To display the list of listening sockets using `-l` or `--listen` option.

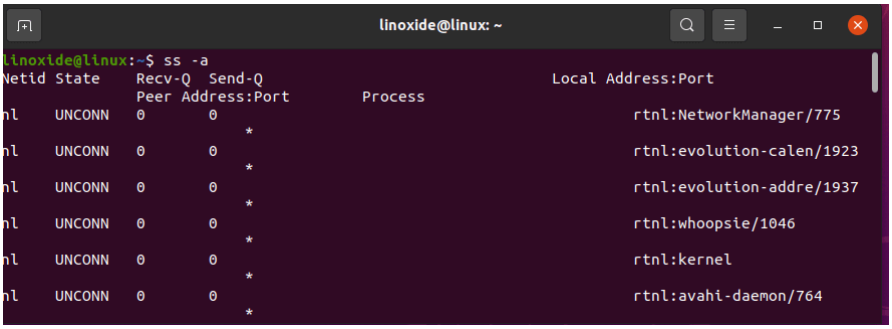
```
$ ss -l
```



3. List all the sockets

You can list all listening and non-listening network connections using the `-a` or `-all` option.

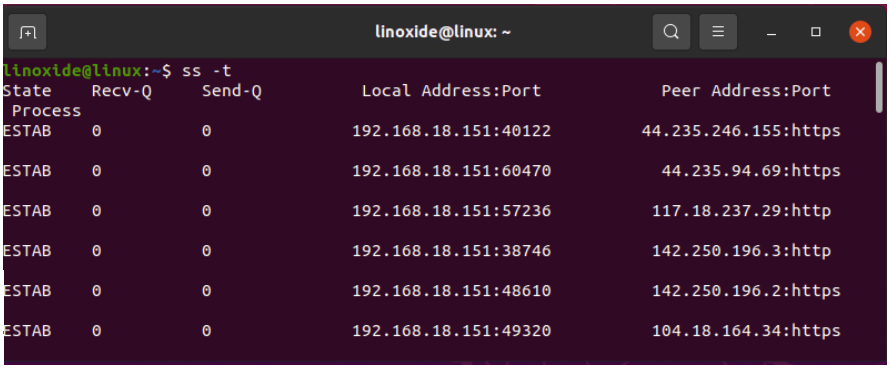
```
$ ss -a
```



4. List TCP Connection

To display the TCP socket connection, use the `-t` or `--tcp` option.

```
$ ss -t
```



To display the list of **all the TCP connections**, you can use the `-a` and `-t` options. This includes all states of the socket.

```
$ ss -at
```

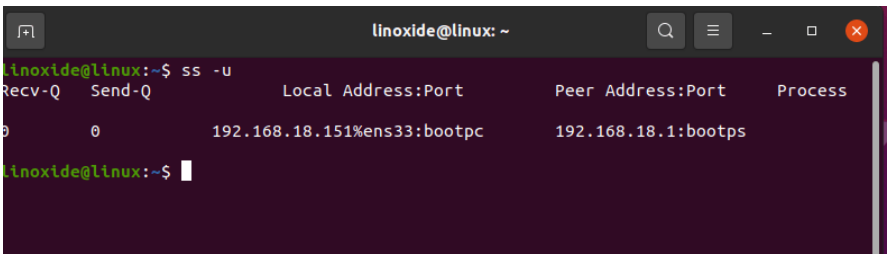
To display the TCP connection for all the listening states, combine `-l` and `-t` options.

```
$ ss -alt
```

5. List UDP Connection

To display the UDP socket connection, use `-u` or `--udp` option.

```
$ ss -u
```



To display the **list of all the UDP connections**, use `-a` and `-u` options. This includes all states of the socket.

```
$ ss -au
```

You can combine `-l` and `-u` to display the UDP connection for **all the listening states**.

```
$ ss -lu
```

6. List Unix Sockets

To display all the Unix sockets, you can use the ss command along with `-f unix` or `-x`.

```
$ ss -f unix
```

7. List Raw Sockets

To display all the Raw sockets, you can use `-w` or --raw option.

