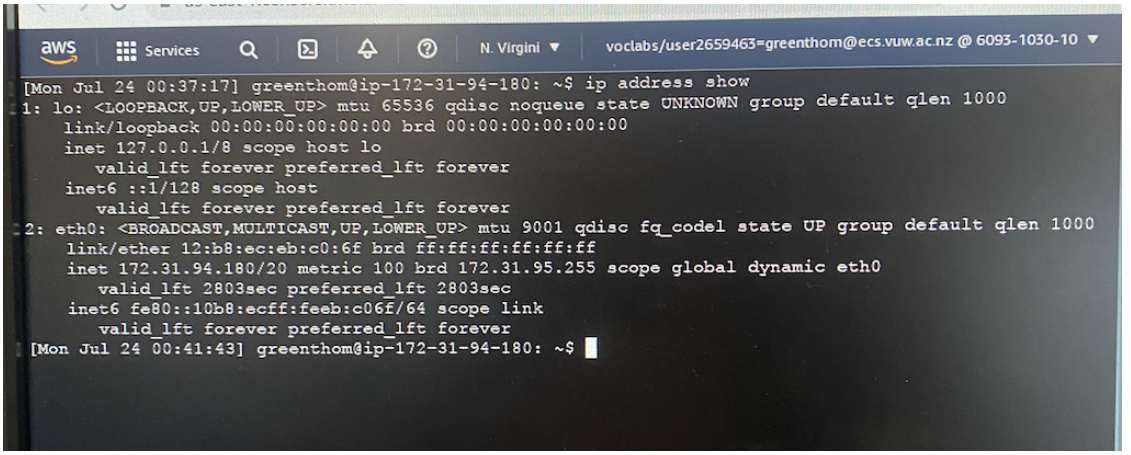
Project 1

NWEN PROJECT 1

1. 

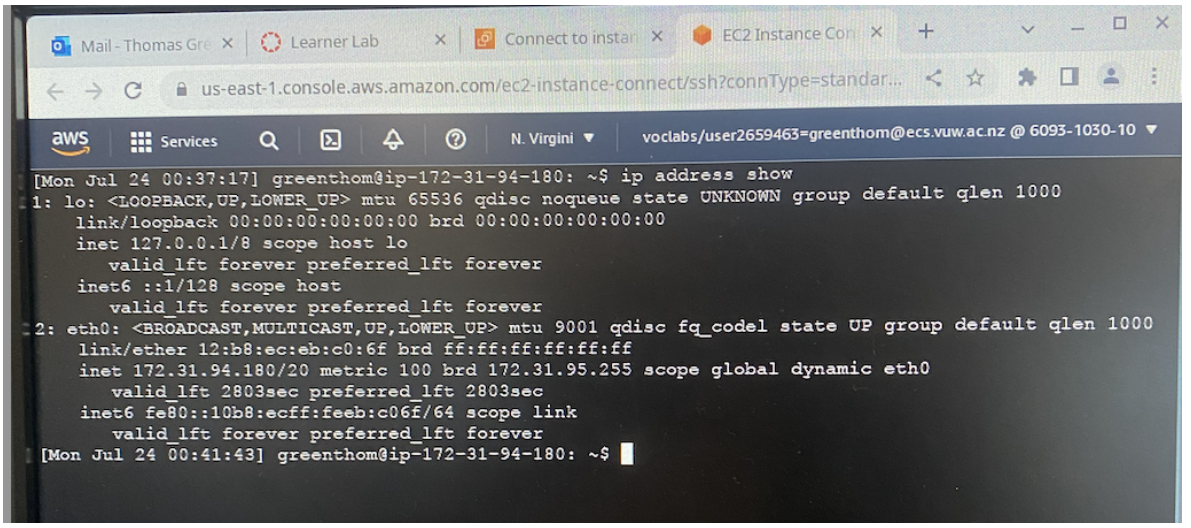
a. eth0

b. 12:b8:ec:eb:c0:6f

c. 00010010 : 10111000 : 11101100 : 11101011 : 11000000 : 01101111

d. 48 bits

e. ff:ff:ff:ff:ff:ff

2. 

a. Private is 172.31.94.180 and public is 52.207.249.77

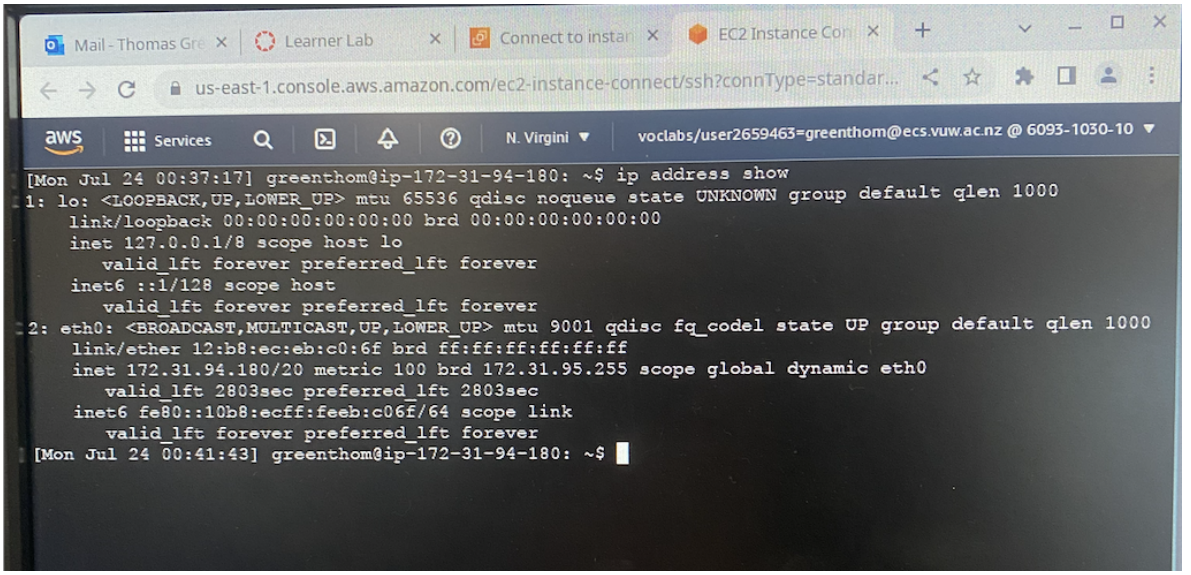
b. fe80::10b8:ecff:feeb:c06f/64

c. 32 bits

d. 00110100.11001111.11111001.01001101

e. 128 bits

3.

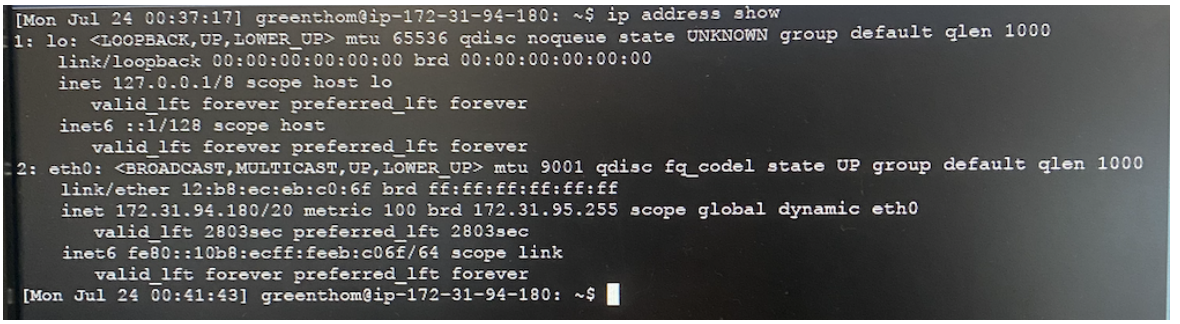


a. Network portion is 172.31 and host portion 94.180 - Subnet mask 172.31.255.255

b. Specified address range for private IPv4 address is as follows: Class A: 10.0.0.0 to 10.255.255.255. Class B: 172.16.0.0 to 172.31.255.255. Class C: 192.168.0.0 to 192.168.255.255

c. Number of distinct usable IPv4 addresses in this LAN would be 4096-2 = 4094 addresses that can be assigned to individual devices within the LAN

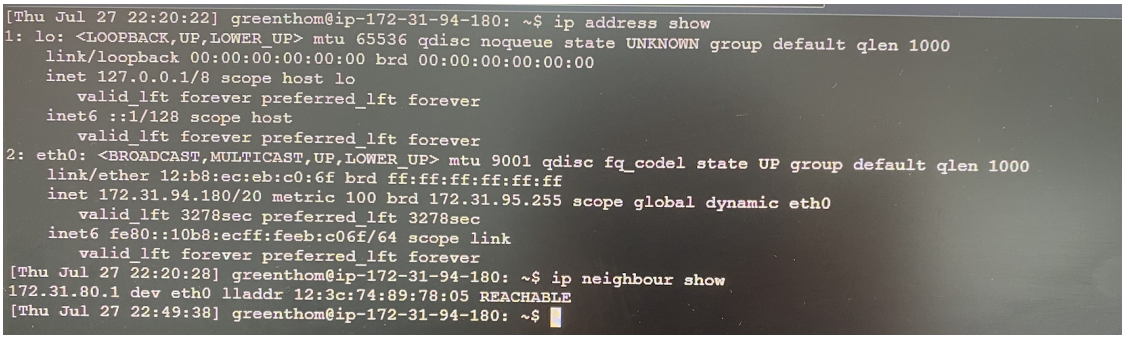
4.



