

## ss

- 소켓의 네트워크 통계를 나타내는 유틸리티
- Ubuntu에는 기본적으로 ss가 default로 존재

## option

- ss -t  
>> TCP Port만 확인
- ss -u  
>> UDP Port만 확인
- ss -a  
>> 모든 Port 확인
- ss -i  
>> RTO, RTT 등 소켓의 네트워크 정보를 자세하게 표시
- ss [Filter]  
Ex) ss src :443  
>> 출발지 포트가 443인 소켓만 조사

Ex) ss src 203.250.xx.xx

>> 출발지 IP가 203.250.xx.xx인 소켓만 조사

Ex) ss src 203.250.xx.xx:443

>> 출발지 IP가 203.250.xx.xx, 포트가 443인 소켓만 조사

Ex) ss src 203.250.xx.xx:443 dst 203.250.xx.xx:443

>> 출발지 IP가 203.250.XX.XX, 포트가 443 / 목적지 IP가 203.250.xx.xx, 포트가 443인 소켓만 조사

Ex) ss -t src :22

>> TCP이며 출발지 포트가 22(ssh)인 소켓만 조사

Ex) ss -t -i src :22

>> TCP이며 출발지 포트가 22(ssh)인 소켓만 조사

>> 또한, RTO RTT 등 소켓에 대한 네트워크 정보를 자세하게 표시

```
wlgns12www@INS_Lab:~$ ss -t -i src :22
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
ESTAB      0            64          220.69.209.200:ssh      36.38.62.102:62343
           cubic wscale:8,7 rto:256 rtt:53.947/11.759 ato:64 mss:1460 pmtu:1500 rcvmss:1168 adv
mss:1460 cwnd:10 bytes_acked:777233 bytes_received:52940 segs_out:1400 segs_in:1732 data_segs
_out:1396 data_segs_in:773 send 2.2Mbps lastsnd:4 lastrcv:4 lastack:4 pacing_rate 4.3Mbps del
ivery_rate 2.6Mbps app_limited busy:42324ms unacked:1 rcv_space:14600 rcv_ssthresh:64076 minr
tt:14.936
```

```
>> wscale:<snd_wscale>:<rcv_wscale> { window scale 옵션 사용 시 송수신 scale factor }
>> rto:<icsk_rto> { tcp 재전송 시간 초과 값, 단위는 밀리초 }
>> rtt:<rtt>/<rttvar> { <rtt>는 평균 rtt 시간, <rttvar>는 <rtt>의 평균 편차, 단위는 밀리초 }
>> ato:<ato> { ack timeout, 지연 ack 모드에서 사용, 단위는 밀리초 }
>> mss:<mss> { max segment size }
>> pmtu:<pmtu> { path MTU value }
>> rcvmss
>> advmss
>> cwnd:<cwnd> { congestion window size }
>> bytes_acked:<bytes_acked> { bytes acked }
>> bytes_received:<bytes_received> { bytes received }
>> segs_out:<segs_out> { segments sent out }
>> segs_in:<segs_in> { segments received }
>> data_segs_out
>> data_segs_in
>> send <send(bps)> { egress bps }
>> lastsnd:<lastsnd> { 마지막 패킷을 보낸 지 얼마나 되었는지, 단위는 밀리초 }
>> lastrcv:<lastrcv> { 마지막 패킷이 수신된 후 얼마나 되었는지, 단위는 밀리초 }
>> lastack:<lastack> { 마지막 ACK를 받은 지 얼마나 되었는지, 단위는 밀리초 }
>> pacing_rate <pacing_rate>bps/<max_pacing_rate>bps
>> rcv_space:<rcv_space>
>> rcv_ssthread
>> minrtt:<minrtt> { 최소 rtt }
```

## Ref

- <https://web-dev.tistory.com/1003>
- <https://man7.org/linux/man-pages/man8/ss.8.html>
- <https://unix.stackexchange.com/questions/542712/detailed-output-of-ss-command>