Tristan Breen

Moubayed

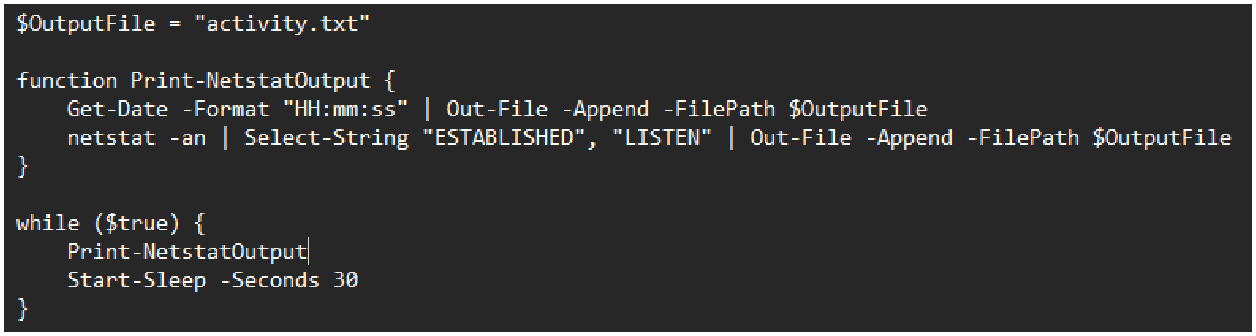
SER335

30 January 2024

Assignment 1 Part 2 Networking basics

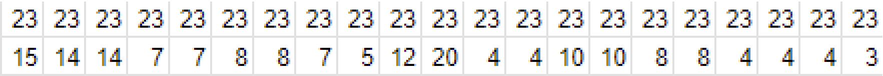
1.1. Understanding TCP network sockets

1.



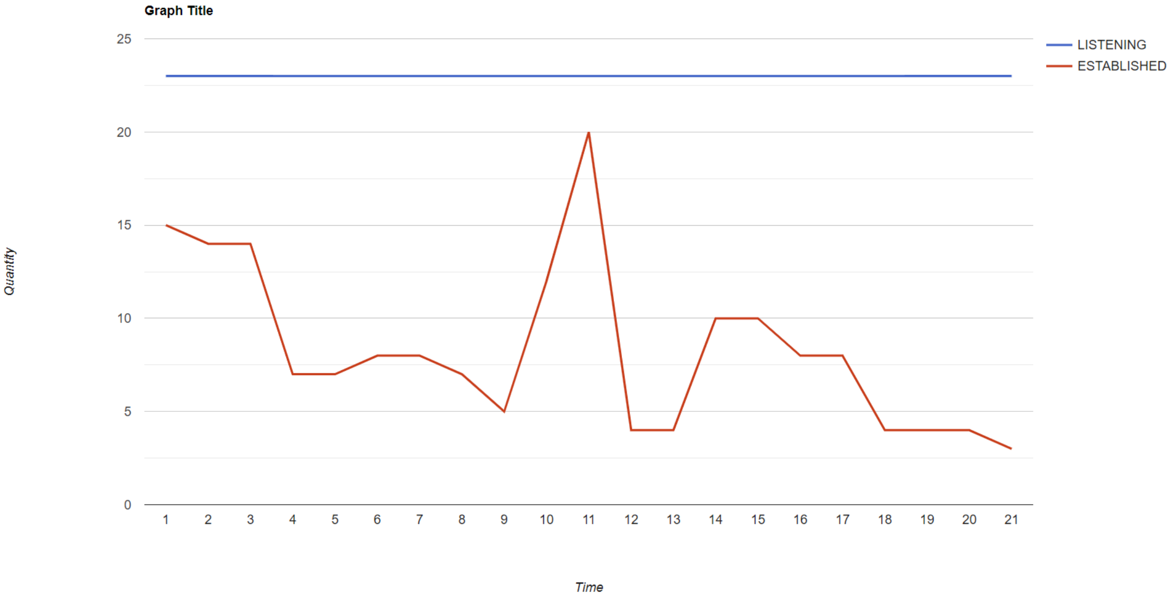
```
$OutputFile = "activity.txt"

function Print-NetstatOutput {
    Get-Date -Format "HH:mm:ss" | Out-File -Append -FilePath $OutputFile
    netstat -an | Select-String "ESTABLISHED", "LISTEN" | Out-File -Append -FilePath $OutputFile
}

while ($true) {
    Print-NetstatOutput
    Start-Sleep -Seconds 30
}
```

2.