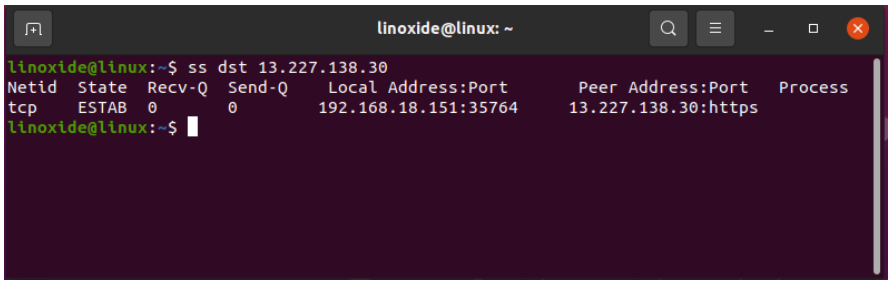
```
$ ss -w
```

8. List connection of an IP address

We can use ss command to display the list connection of a specific destination or source IP address.

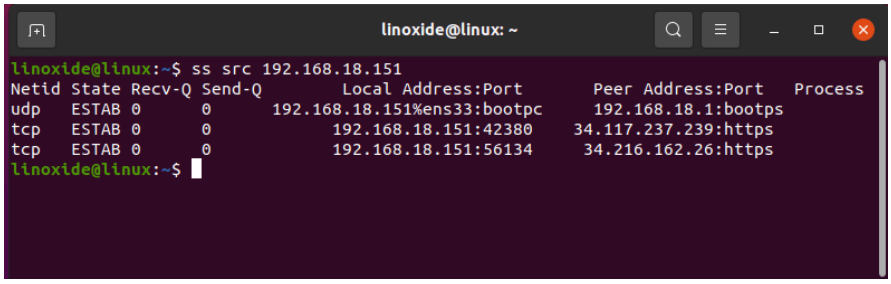
For example to list connection of destination IP address:

```
$ ss dst 13.227.138.30
```



For example to list connection of source IP address:

```
$ ss src 192.168.18.151
```

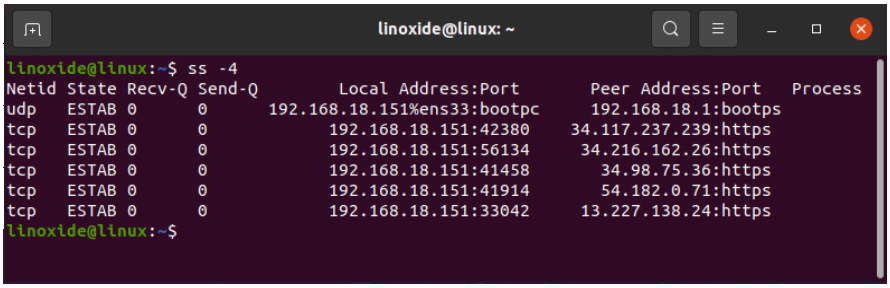


9. List IPv4 and IPv6 Socket Connection

If you want to display the list of IPv4 socket connections use -4 option and -6 to display the list of IPv6 socket connections.

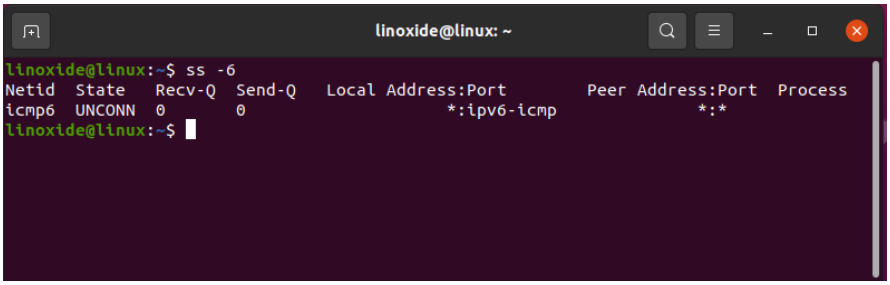
To display IPv4 socket connection list:

```
$ ss -4
```



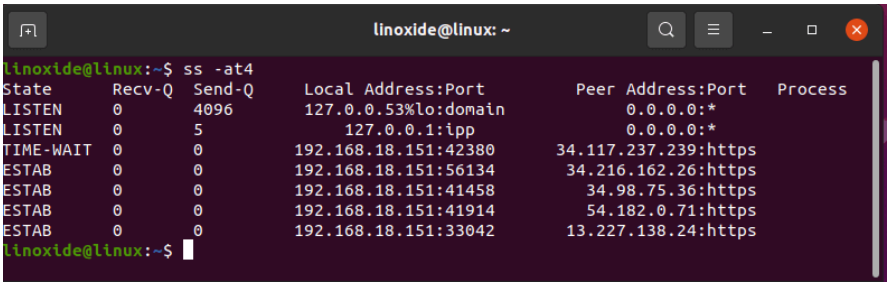
To display the IPv6 socket connection list:

```
$ ss -6
```



To list all the IPv4 TCP connections, you can use the following example.