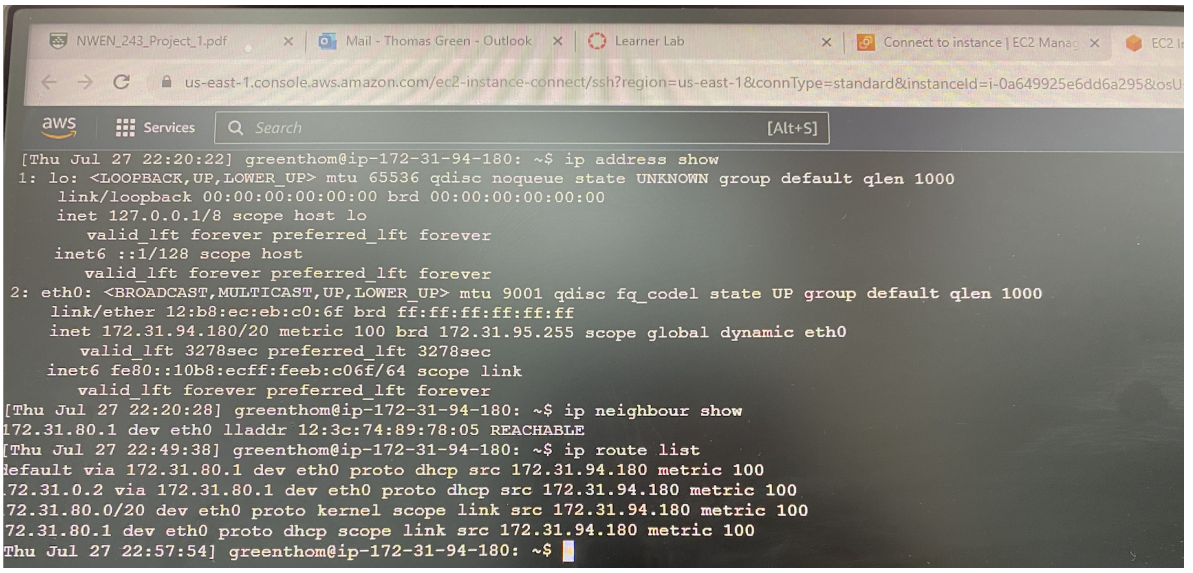
a. From the information provided earlier, the private IPv4 address is 172.31.94.180, and the network interface (eth0) is configured with a subnet mask of 255.255.240.0. In CIDR notation, subnet mask is represented by the number of leading bits set to 1. For the subnet mask 255.255.240.0, there are 20 leading bits set to 1, which corresponds to a /20 subnet. Therefore, the netmask for the private IPv4 address 172.31.94.180 is 255.255.240.0 (or /20 in CIDR notation)

b. Convert IPv4 address to binary: 172.31.94.180 -> 10101100.00011111.01011110.10110100. Convert the subnet to binary: 225.255.240.0 -> 11111111.11111111.11110000.00000000. Perform bitwise not

on subnet: 00000000.00000000.00001111.11111111. Perform bitwise OR operation: 10101100.00011111.01011111.11111111. Convert back to decimal: 172.31.95.255. Broadcast IPv4 address for given private IPv4 address with subnet mask is 172.31.95.225.
c. 1. Convert IPv4 address and netmask to binary format. 2. Perform bitwise NOT operation on netmask. Perform bitwise OR operation between the binary representation of the IPv4 address and the complemented netmask. Convert the result of OR operation back to decimal to get the broadcast IPv4 address

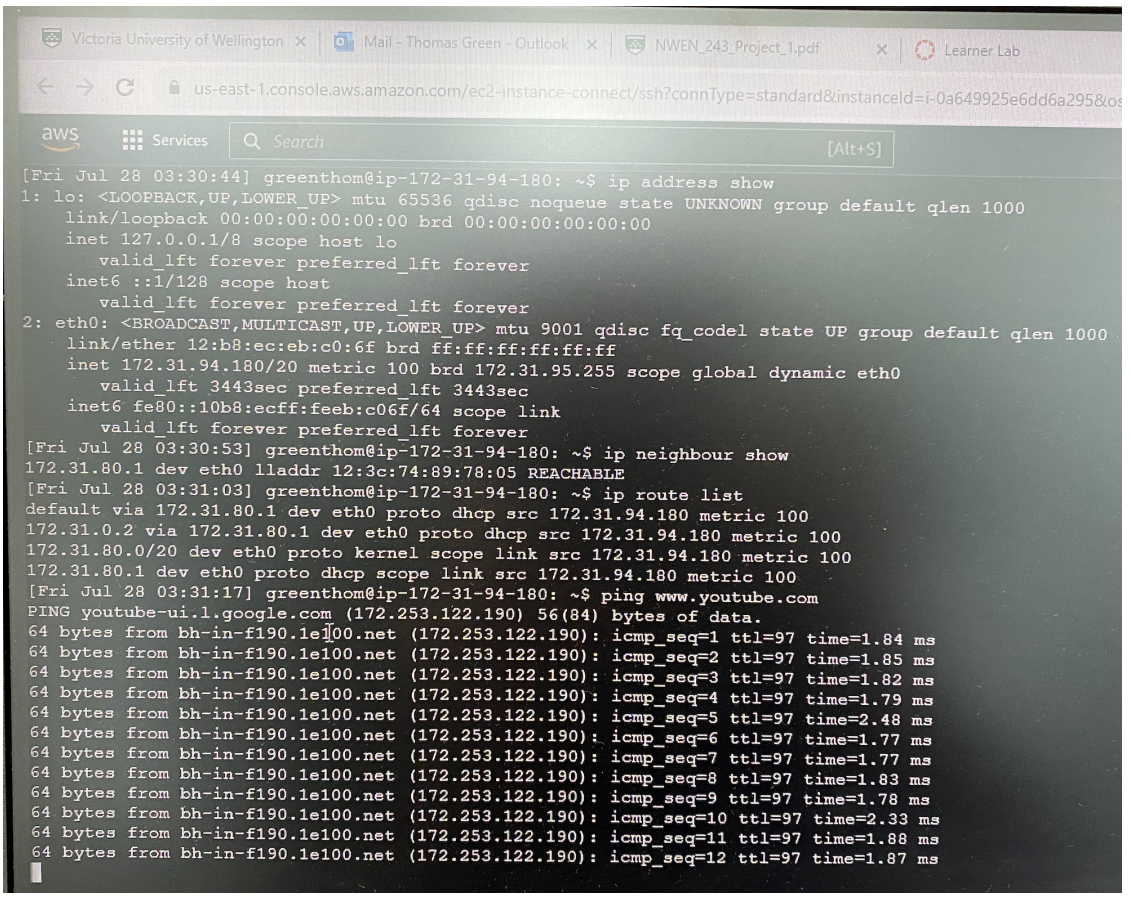
1. 

a. IP Address: 172.31.80.1. Corresponds to another device in the same local network as the VM
MAC address: 12:3c:74:89:78:05: Media Access Control address, which is a unique identifier assigned to the network interface of the device with the IP address found
b.



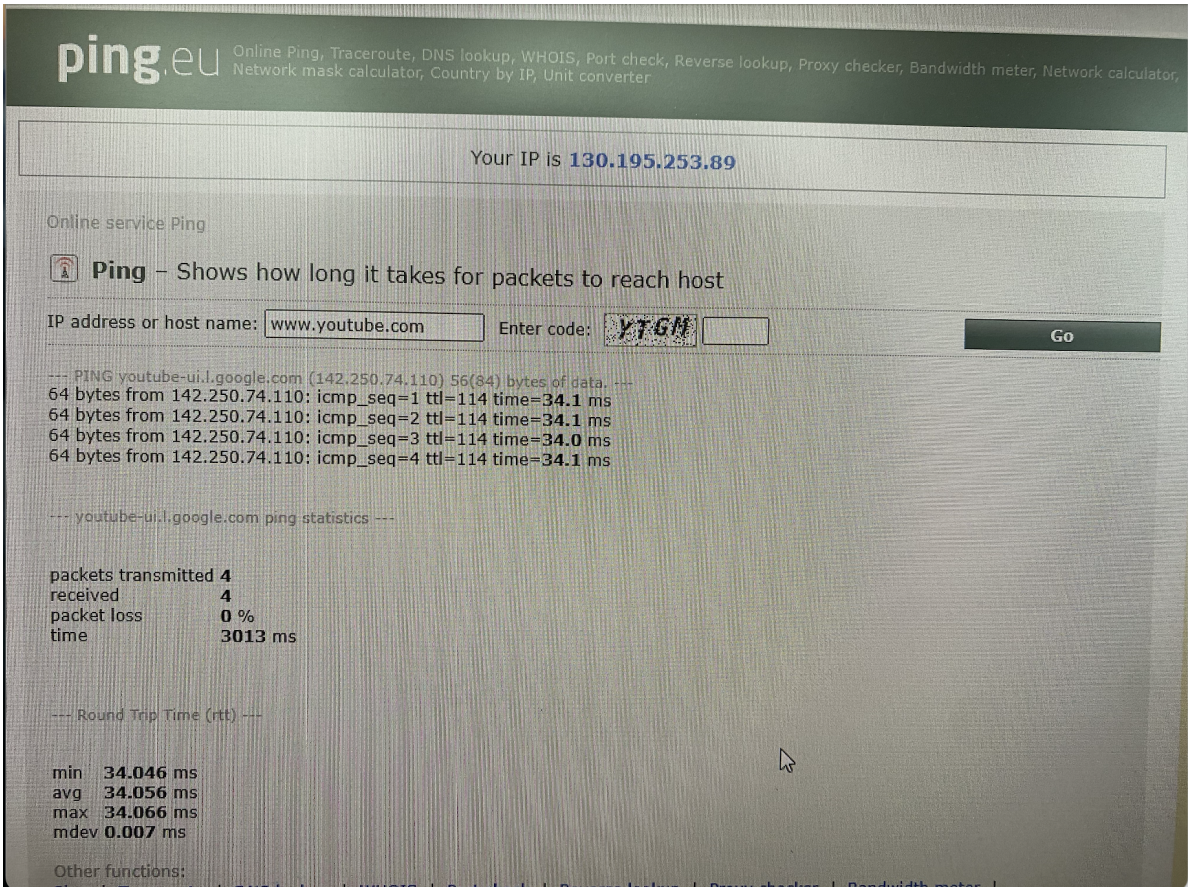
output of 'ip neighbour show' vm learned MAC address is 12:3c:74:89:78:05 associated with the IP address 172:31:80:1, entry marked as "REACHABLE",

indicating that the MAC address resolution was successful. 'ip route list' VM default route set up to send all traffic to gateway 172.31.80.1 via eth0 interface. Specified by "default via 172.31.80.1" line, showing that any traffic that doesnt match a more specific route will be sent to this gateway. Additionally, there are other routes defined for the 172.31.80.0/20 network, which includes the VM's own subnet. Routes indicate how traffic within VM's local network should be handled. "src 172.31.94.180" part specifies that the VM's source IP address for outgoing traffic will be 172.31.94.180". **Information** obtained from a) the MAC address associated with the IP address is used in conjunction with the routing information obtained from 'ip route list' to facilitate communication within the local network and beyond. When the VM needs to communicate with a device on the local network, it first checks its ARP cache to find the MAC address associated with the destination IP address. If the MAC address is not in the ARP cache or is marked as "STALE", the VM will send an ARP request to resolve the MAC address. Once the MAC address is known, the VM can forward packets to the appropriate destination on the local network. Furthermore, information obtained from 'ip route list' is used to determine the next hop for packets destined for networks outside the local network. In this case, the default route specifies that any traffic destined for external networks should be sent to the gateway IP address 172.31.80.1 (router for VM's network). Router will then handle forwarding packets to their respective destinations on the internet or other external networks.