23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23
15	14	14	7	7	8	8	7	5	12	20	4	4	10	10	8	8	4	4	4	3

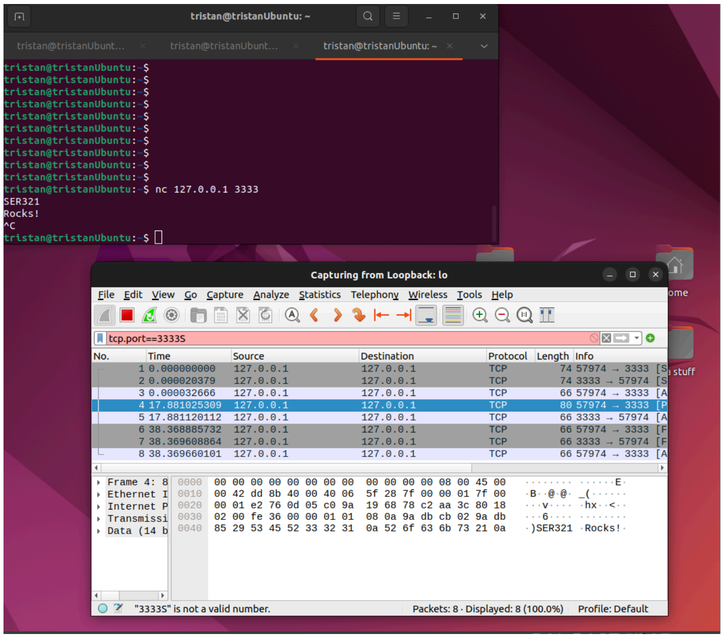
3.



1.2. Sniffing TCP/UDP traffic

Step 1 (TCP)

1.



```
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$
tristan@tristanUbuntu:~$ nc 127.0.0.1 3333
SER321
Rocks!
^C
tristan@tristanUbuntu:~$
```

Capturing from Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ome
stuff

tcp.port==3333S

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	57974 → 3333 [S
2	0.000020379	127.0.0.1	127.0.0.1	TCP	74	3333 → 57974 [S
3	0.000032666	127.0.0.1	127.0.0.1	TCP	66	57974 → 3333 [A
4	17.881025309	127.0.0.1	127.0.0.1	TCP	80	57974 → 3333 [P
5	17.881120112	127.0.0.1	127.0.0.1	TCP	66	3333 → 57974 [A
6	38.368885732	127.0.0.1	127.0.0.1	TCP	66	57974 → 3333 [F
7	38.369608864	127.0.0.1	127.0.0.1	TCP	66	3333 → 57974 [F
8	38.369660101	127.0.0.1	127.0.0.1	TCP	66	57974 → 3333 [A

```
▸ Frame 4: 8    0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
▸ Ethernet I    0010  00 42 dd 8b 40 00 40 06  5f 28 7f 00 00 01 7f 00   ·B··@·@· _(······
▸ Internet P    0020  00 01 e2 76 0d 05 c0 9a  19 68 78 c2 aa 3c 80 18   ···v···· ·hx··<··
▸ Transmissi    0030  02 00 fe 36 00 00 01 01  08 0a 9a db cb 02 9a db   ···6···· ········
▸ Data (14 b    0040  85 29 53 45 52 33 32 31  0a 52 6f 63 6b 73 21 0a   ·)SER321 ·Rocks!·
```

"3333S" is not a valid number.   Packets: 8 · Displayed: 8 (100.0%)   Profile: Default

2.

a. The first command used netcat to set up a server on port 3333 that listens for others on the same port. The second command connects to the server running on the ip 127.0.0.1 (local host) on the port 3333.

b. There are 3 initial frames to start the connection and depending on if you send Ser321 Rocks! Separately or together determines the number of frames. I pasted them in together so I received 2 frames (4 and 5 shown in the image). If you sent them separately then it would be 4 frames. Then signing off is another 3 frames.