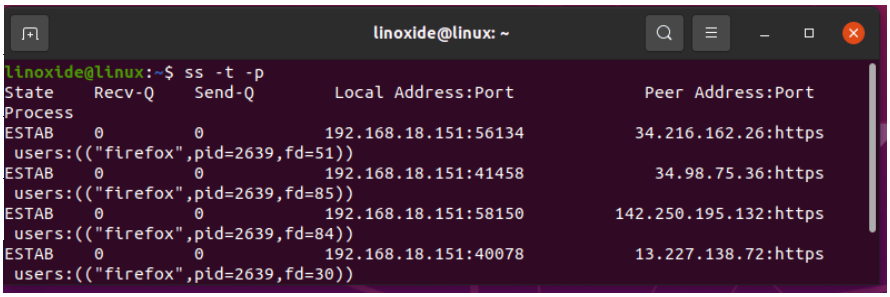
```
$ ss -at4
```



10. Identify processes

You can find the **processes of sockets** using the `-p` option along with the ss command. To identify the process, you will need sudo permission.

```
$ sudo ss -t -p
```



11. List Connection with no hostname resolution

To resolve the numeric address/ports use -r (resolve) option. Whereas the -n option does not try to resolve service names.

Here in the example, you can see the difference between the two:

```
$ ss -tn
 State  Recv-Q Send-Q  Local Address:Port Peer
 ESTAB  0      0       74.208.235.196:22  48.19
$ ss -tr
 State  Recv-Q Send-Q Local Address:Port
 ESTAB  0      64     li82-186.members.linode.c
$
```

12. Filter by Connection

Let's check few examples of how to apply filters to output specific information.

To filter TCP connection with state listening, type:

```
$ ss -t state listening
```

To display established ssh port connections:

```
$ ss -tr state established '( dport = :22 or sp
```

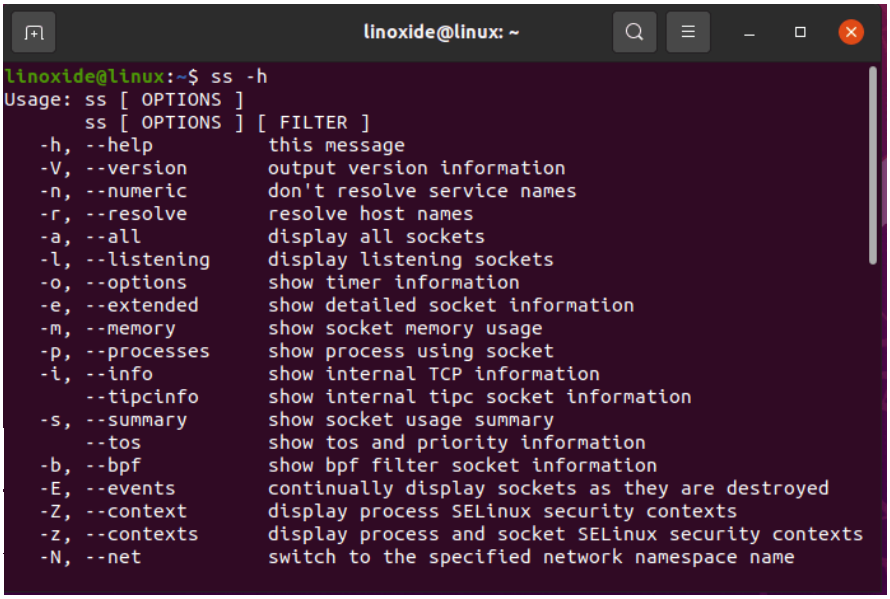
You can also the traditional way of grep command to filter. Here I am displaying all the TCP connections which are listening in the state:

```
$ ss -at | grep LISTEN
```

ss command options

The ss command provides various options to control the output to be displayed as per your requirement. You can use -h or --help along with the ss command to view the basic options available with the ss command-utility.

```
$ ss -h
```



ss vs netstat command

The ss tool is included under iproute2 package and its default in most Linux distributions. To have netstat you need to install net-tools, which is already deprecated. The ss command is much faster as it fetches directly from the kernel. The ss is not a complete replacement of netstat, some of the netstat command is replaced by ip command.

Conclusion

In this tutorial, we learned about the ss command with some useful examples. You can refer ss command man page for more information.

Leave a Comment