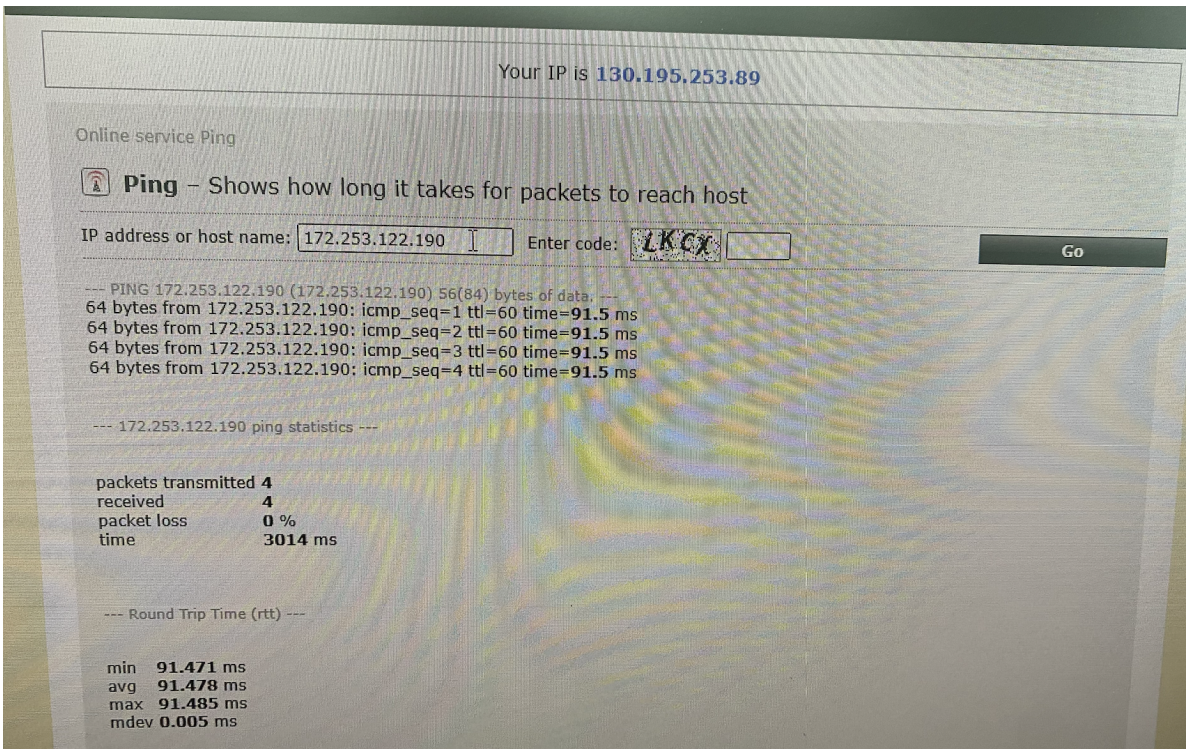
1. 

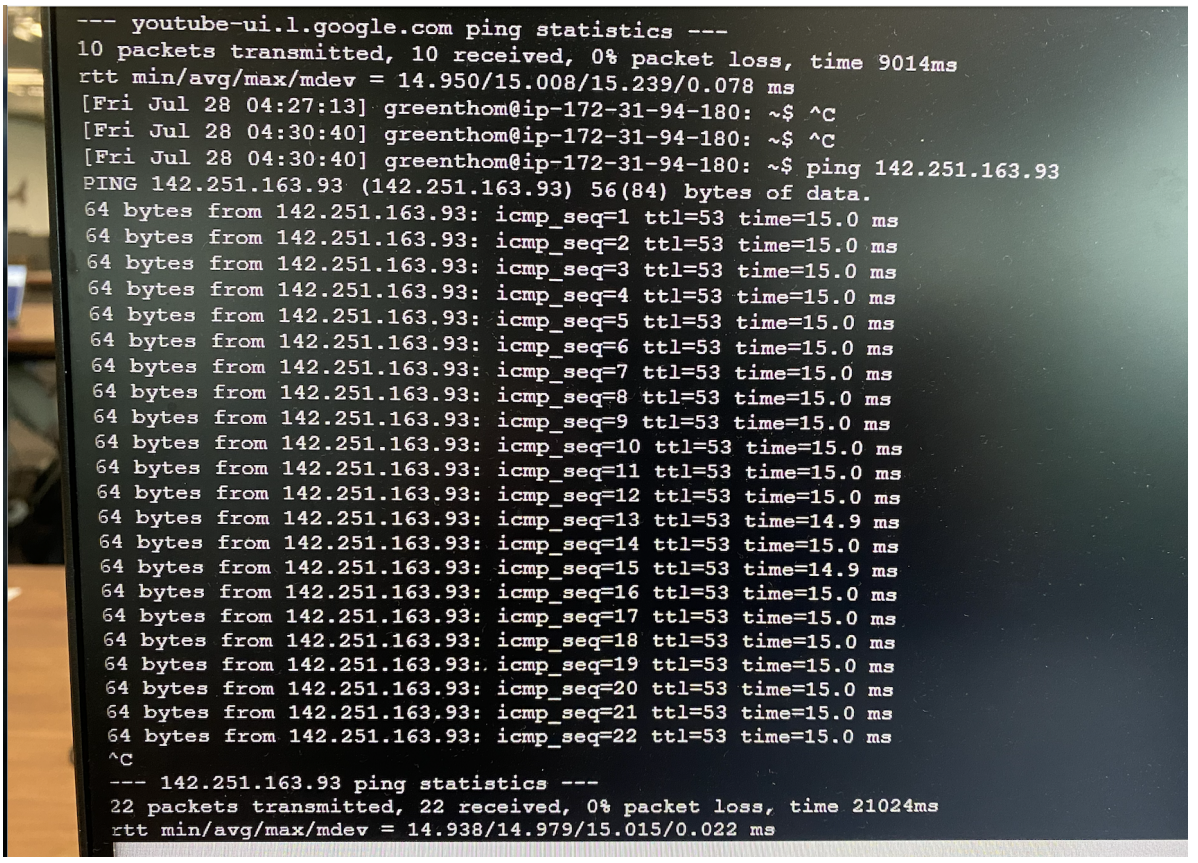
a. Based on the output, chosen destination IP address for www.youtube.com is 142.251.163.93. This IP address (142.251.163.93) is IP1.

b.



Based on the output, chosen destination IP address for www.youtube.com is 142.250.74.46. This IP address (142.250.74.46) is IP2.





c. All answers for Virtual Machine have been stopped using command c. RTL_VM_IP1: 15.313 ms. RTL_VM_IP2: 95.619 ms. RTL_LOCAL_IP1: 91.268 ms. RTL_LOCAL_IP2: 31.315 ms.

d. Different destination networks since RTLs for IP1 and IP2 suggest they belong to different networks. The routes and configuration for both IP1 and IP2

might vary leading to different RTL times. Network Congestion would lead to varying RTLs. If the network carrying traffic to IP1 or IP2 experiences congestion or high traffic during when I ran it it would impact the times. Distance between RTLs could be an impact as the distance between VM or local machine and servers of destination IP (IP1 and IP2) which would result in higher latency, reflected in the RLS. Different routes might be taken to reach IP1 and IP2 depending on their networks and configuration which could have effected latency experienced in the RTL. Load on the servers at the destination IP addresses which could have lead to increased response time.

7.

a.

```
[Sun Jul 30 10:18:47] greenthom@ip-172-31-94-180: ~$ traceroute 142.251.163.93
traceroute to 142.251.163.93 (142.251.163.93), 30 hops max, 60 byte packets
 1  244.5.1.79 (244.5.1.79)  8.394 ms 216.182.229.217 (216.182.229.217)  3.130 ms 216.182.229.224 (216.182.229.224)  7.371 ms
 2  100.65.80.16 (100.65.80.16)  36.835 ms 100.66.9.24 (100.66.9.24)  15.928 ms 100.65.54.112 (100.65.54.112)  1.824 ms
 3  100.66.14.150 (100.66.14.150)  12.902 ms 240.1.208.15 (240.1.208.15)  16.230 ms 100.66.24.176 (100.66.24.176)  63.090 ms
 4  15.230.56.76 (15.230.56.76)  14.527 ms 100.66.30.68 (100.66.30.68)  8.017 ms 241.0.5.19 (241.0.5.19)  0.425 ms
 5  100.66.6.79 (100.66.6.79)  5.307 ms 15.230.56.78 (15.230.56.78)  15.790 ms 240.0.28.31 (240.0.28.31)  0.512 ms
 6  240.0.28.16 (240.0.28.16)  0.495 ms 240.0.28.27 (240.0.28.27)  0.311 ms 100.66.4.45 (100.66.4.45)  1.135 ms
 7  99.83.95.207 (99.83.95.207)  14.495 ms 99.83.95.209 (99.83.95.209)  16.347 ms 100.65.89.130 (100.65.89.130)  5.466 ms
 8  108.170.249.97 (108.170.249.97)  16.817 ms 108.170.249.163 (108.170.249.163)  15.028 ms 108.170.249.98 (108.170.249.98)  14.613 ms
 9  108.170.249.44 (108.170.249.44)  15.692 ms 100.66.54.22 (100.66.54.22)  18.467 ms *
10  241.0.4.130 (241.0.4.130)  0.429 ms 108.170.229.80 (108.170.229.80)  17.776 ms 216.239.43.24 (216.239.43.24)  16.446 ms
11  108.170.249.44 (108.170.249.44)  14.706 ms 240.1.208.14 (240.1.208.14)  16.148 ms *
12  15.230.56.94 (15.230.56.94)  29.127 ms 142.251.51.20 (142.251.51.20)  17.106 ms 240.1.208.13 (240.1.208.13)  16.186 ms
13  216.239.56.134 (216.239.56.134)  15.922 ms 108.170.229.80 (108.170.229.80)  18.295 ms 142.250.209.70 (142.250.209.70)  16.016 ms
14  142.251.254.57 (142.251.254.57)  17.026 ms 142.251.254.59 (142.251.254.59)  17.692 ms 142.250.209.42 (142.250.209.42)  16.978 ms
15  * * 99.83.95.207 (99.83.95.207)  15.435 ms
16  * * 108.170.249.76 (108.170.249.76)  14.578 ms
17  142.250.235.85 (142.250.235.85)  17.082 ms 74.125.37.158 (74.125.37.158)  15.898 ms *
18  * * 142.250.235.85 (142.250.235.85)  18.297 ms
19  * * *
20  * * *
21  216.239.56.72 (216.239.56.72)  16.405 ms * *
22  * * *
23  * * *
24  * * wv-in-f93.1e100.net (142.251.163.93)  15.099 ms
[Sun Jul 30 10:19:01] greenthom@ip-172-31-94-180: ~$
```

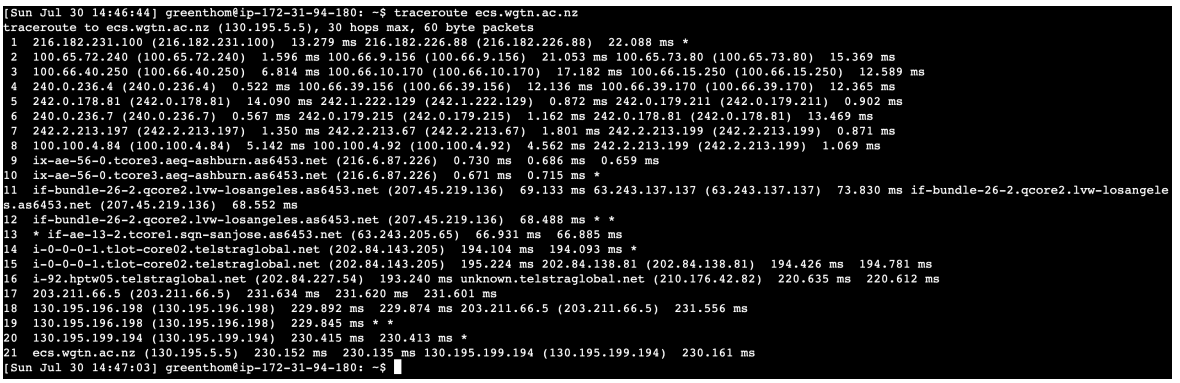
b.