c. 8 packets

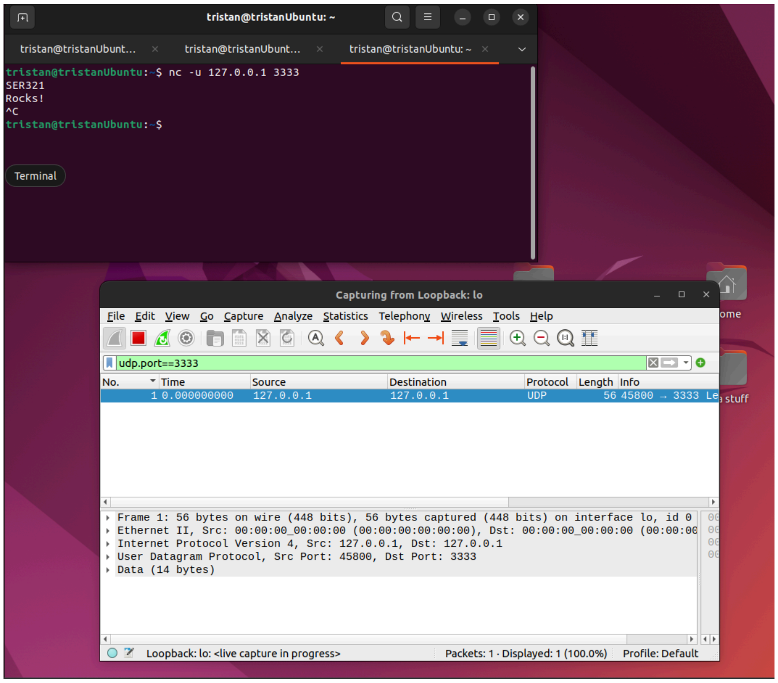
d. 2 Packets

e. 98 + 66 =164 bytes

f. 1,086 bytes

g. 1086 - 164 = 922 bytes

Step 2 (UDP)

1. 

```
tristan@tristanUbuntu:~$ nc -u 127.0.0.1 3333
SER321
Rocks!
^C
tristan@tristanUbuntu:~$
```

Capturing from Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ome
stuff

udp.port==3333

```
No.   Time          Source      Destination  Protocol  Length  Info
    1 0.000000000   127.0.0.1   127.0.0.1    UDP           56 45800 → 3333 Le
```

```
Frame 1: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
User Datagram Protocol, Src Port: 45800, Dst Port: 3333
Data (14 bytes)
```

Loopback: lo: <live capture in progress>   Packets: 1 · Displayed: 1 (100.0%)   Profile: Default

2.

   a. The first command used netcat to set up a udp server on port 3333 that listens for others on the same port. The second command connects to the server running on the ip 127.0.0.1 (local host) on the port 3333.

   b. 1 frame

   c. 1 packet

   d. 1 packet

   e. 49 bytes

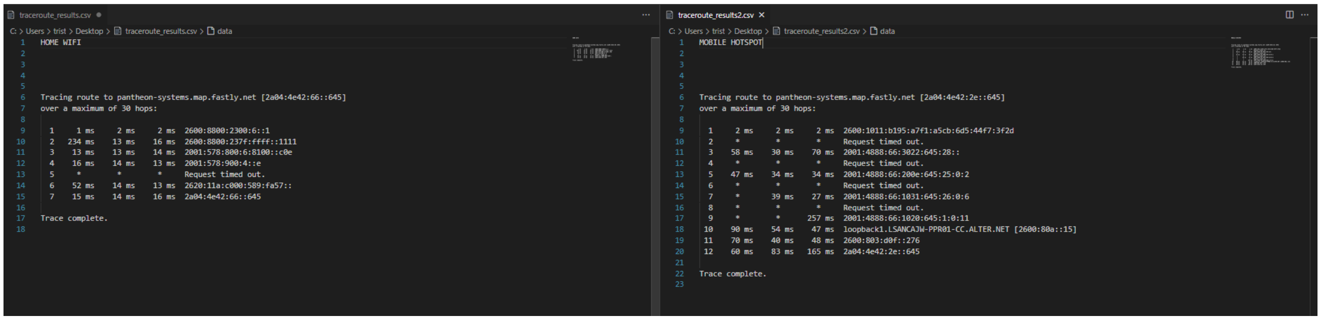
   f. 49 bytes

   g. 0 bytes

   h. UDP does not check for anyone listening so it can just send out the information unlike TCP which needs to verify the other end is listening so it can send out its data.