1				*	*	*
2	core24.fsn1.hetzner.com core23.fsn1.hetzner.com core24.fsn1.hetzner.com	213.239.245.241 213.239.245.237 213.239.245.241	de de de	**1.760** ms **3.092** ms **1.760** ms		
3	core1.fra.hetzner.com core5.fra.hetzner.com core1.fra.hetzner.com	213.239.224.70 213.239.224.78 213.239.224.86	de de de	**5.000** ms **4.863** ms **4.972** ms		
4				*	*	*
5				*	*	*
6				*	*	*
7				*	*	*
8				*	*	*

c. VM Machine Traceroute: - The traceroute from the virtual machine shows multiple intermediate routers identified by their IP addresses. - There are several hops before reaching the destination IP, and some router respond with their IP addresses while others do not respond. - The vitual machine's traceroute path provides more detailed information about the intermediate routers traversed, indicating more specific network details. Website Traceroute: - THe traceroute from the website also shows intermediate routers but does not provide detailed information like IP addresses or hostnames. - Some of the intermediate routers respond with their hostnames, while others show * for no response. - THe website traceroute path provides a more user-friendly view but lacks detailed information about the routers involved. Comparison and Common Links: - Both tracerouters show a similar pattern in terms of unreachable (no response) hops after a certain point in the path. - The common links toward the beginning of the path are likely the routers that respond with their IP addresses or hostnames. - As the traceroutes progress futher towards the destination, there are several hops that do not respond. - The common links are likely the initial routers closer to the source that are directly accessible and respond to the traceroute requests. Reasoning: The common links (routers that respond) toward the beginning of the path are the routers through which the packets successfully pass, and they belond to a more direct and accessible path from the source to the desination IP address (142.251.163.93). As the traceroutes go further and encounter hops that do not respond, it indicates that some routers might not allow traceroute requests to pass through, or they are configured not to respond for security or privacy reasons. Overall, the common links (responding routers) indicate the initial portion of the path, which is accessible and more likely part of the direct route to the destination. THe hops that do not respond later in the path might represent firewalls, network filtering, or routers that do not respond to traceroute requests, making them unreachable in the traceroute results.

8.

a.



b.

c. For ecs.wgtn.ac.nz trace-route: - The common link towards the beginning is likely within the local network or the first few hops (1-4) before reaching the public internet. - The paths start to diverge after hop 5, and there are different routers and IP addresses observed in the subsequent hops. - The paths do not converge towards the end, and they remain different until reaching the final desitination of ecs.wgtn.ac.nz (130.195.5.5). For www.wgtn.ac.nz trace-route: - There are no common links towards the beginning, as each of the first four hops shows different routers and IP addresses. - The paths continue to diverge, and there is no indication of convergence at any point along the path. - The trace-route is incomplete, showing asterisks (*) for all subsequent hops, indicating that no response was received from the routers beyond the initial hops. Justification: The trace-route results indicate the paths of both addresses are different and do not converge at any point. This suggests that the two destinations are likely hosted on separate networks or server clusters, with different routes taken from your VM to each destination. The lack of common links and divergence in the paths

from the beginning suggest that there is no overlap or shared infrastructure between the two desinations. Therefore, the paths to both the addresses are independent of each other, and there is no evidence of any common links or shared route among the paths.

1. 
```
[Sun Jul 30 14:58:03] greenthom@ip-172-31-94-180: ~$ traceroute google.com
traceroute to google.com (172.253.62.138), 30 hops max, 60 byte packets
 1  * * 216.182.229.238 (216.182.229.238)  1.874 ms
 2  * 100.66.32.194 (100.66.32.194)  1.220 ms 100.66.32.254 (100.66.32.254)  1.359 ms
 3  240.0.184.4 (240.0.184.4)  0.537 ms * *
 4  241.0.5.9 (241.0.5.9)  0.413 ms 241.0.5.24 (241.0.5.24)  0.401 ms 241.0.5.15 (241.0.5.15)  0.387 ms
 5  240.0.184.4 (240.0.184.4)  0.518 ms  0.531 ms 240.0.184.5 (240.0.184.5)  0.899 ms
 6  100.100.36.104 (100.100.36.104)  1.427 ms 240.0.184.4 (240.0.184.4)  0.414 ms 100.100.36.110 (100.100.36.110)  1.022 ms
 7  99.82.180.135 (99.82.180.135)  1.534 ms 99.82.180.131 (99.82.180.131)  1.488 ms 99.82.180.135 (99.82.180.135)  1.562 ms
 8  99.82.180.135 (99.82.180.135)  1.533 ms * 99.82.180.131 (99.82.180.131)  1.281 ms
 9  108.170.246.33 (108.170.246.33)  2.313 ms 142.251.67.234 (142.251.67.234)  1.116 ms 216.239.48.101 (216.239.48.101)  2.156 ms
10  108.170.246.33 (108.170.246.33)  2.055 ms 108.170.240.97 (108.170.240.97)  2.211 ms 108.170.246.67 (108.170.246.67)  1.376 ms
11  142.251.49.75 (142.251.49.75)  2.352 ms 216.239.48.101 (216.239.48.101)  2.120 ms 142.251.49.187 (142.251.49.187)  2.092 ms
12  142.250.236.147 (142.250.236.147)  2.603 ms 142.250.209.75 (142.250.209.75)  2.555 ms 142.251.49.209 (142.251.49.209)  2.397 ms
13  142.251.77.148 (142.251.77.148)  11.281 ms 209.85.244.151 (209.85.244.151)  2.316 ms *
14  142.251.77.148 (142.251.77.148)  21.039 ms 172.253.67.20 (172.253.67.20)  2.545 ms 142.251.77.104 (142.251.77.104)  2.745 ms
15  142.251.245.101 (142.251.245.101)  1.558 ms 142.250.209.96 (142.250.209.96)  2.016 ms 142.250.209.42 (142.250.209.42)  2.461 ms
16  * 172.253.68.73 (172.253.68.73)  1.765 ms *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * bc-in-f138.1e100.net (172.253.62.138)  1.949 ms  1.599 ms
```

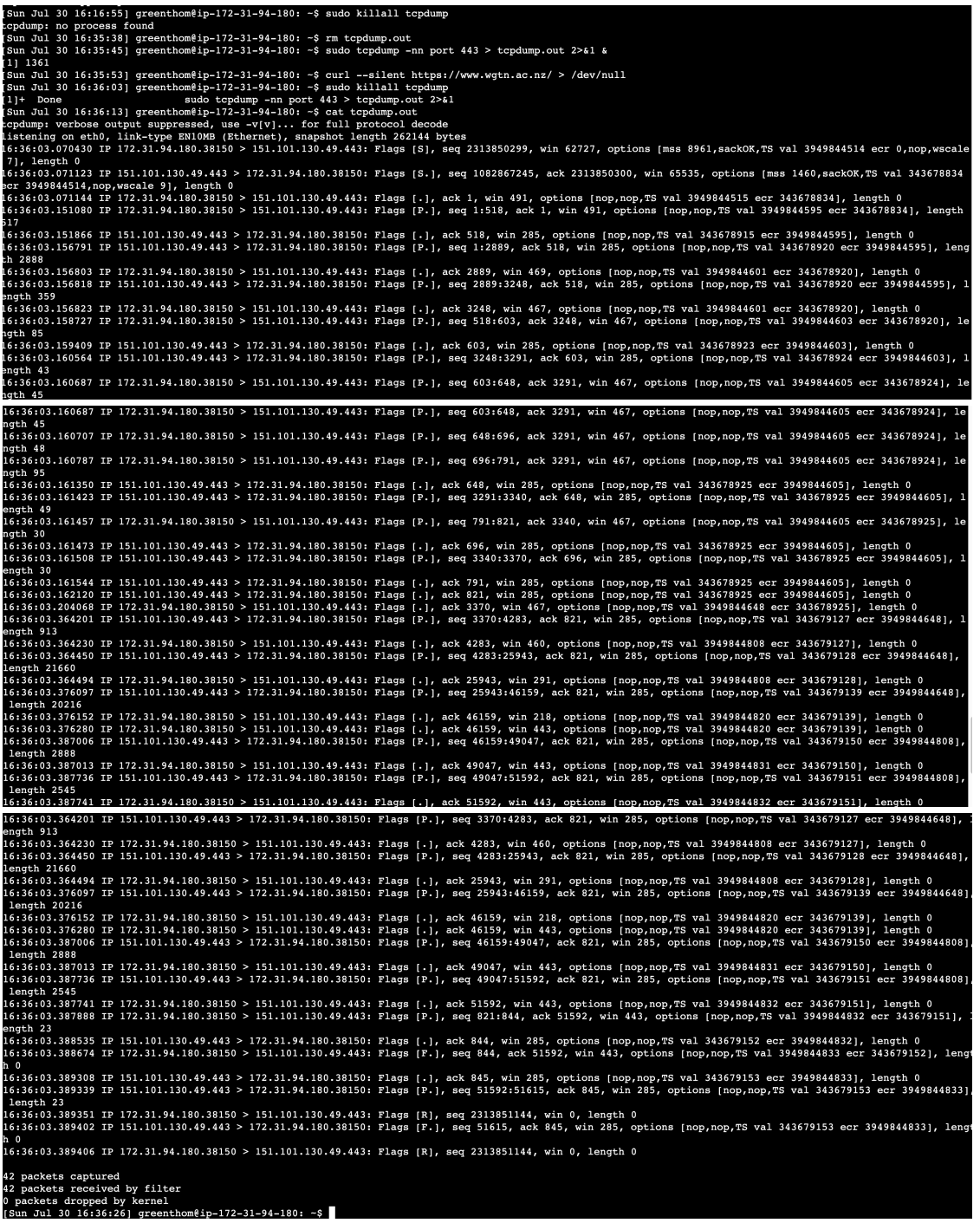
Based on the ouput we can observe evidence of multiplicity and path redundances: - Multiple Timeouts (*): In the traceroute output, you can see several hops where the response times are represented by *, which indicates that the packets either didnt reach those hops or the routers at those hops didnt respond within the specified time limit. This could be due to network congestion, router prioritization, or load balancing mechanisms. The presence of timeouts suggests that there are alternative paths available for the packets to take, and different packets may traverse different routes. -Varying IP Addresses: For several hops in the traceroute, you can observe varying IP addresses, even within a single TTL value. For example, in hop 8, the IP addresses 99.82.180.135 and 99.82.180.131 appear in different attempts, indicating that the packets took different paths during those attempts. This demonstrates that there are multiple paths to reach the same hop, offering redundancy and fault tolerance. -Load Balancing and Dynamic Routing: The changing of IP addresses and response times for the same hops in different attempts suggest that the network employs load balancing and dynamic routing techniques. These techniques distribute traffic across multiple paths to optimize performance and ensure resilience. Different packets may be routed differently, leading to multiplicity in the networks paths. -Asymmetric Paths: In some instances, you may notice that the path taken in the forward direction (e.g. from your VM to an intermediate router) is different from the path taken in the reverse direction (e.g. from the intermediate router back to your VM). This indicates that the network has asymmetrical routing, where packets follow different paths in each direction, further showcasing multiplicity in the networks topology.

1. 

a.Identifying the packets: Packet 1: Source:172.31.94.180 Destination:172.31.0.2 Purpose: DNS query for the A record (IPv4 address) of the domain name www.wgtn.ac.nz . Packet 2: Source:172.31.94.180 Destination: 172.31.0.2 Purpose: DNS query for the AAAA record (IPv6 address) of the domain name www.wgtn.ac.nz Packet 3: Source:172.31.0.2 Destination: 172.31.94.180 Purpose: DNS response to Packet 1, providing the A records of the domain name www.wgtn.ac.nz. The response contains four IPv4 addresses: 151.101.66.49, 151.101.130.49, 151.101.194.49 and 151.101.2.49 Packet 4: Source: 172.31.0.2 Destination: 172.31.94.180 Purpose: DNS response to packet 2, indicating that there are no AAAA records (IPv6 addresses) available for the domain name www.wgtn.ac.nz

b. Based on the output the transport layer protocol used by the captured packets is UDP. We see that the 'proto' field is indicated as UDP (18) for all packets, which means they are using the UDP protocol. UDP is the appropriate choice for their purpose because it is a connectionless, lightweight, and fast transport layer protocol. DNS queries and responses typically do not require the reliability and connection setup/teardown overhead that TCP (Transmission Control Protocol) provides. DNS is based on a simple request-response model where a client (in this case, the VM) sends a query to a DNS server, and the server responds with the requested information. SInce DNS queries are short-lived and usually do not involve large amounts of data, UDP is well-suited for this purpose. Using UDP for DNS allows a faster query resolution and reduced overhead compared to TCP. In DNS< if a response packet is lost due to the connectionless nature of UDP, the client can simply re-send the query, and the DNS server will provide the response again. For these reasons, UDP is commonly chosen for DNS queries and responses.

1.



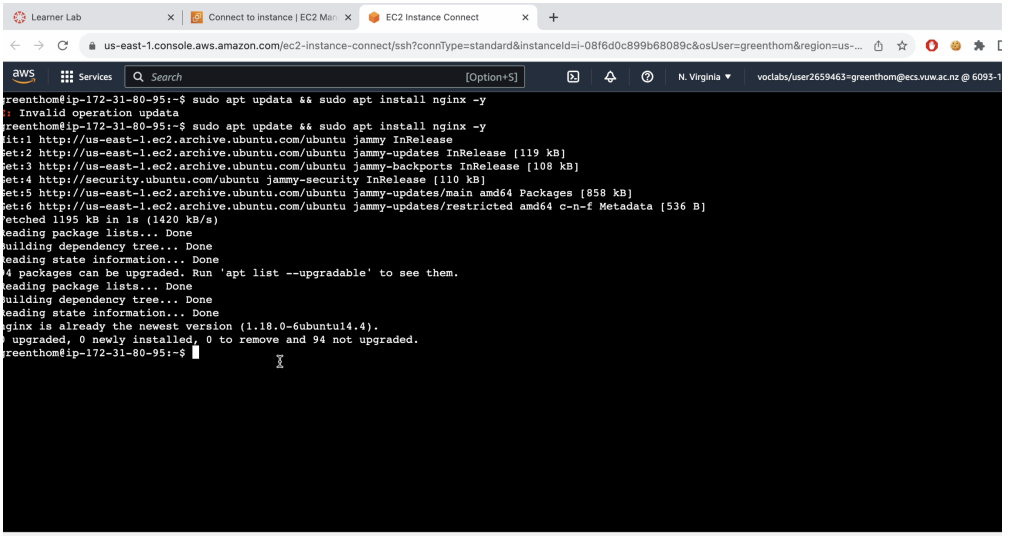
Packet 1: SourceIP: 172.31.94.180 Destination IP: 151.101.130.49 Source Port: 38150 Destination Port: 443 TCP Flags: [S] (SYN) Purpose: This packet is the initial SYN packet of a TCP three-way handshake. It is sent by your VM (source) to the web server www.wgtn.ac.nz to initiate a connection. Packet 2: SourceIP: 151.101.130.49 Destination IP: 172.31.94.180 Source Port: 443 Destination Port: 38150 TCP Flags: [S.] (SYN-ACK) Purpose: This packet is the SYN-ACK packet in response to Packet 1. The web server (source) is acknowledging the VM's request for a connection (destination) Packet 3:

SourceIP: 172.31.94.180 Destination IP: 151.101.130.49 Source Port: 38150 Destination Port: 443 TCP Flags: [.] ACK (Acknowledgment) Purpose: This packet is an ACK packet sent by the VM to acknowledge the web server's response (destination) and complete the TCP three-way handshake, establishing the connection

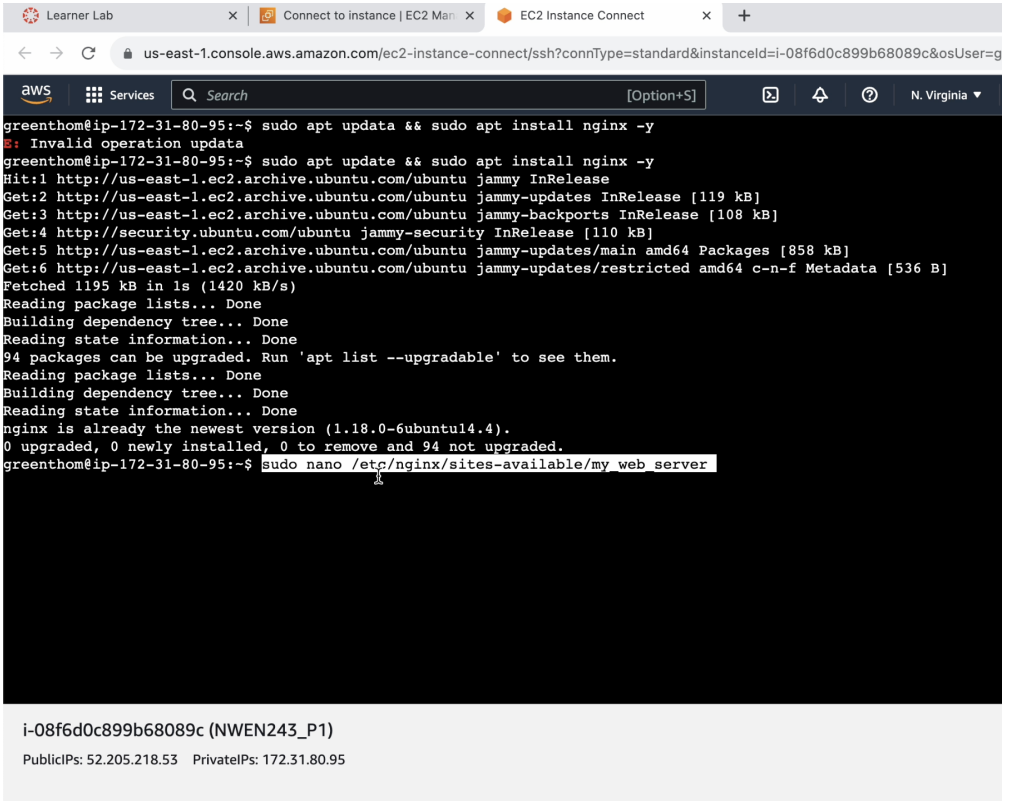
1. To understand how the sequence number and acknowledgement numbers are calculated, we need to analyze the TCP headers of the first 10 packets captured. The sequence number and acknowledgement number fields in the TCP header are used to manage reliable data transmission and ensure that packets are received in the correct order. MAKE SURE TO LOOK INTO THE 10 PACKET ANSWER NOT THE FUCKING THREE IM ABOUT TO DO U DONUT. Packet 1: SourceIP: 172.31.94.180 DestinationIP: 151.101.130.49 SourcePort: 38150 DestinationPort: 443 TCP Flags: [S] Purpose: This is the initial SYN packet. Packet 2: SourceIP: 151.101.130.49 DestinationIP: 172.31.94.180 Source Port: 443 Destination Port: 38150 TCP Flags: [S.] Purpose: This is the SYN-ACK packet in response to packet 1. Packet 3: SourceIP: 172.31.94.180. DestinationIP: 151.101.130.49 SourcePort: 38150 DestinationPort:443 TCP Flags: [.], ACK Purpose: This is the ACK packet to achnowledge the SYN-ACK packet. In the TCP three-way handshake, the sequence number is randomly generated by the sender and represents the first byte of data in the packet. The acknowledgment number, on the other hand, is calculated by adding 1 to the sequence number received in the SYN packet. Now lets examine the first 10 BUT WE ARE ONLY DOING FUCKING THREE I NEED TO CHECK THIS and calculate the sequence and acknowledgment numbers. Packet 1: Sequence Number: Randomly generated(X). Acknowledgmenet Number: Not applicable (SYN packet) Packet 2: Sequence Number: Randomly generated (Y) Acknowledgment Number: X+1 (Acknowledging the SYN packet) Packet 3: Sequence Number: Y+1 (Acknowledging the SYN-ACK packet) FOr the subsequent packets, the sequence number will be increased by the size of the data carried in the previous packet. THe acknowledgment number will be the next expected sequence number, which is calculated by adding the size of the data in the received packet. This process ensures the data is delivered reliably and in the correct order between the two communicating parties