1.3. Internet Protocol (IP) Routing

1.



traceroute_results.csv
C: > Users > trist > Desktop > traceroute_results.csv > data

```
HOME WIFI




Tracing route to pantheon-systems.map.fastly.net [2a04:4e42:66::645]
over a maximum of 30 hops:

  1     1 ms     2 ms     2 ms  2600:8800:2300:6::1
  2   234 ms    13 ms    16 ms  2600:8800:237f:ffff::1111
  3    13 ms    13 ms    14 ms  2001:578:800:6:8100::c0e
  4    16 ms    14 ms    13 ms  2001:578:900:4::e
  5     *        *        *     Request timed out.
  6    52 ms    14 ms    13 ms  2620:11a:c000:589:fa57::
  7    15 ms    14 ms    16 ms  2a04:4e42:66::645

Trace complete.
```

traceroute_results2.csv
C: > Users > trist > Desktop > traceroute_results2.csv > data

```
MOBILE HOTSPOT




Tracing route to pantheon-systems.map.fastly.net [2a04:4e42:2e::645]
over a maximum of 30 hops:

  1     2 ms     2 ms     2 ms  2600:1011:b195:a7f1:a5cb:6d5:44f7:3f2d
  2     *        *        *     Request timed out.
  3    58 ms    30 ms    70 ms  2001:4888:66:3022:645:28::
  4     *        *        *     Request timed out.
  5    47 ms    34 ms    34 ms  2001:4888:66:200e:645:25:0:2
  6     *        *        *     Request timed out.
  7     *       39 ms    27 ms  2001:4888:66:1031:645:26:0:6
  8     *        *        *     Request timed out.
  9     *        *       257 ms  2001:4888:66:1020:645:1:0:11
 10    90 ms    54 ms    47 ms  loopback1.LSANCAJW-PPR01-CC.ALTER.NET [2600:80a::15]
 11    70 ms    40 ms    48 ms  2600:803:d0f::276
 12    60 ms    83 ms   165 ms  2a04:4e42:2e::645

Trace complete.
```

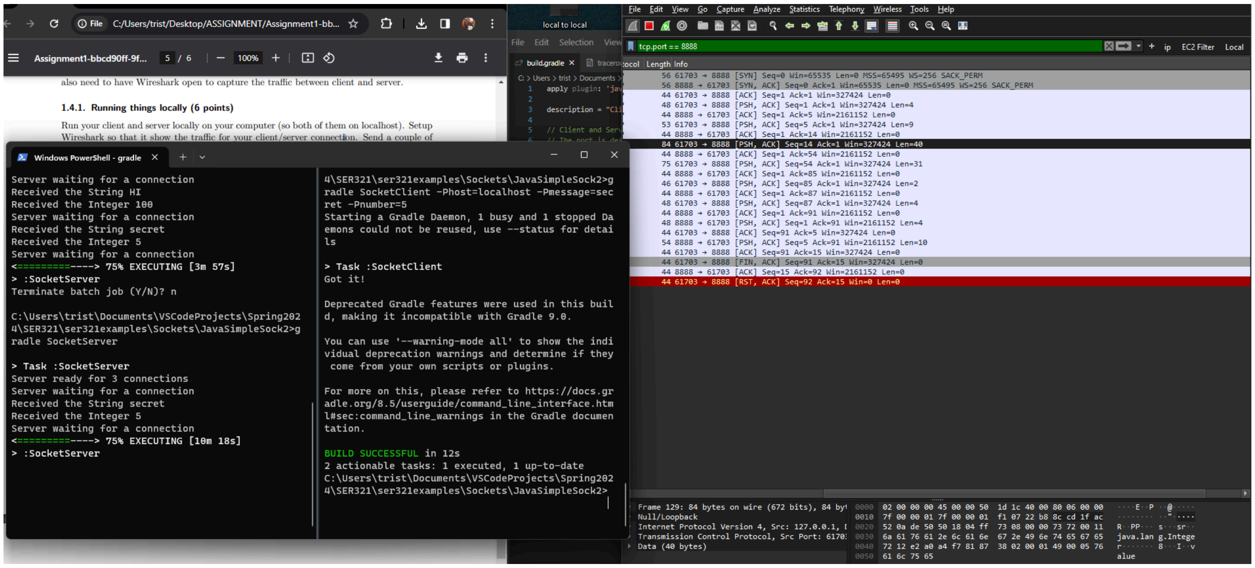
2.

a. Over my home wifi was much faster

b. Over my home wifi I had fewer hops

1.4.1. Running things locally

1.