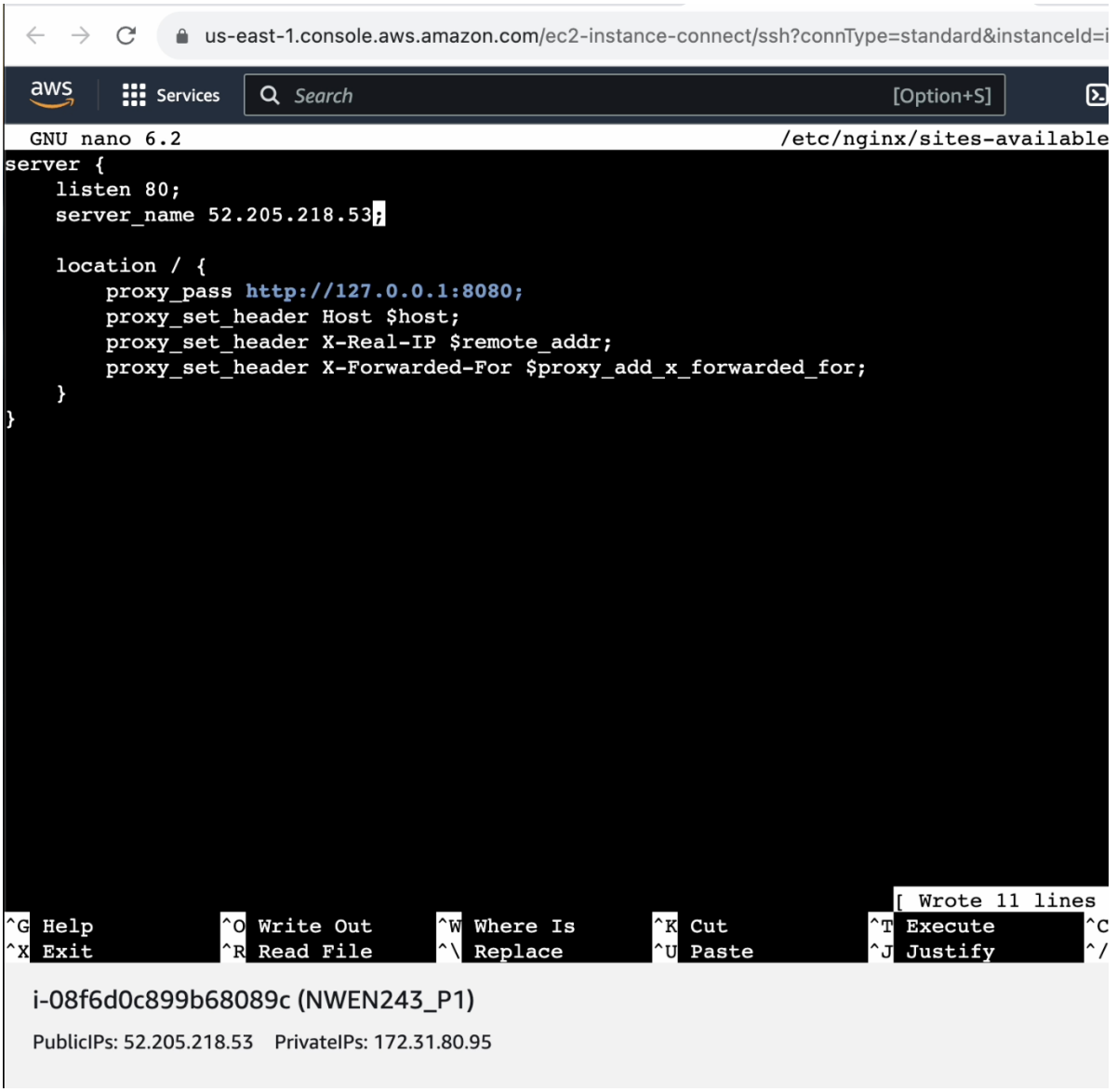
1.

1.

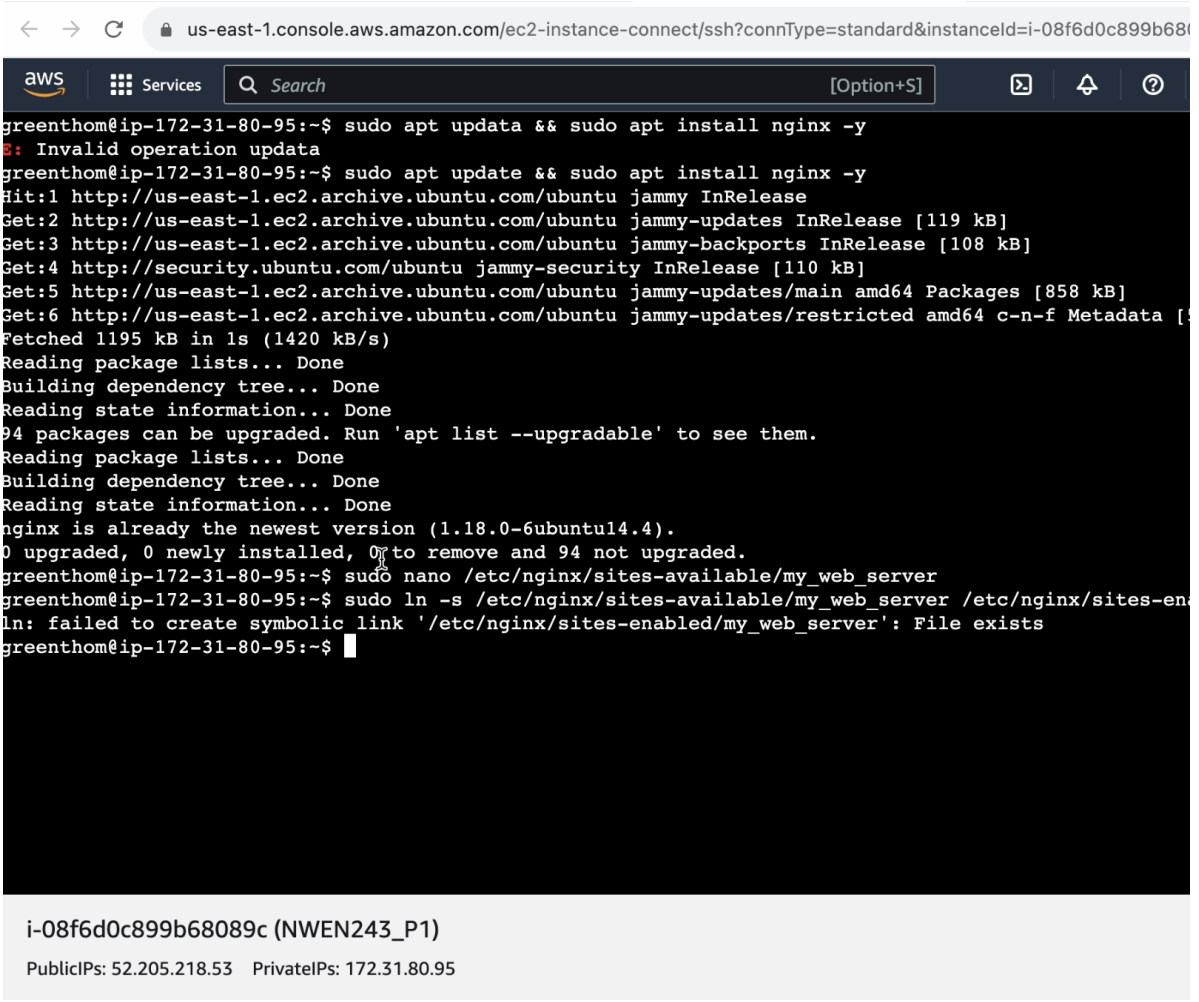

Learner Lab
Connect to instance | EC2 Man
EC2 Instance Connect
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-08f6d0c899b68089c&osUser=greenthom&region=us-...
aws
Services
Search
[Option+S]
N. Virginia
voclabs/user2659463=greenthom@ecs.vuw.ac.nz @ 6093-1
reenthom@ip-172-31-80-95:~$ sudo apt updata && sudo apt install nginx -y
: Invalid operation updata
reenthom@ip-172-31-80-95:~$ sudo apt update && sudo apt install nginx -y
it:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
et:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
et:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
et:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
et:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [858 kB]
et:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [536 B]
etched 1195 kB in 1s (1420 kB/s)
eading package lists... Done
uilding dependency tree... Done
eading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
eading package lists... Done
uilding dependency tree... Done
eading state information... Done
ginx is already the newest version (1.18.0-6ubuntu14.4).
upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
reenthom@ip-172-31-80-95:~$
i-08f6d0c899b68089c (NWEN243_P1)
PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95

2.

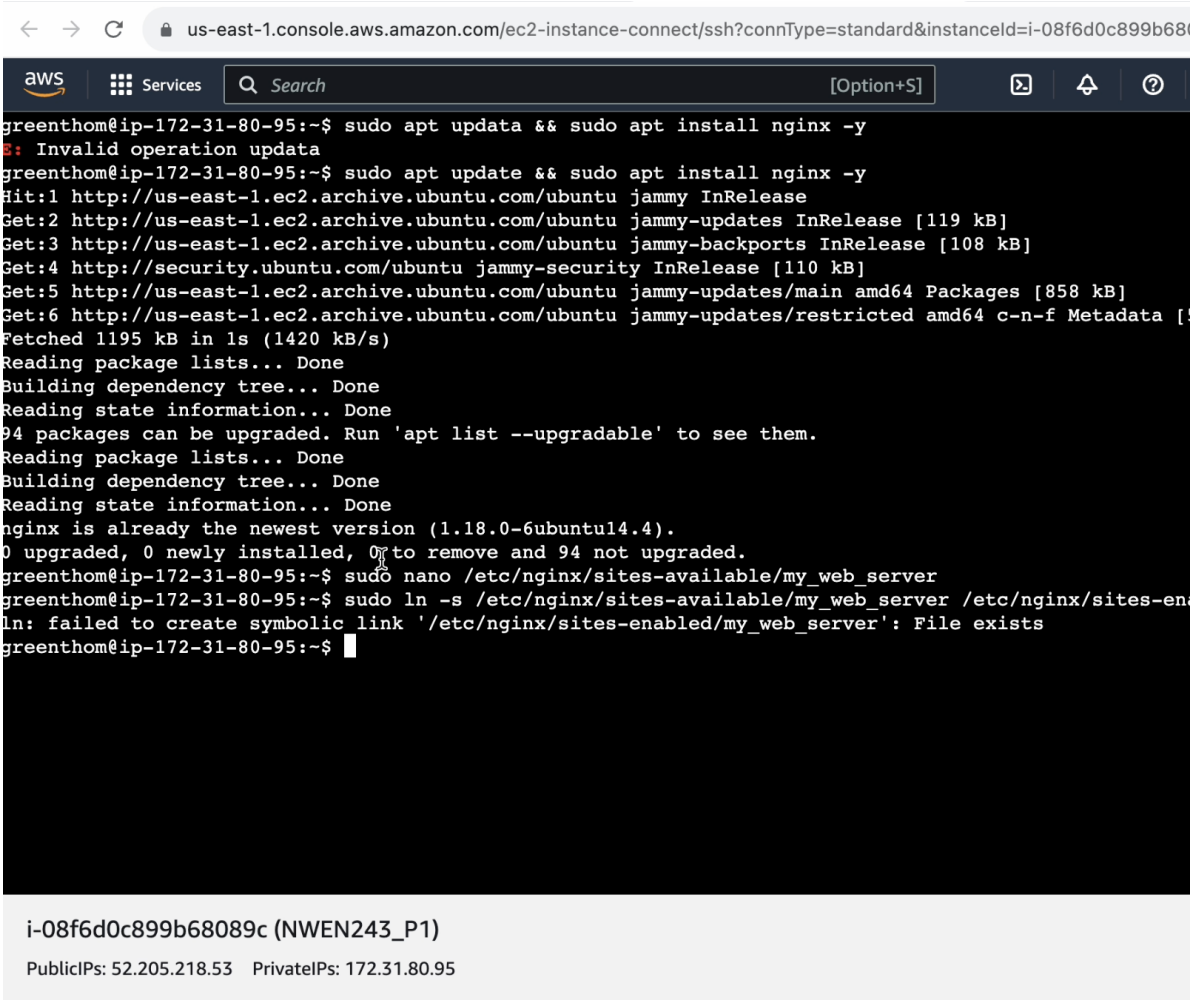

Learner Lab
Connect to instance | EC2 Man
EC2 Instance Connect
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-08f6d0c899b68089c&osUser=g
aws
Services
Search
[Option+S]
N. Virginia
greenthom@ip-172-31-80-95:~$ sudo apt updata && sudo apt install nginx -y
E: Invalid operation updata
greenthom@ip-172-31-80-95:~$ sudo apt update && sudo apt install nginx -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [858 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [536 B]
Fetched 1195 kB in 1s (1420 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
greenthom@ip-172-31-80-95:~$ sudo nano /etc/nginx/sites-available/my_web_server
i-08f6d0c899b68089c (NWEN243_P1)
PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95