File
C:/Users/trist/Desktop/ASSIGNMENT/Assignment1-bb...
Assignment1-bbcd90ff-9f...
5 / 6
100%
local to local
File Edit Selection View
build.gradle
tracerou
C: > Users > trist > Documents >
Windows PowerShell - gradle
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.port == 8888
ip
EC2 Filter
Local
tocol
Length
Info

also need to have Wireshark open to capture the traffic between client and server.

1.4.1. Running things locally (6 points)

Run your client and server locally on your computer (so both of them on localhost). Setup Wireshark so that it show the traffic for your client/server connection. Send a couple of

```
apply plugin: 'jav

description = "Cli

// Client and Serv
// The port is def
```

```
Server waiting for a connection
Received the String HI
Received the Integer 100
Server waiting for a connection
Received the String secret
Received the Integer 5
Server waiting for a connection
<=========----> 75% EXECUTING [3m 57s]
> :SocketServer
Terminate batch job (Y/N)? n

C:\Users\trist\Documents\VSCodeProjects\Spring202
4\SER321\ser321examples\Sockets\JavaSimpleSock2>g
radle SocketServer

> Task :SocketServer
Server ready for 3 connections
Server waiting for a connection
Received the String secret
Received the Integer 5
Server waiting for a connection
<=========----> 75% EXECUTING [10m 18s]
> :SocketServer
```

```
4\SER321\ser321examples\Sockets\JavaSimpleSock2>g
radle SocketClient -Phost=localhost -Pmessage=sec
ret -Pnumber=5
Starting a Gradle Daemon, 1 busy and 1 stopped Da
emons could not be reused, use --status for detai
ls

> Task :SocketClient
Got it!

Deprecated Gradle features were used in this buil
d, making it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the indi
vidual deprecation warnings and determine if they
 come from your own scripts or plugins.

For more on this, please refer to https://docs.gr
adle.org/8.5/userguide/command_line_interface.htm
l#sec:command_line_warnings in the Gradle documen
tation.

BUILD SUCCESSFUL in 12s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\trist\Documents\VSCodeProjects\Spring202
4\SER321\ser321examples\Sockets\JavaSimpleSock2>
```

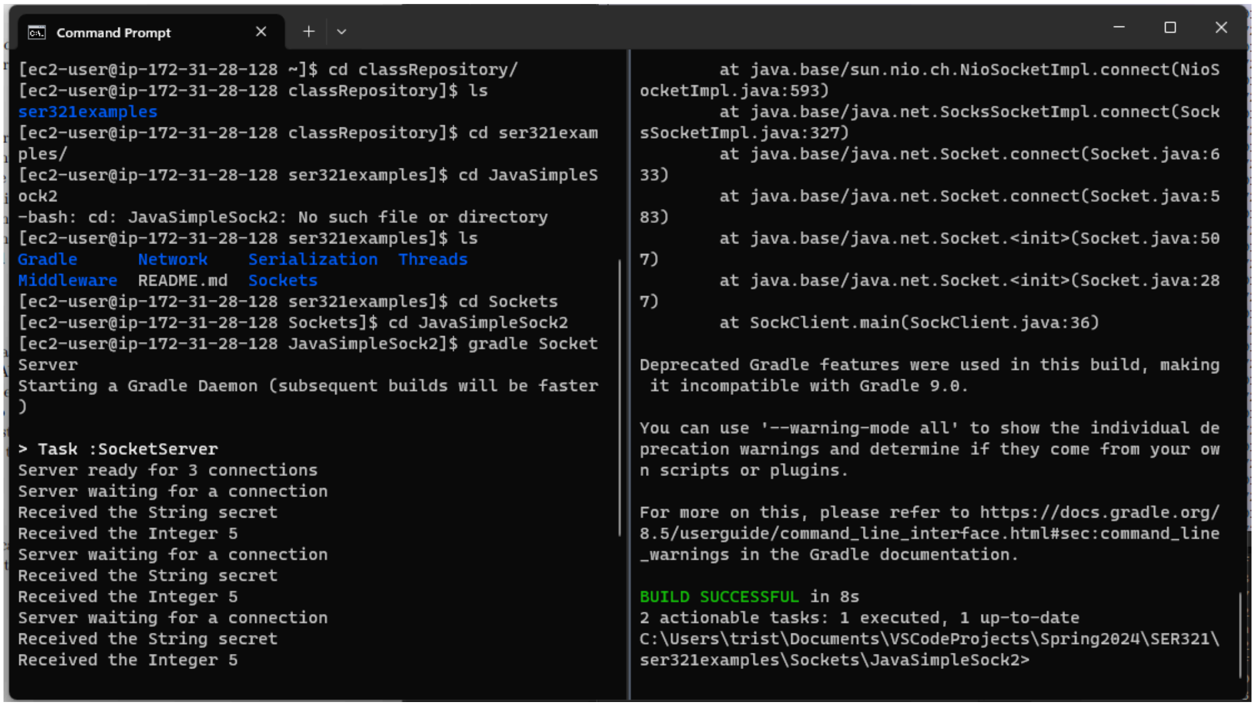
```
56 61703 → 8888 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
56 8888 → 61703 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
44 61703 → 8888 [ACK] Seq=1 Ack=1 Win=327424 Len=0
48 61703 → 8888 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=4
44 8888 → 61703 [ACK] Seq=1 Ack=5 Win=2161152 Len=0
53 61703 → 8888 [PSH, ACK] Seq=5 Ack=1 Win=327424 Len=9
44 8888 → 61703 [ACK] Seq=1 Ack=14 Win=2161152 Len=0
84 61703 → 8888 [PSH, ACK] Seq=14 Ack=1 Win=327424 Len=40
44 8888 → 61703 [ACK] Seq=1 Ack=54 Win=2161152 Len=0
75 61703 → 8888 [PSH, ACK] Seq=54 Ack=1 Win=327424 Len=31
44 8888 → 61703 [ACK] Seq=1 Ack=85 Win=2161152 Len=0
46 61703 → 8888 [PSH, ACK] Seq=85 Ack=1 Win=327424 Len=2
44 8888 → 61703 [ACK] Seq=1 Ack=87 Win=2161152 Len=0
48 61703 → 8888 [PSH, ACK] Seq=87 Ack=1 Win=327424 Len=4
44 8888 → 61703 [ACK] Seq=1 Ack=91 Win=2161152 Len=0
48 8888 → 61703 [PSH, ACK] Seq=1 Ack=91 Win=2161152 Len=4
44 61703 → 8888 [ACK] Seq=91 Ack=5 Win=327424 Len=0
54 8888 → 61703 [PSH, ACK] Seq=5 Ack=91 Win=2161152 Len=10
44 61703 → 8888 [ACK] Seq=91 Ack=15 Win=327424 Len=0
44 61703 → 8888 [FIN, ACK] Seq=91 Ack=15 Win=327424 Len=0
44 8888 → 61703 [ACK] Seq=15 Ack=92 Win=2161152 Len=0
44 61703 → 8888 [RST, ACK] Seq=92 Ack=15 Win=0 Len=0
```

```
Frame 129: 84 bytes on wire (672 bits), 84 byt
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, D
Transmission Control Protocol, Src Port: 6170
Data (40 bytes)
```

```
0000  02 00 00 00 45 00 00 50  1d 1c 40 00 80 06 00 00   ····E··P ··@·····
0010  7f 00 00 01 7f 00 00 01  f1 07 22 b8 8c cd 1f ac   ········ ··"·····
0020  52 0a de 50 50 18 04 ff  73 08 00 00 73 72 00 11   R··PP··· s···sr··
0030  6a 61 76 61 2e 6c 61 6e  67 2e 49 6e 74 65 67 65   java.lan g.Intege
0040  72 12 e2 a0 a4 f7 81 87  38 02 00 01 49 00 05 76   r······· 8···I··v
0050  61 6c 75 65                                        alue
```