us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i
aws
Services
Search
[Option+S]
GNU nano 6.2
/etc/nginx/sites-available
server {
listen 80;
server_name 52.205.218.53;
location / {
proxy_pass http://127.0.0.1:8080;
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
}
[Wrote 11 lines
^G Help
^O Write Out
^W Where Is
^K Cut
^T Execute
^C
^X Exit
^R Read File
^\ Replace
^U Paste
^J Justify
^/
i-08f6d0c899b68089c (NWEN243_P1)
PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95

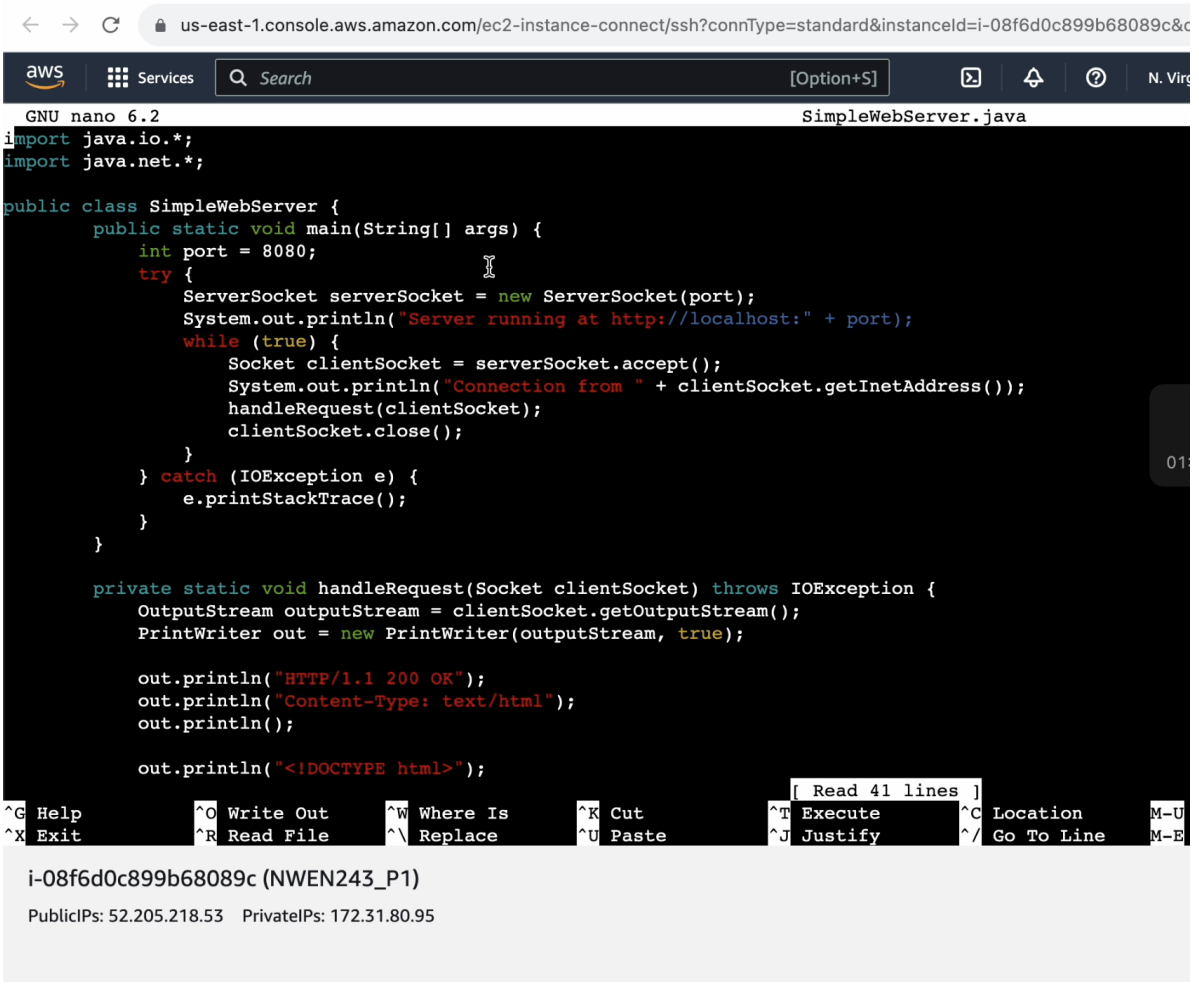
3.


us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-08f6d0c899b68
aws
Services
Search
[Option+S]
greenthom@ip-172-31-80-95:~$ sudo apt updata && sudo apt install nginx -y
E: Invalid operation updata
greenthom@ip-172-31-80-95:~$ sudo apt update && sudo apt install nginx -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [858 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [
Fetched 1195 kB in 1s (1420 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
greenthom@ip-172-31-80-95:~$ sudo nano /etc/nginx/sites-available/my_web_server
greenthom@ip-172-31-80-95:~$ sudo ln -s /etc/nginx/sites-available/my_web_server /etc/nginx/sites-ena
ln: failed to create symbolic link '/etc/nginx/sites-enabled/my_web_server': File exists
greenthom@ip-172-31-80-95:~$
i-08f6d0c899b68089c (NWEN243_P1)
PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95

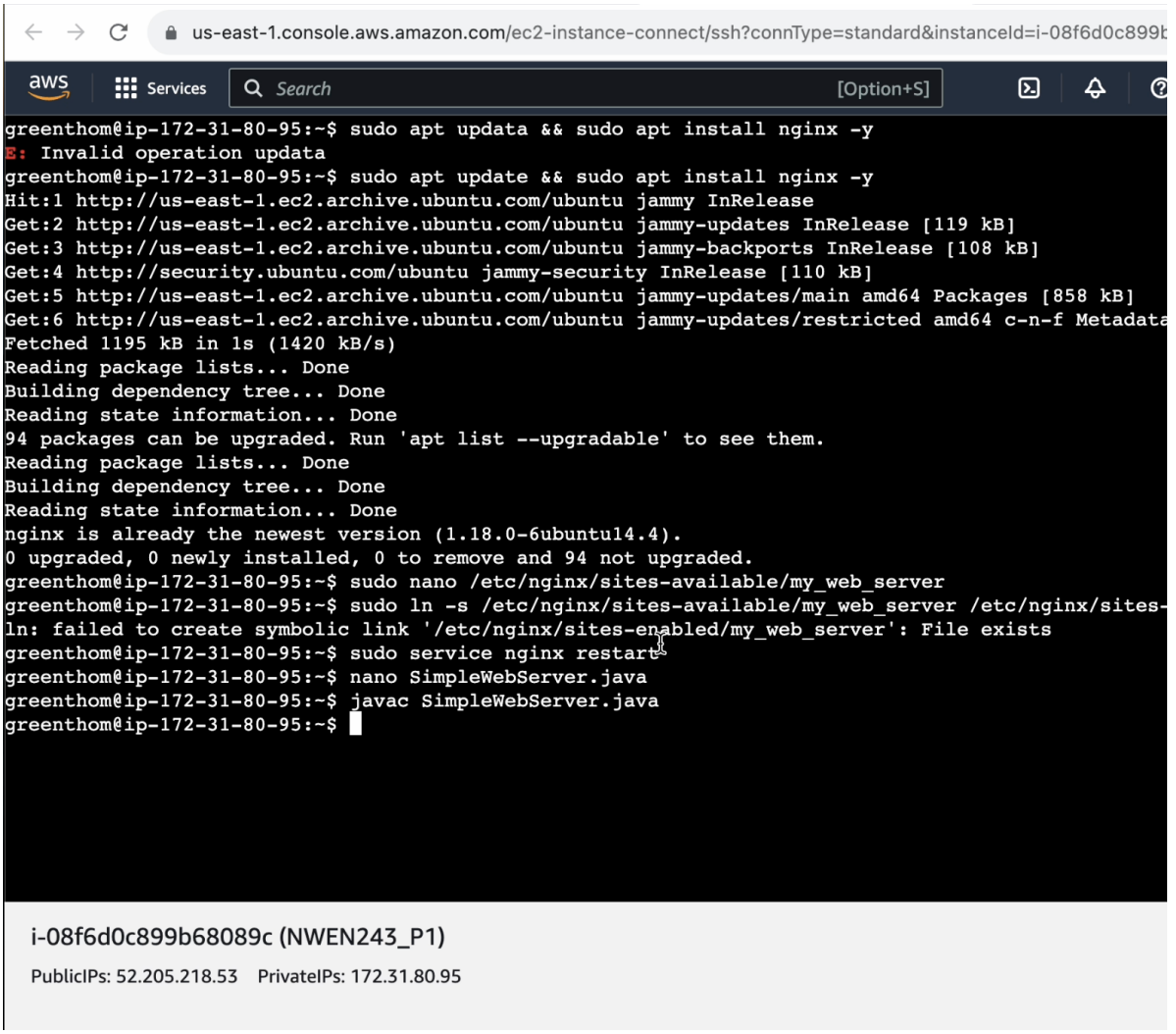
3.


us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-08f6d0c899b68
aws
Services
Search
[Option+S]
greenthom@ip-172-31-80-95:~$ sudo apt updata && sudo apt install nginx -y
E: Invalid operation updata
greenthom@ip-172-31-80-95:~$ sudo apt update && sudo apt install nginx -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [858 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [
Fetched 1195 kB in 1s (1420 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
greenthom@ip-172-31-80-95:~$ sudo nano /etc/nginx/sites-available/my_web_server
greenthom@ip-172-31-80-95:~$ sudo ln -s /etc/nginx/sites-available/my_web_server /etc/nginx/sites-ena
ln: failed to create symbolic link '/etc/nginx/sites-enabled/my_web_server': File exists
greenthom@ip-172-31-80-95:~$
i-08f6d0c899b68089c (NWEN243_P1)
PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95

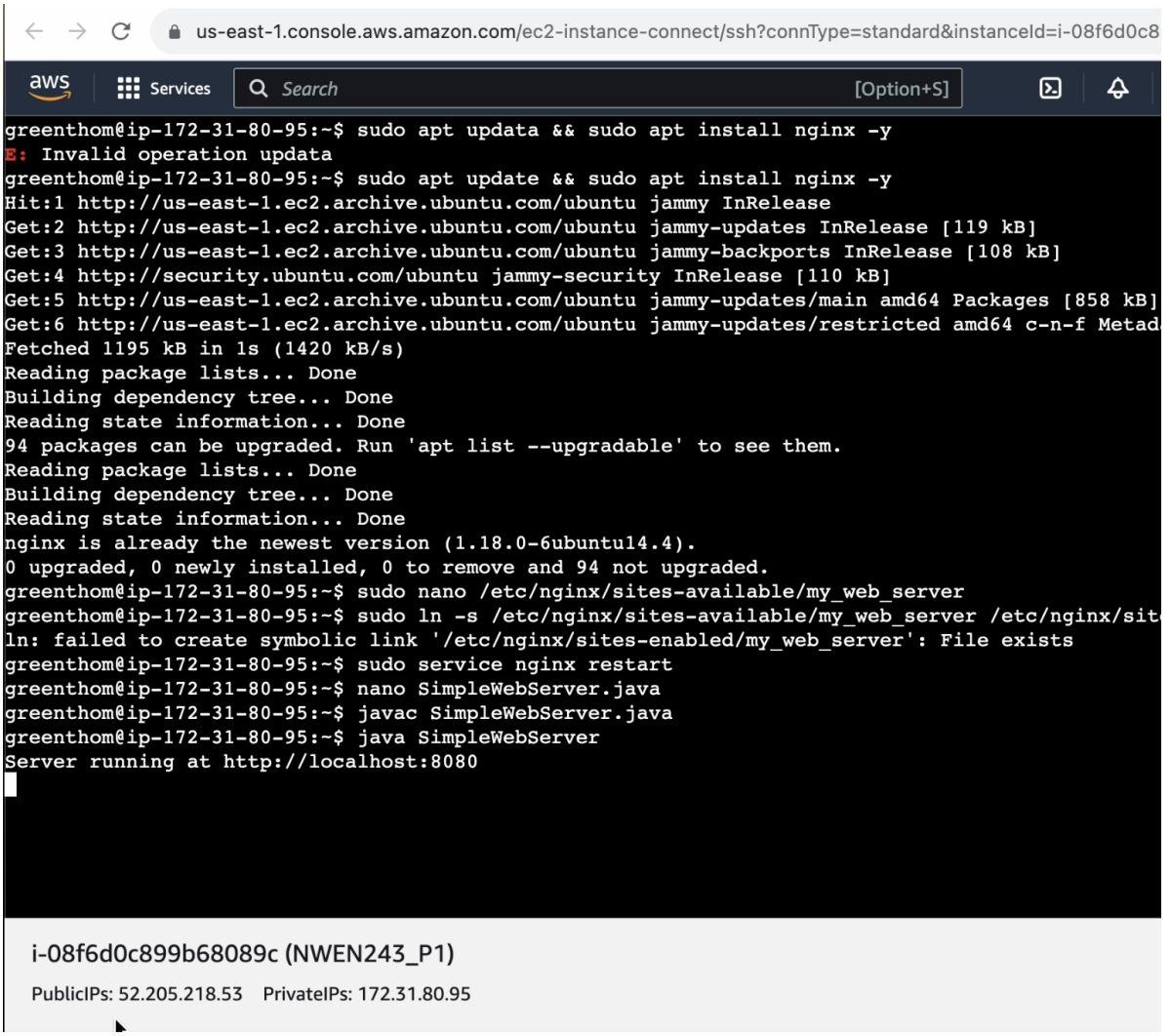
4.1.


us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-08f6d0c899b68089c&c
aws
Services
Search
[Option+S]
N. Virg
GNU nano 6.2
SimpleWebServer.java
import java.io.*;
import java.net.*;
public class SimpleWebServer {
public static void main(String[] args) {
int port = 8080;
try {
ServerSocket serverSocket = new ServerSocket(port);
System.out.println("Server running at http://localhost:" + port);
while (true) {
Socket clientSocket = serverSocket.accept();
System.out.println("Connection from " + clientSocket.getInetAddress());
handleRequest(clientSocket);
clientSocket.close();
}
} catch (IOException e) {
e.printStackTrace();
}
}
private static void handleRequest(Socket clientSocket) throws IOException {
OutputStream outputStream = clientSocket.getOutputStream();
PrintWriter out = new PrintWriter(outputStream, true);
out.println("HTTP/1.1 200 OK");
out.println("Content-Type: text/html");
out.println();
out.println("<!DOCTYPE html>");
[Read 41 lines]
^G Help
^O Write Out
^W Where Is
^K Cut
^T Execute
^C Location
M-U
^X Exit
^R Read File
^\ Replace
^U Paste
^J Justify
^/ Go To Line
M-E
i-08f6d0c899b68089c (NWEN243_P1)
PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95

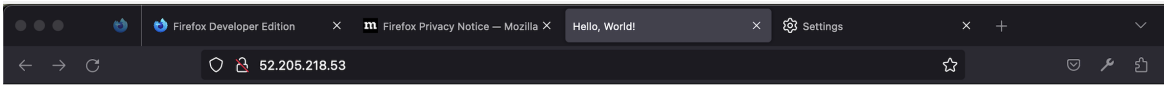
4.2.


us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-08f6d0c899b
aws
Services
Search
[Option+S]
greenthom@ip-172-31-80-95:~$ sudo apt updata && sudo apt install nginx -y
E: Invalid operation updata
greenthom@ip-172-31-80-95:~$ sudo apt update && sudo apt install nginx -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [858 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata
Fetched 1195 kB in 1s (1420 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
greenthom@ip-172-31-80-95:~$ sudo nano /etc/nginx/sites-available/my_web_server
greenthom@ip-172-31-80-95:~$ sudo ln -s /etc/nginx/sites-available/my_web_server /etc/nginx/sites-
ln: failed to create symbolic link '/etc/nginx/sites-enabled/my_web_server': File exists
greenthom@ip-172-31-80-95:~$ sudo service nginx restart
greenthom@ip-172-31-80-95:~$ nano SimpleWebServer.java
greenthom@ip-172-31-80-95:~$ javac SimpleWebServer.java
greenthom@ip-172-31-80-95:~$
i-08f6d0c899b68089c (NWEN243_P1)
PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95

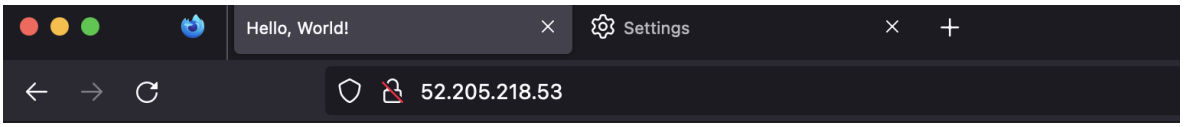
4.3.


us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-08f6d0c8
aws
Services
Search
[Option+S]
greenthom@ip-172-31-80-95:~$ sudo apt updata && sudo apt install nginx -y
E: Invalid operation updata
greenthom@ip-172-31-80-95:~$ sudo apt update && sudo apt install nginx -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [858 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metad
Fetched 1195 kB in 1s (1420 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
94 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 94 not upgraded.
greenthom@ip-172-31-80-95:~$ sudo nano /etc/nginx/sites-available/my_web_server
greenthom@ip-172-31-80-95:~$ sudo ln -s /etc/nginx/sites-available/my_web_server /etc/nginx/sit
ln: failed to create symbolic link '/etc/nginx/sites-enabled/my_web_server': File exists
greenthom@ip-172-31-80-95:~$ sudo service nginx restart
greenthom@ip-172-31-80-95:~$ nano SimpleWebServer.java
greenthom@ip-172-31-80-95:~$ javac SimpleWebServer.java
greenthom@ip-172-31-80-95:~$ java SimpleWebServer
Server running at http://localhost:8080
i-08f6d0c899b68089c (NWEN243_P1)
PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95

4.4.



Hello, World!



Hello, my name is Thomas Green!

3.

```
GNU nano 6.2                                                SimpleWebServer.java *
        private static void handleRequest(Socket clientSocket) throws IOException {
            InputStream inputStream = clientSocket.getInputStream();
            BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));

            String line;
            while ((line = reader.readLine()) != null) {
                if (line.startsWith("X-Real-IP: ")) {
                    String clientIPAddress = line.substring(11);

            OutputStream outputStream = clientSocket.getOutputStream();
            PrintWriter out = new PrintWriter(outputStream, true);

            out.println("HTTP/1.1 200 OK");
            out.println("Content-Type: text/html");
            out.println();

            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Hello, World!</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Hello, your IP address is: " + clientIPAddress + "</h1>");
            out.println("</body>");
            out.println("</html>");

            out.close();
            return;
           }}
^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Locatio
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To L
```

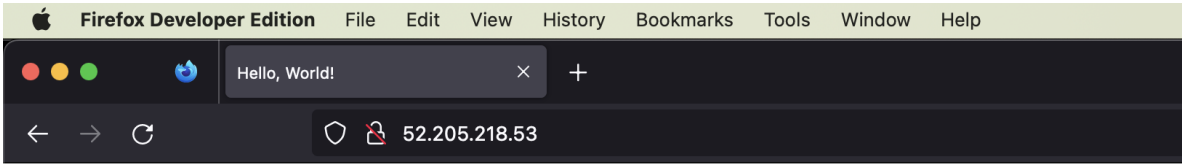
i-08f6d0c899b68089c (NWEN243_P1)

PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95

```
greenthom@ip-172-31-80-95:~$ nano SimpleWebServer.java
greenthom@ip-172-31-80-95:~$ javac SimpleWebServer.java
greenthom@ip-172-31-80-95:~$ java SimpleWebServer
Server running at http://localhost:8080
Connection from /127.0.0.1
```

i-08f6d0c899b68089c (NWEN243_P1)

PublicIPs: 52.205.218.53 PrivateIPs: 172.31.80.95



Hello, your IP address is: 130.195.253.21

3. First off i just wanted to state that the fact you can search http://ip-api.com/json/ into your browser and it comes up with the country your in, but the city, the area your in, the latitude and longitude coordinates is incredibly tapped. The definition of tapped: Someone/Something that is messed up.