2. https://drive.google.com/file/d/1Cla-kgUt_jjwe7t2sMyCW1x0ugE1mJBd/view?usp=sharing

1.4.2. Server on AWS

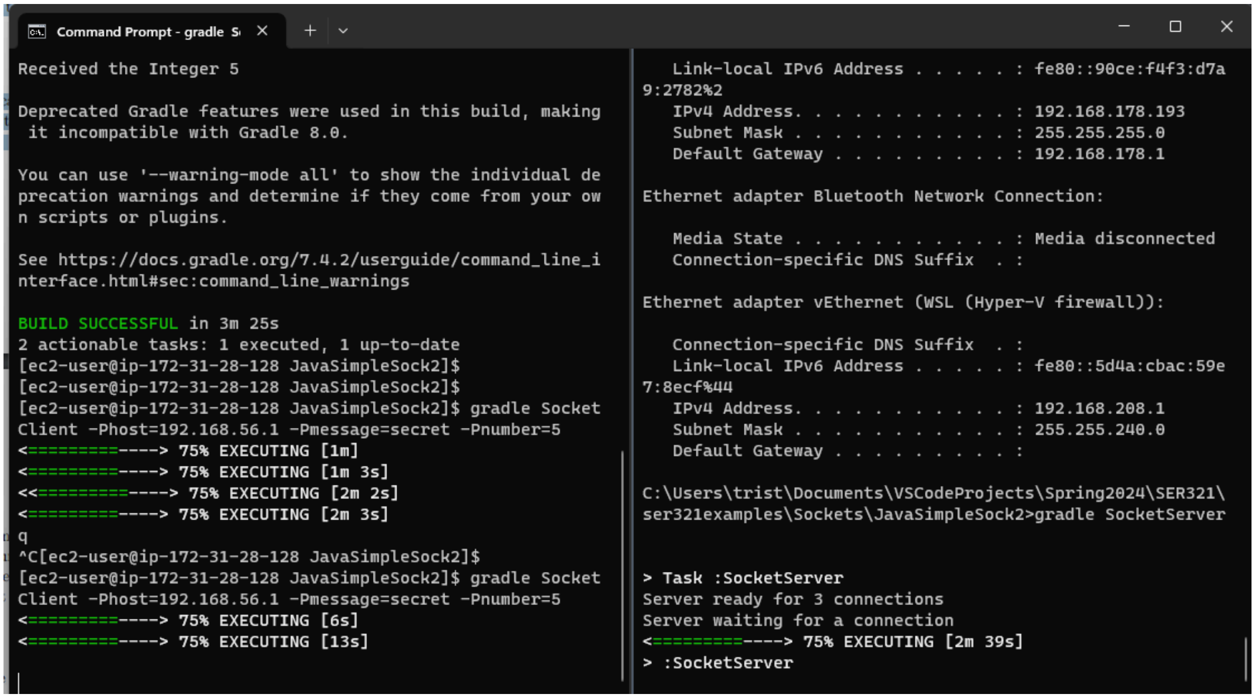
1. 

```
[ec2-user@ip-172-31-28-128 ~]$ cd classRepository/
[ec2-user@ip-172-31-28-128 classRepository]$ ls
ser321examples
[ec2-user@ip-172-31-28-128 classRepository]$ cd ser321exam
ples/
[ec2-user@ip-172-31-28-128 ser321examples]$ cd JavaSimpleS
ock2
-bash: cd: JavaSimpleSock2: No such file or directory
[ec2-user@ip-172-31-28-128 ser321examples]$ ls
Gradle      Network    Serialization  Threads
Middleware  README.md  Sockets
[ec2-user@ip-172-31-28-128 ser321examples]$ cd Sockets
[ec2-user@ip-172-31-28-128 Sockets]$ cd JavaSimpleSock2
[ec2-user@ip-172-31-28-128 JavaSimpleSock2]$ gradle Socket
Server
Starting a Gradle Daemon (subsequent builds will be faster
)

> Task :SocketServer
Server ready for 3 connections
Server waiting for a connection
Received the String secret
Received the Integer 5
Server waiting for a connection
Received the String secret
Received the Integer 5
Server waiting for a connection
Received the String secret
Received the Integer 5
```

```
        at java.base/sun.nio.ch.NioSocketImpl.connect(NioS
ocketImpl.java:593)
        at java.base/java.net.SocksSocketImpl.connect(Sock
sSocketImpl.java:327)
        at java.base/java.net.Socket.connect(Socket.java:6
33)
        at java.base/java.net.Socket.connect(Socket.java:5
83)
        at java.base/java.net.Socket.<init>(Socket.java:50
7)
        at java.base/java.net.Socket.<init>(Socket.java:28
7)
        at SockClient.main(SockClient.java:36)

Deprecated Gradle features were used in this build, making
 it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the individual de
precation warnings and determine if they come from your ow
n scripts or plugins.

For more on this, please refer to https://docs.gradle.org/
8.5/userguide/command_line_interface.html#sec:command_line
_warnings in the Gradle documentation.

BUILD SUCCESSFUL in 8s
2 actionable tasks: 1 executed, 1 up-to-date
C:\Users\trist\Documents\VSCodeProjects\Spring2024\SER321\
ser321examples\Sockets\JavaSimpleSock2>
```

2. Above is what I ran in the terminal. The server did not need to be changed but the client had to call the host via the EC2 instance's ip and can not just rely on it being local.In wireshark I had to change the port to filter for the EC2 instance as well.

1.4.3. Client on AWS

1. 

```
Received the Integer 5

Deprecated Gradle features were used in this build, making
 it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual de
precation warnings and determine if they come from your ow
n scripts or plugins.

See https://docs.gradle.org/7.4.2/userguide/command_line_i
nterface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 3m 25s
2 actionable tasks: 1 executed, 1 up-to-date
[ec2-user@ip-172-31-28-128 JavaSimpleSock2]$
[ec2-user@ip-172-31-28-128 JavaSimpleSock2]$
[ec2-user@ip-172-31-28-128 JavaSimpleSock2]$ gradle Socket
Client -Phost=192.168.56.1 -Pmessage=secret -Pnumber=5
<=========----> 75% EXECUTING [1m]
<=========----> 75% EXECUTING [1m 3s]
<<=========----> 75% EXECUTING [2m 2s]
<=========----> 75% EXECUTING [2m 3s]

q
^C[ec2-user@ip-172-31-28-128 JavaSimpleSock2]$
[ec2-user@ip-172-31-28-128 JavaSimpleSock2]$ gradle Socket
Client -Phost=192.168.56.1 -Pmessage=secret -Pnumber=5
<=========----> 75% EXECUTING [6s]
<=========----> 75% EXECUTING [13s]

|
```

```
   Link-local IPv6 Address . . . . . : fe80::90ce:f4f3:d7a
9:2782%2
   IPv4 Address. . . . . . . . . . . : 192.168.178.193
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.178.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter vEthernet (WSL (Hyper-V firewall)):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::5d4a:cbac:59e
7:8ecf%44
   IPv4 Address. . . . . . . . . . . : 192.168.208.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :

C:\Users\trist\Documents\VSCodeProjects\Spring2024\SER321\
ser321examples\Sockets\JavaSimpleSock2>gradle SocketServer


> Task :SocketServer
Server ready for 3 connections
Server waiting for a connection
<=========----> 75% EXECUTING [2m 39s]
> :SocketServer
```

2. While it should work theoretically…it did not in my practice even assuring I put in my Ip address instead of the Ec2 instances. It might end up working but I had been waiting for upwards of 2 minutes with no results. Amazon's servers might have restrictions on out of bounds traffic that I am unaware of.

1.4.4. Client on AWS 2

1. I think if we had a configuration where I could also give the server my private ip address in my household then I could get the messages. Currently the information returned from AWS is going to my router and does not know where to go after that but that would not be the case if I had also provided my private Ip so my router knows who wants the information. That would also be true if I wanted to reach someone else in the outside world…I need their public and private Ip address.

TO GRADER…I DON'T KNOW WHAT YOU WOULD LIKE ME TO PROVIDE FOR THIS BUT IF I YOU NEED ANYTHING ELSE FOR THIS ASSIGNMENT PLEASE LET ME KNOW.

Gradle/Java directory from the ?? section.