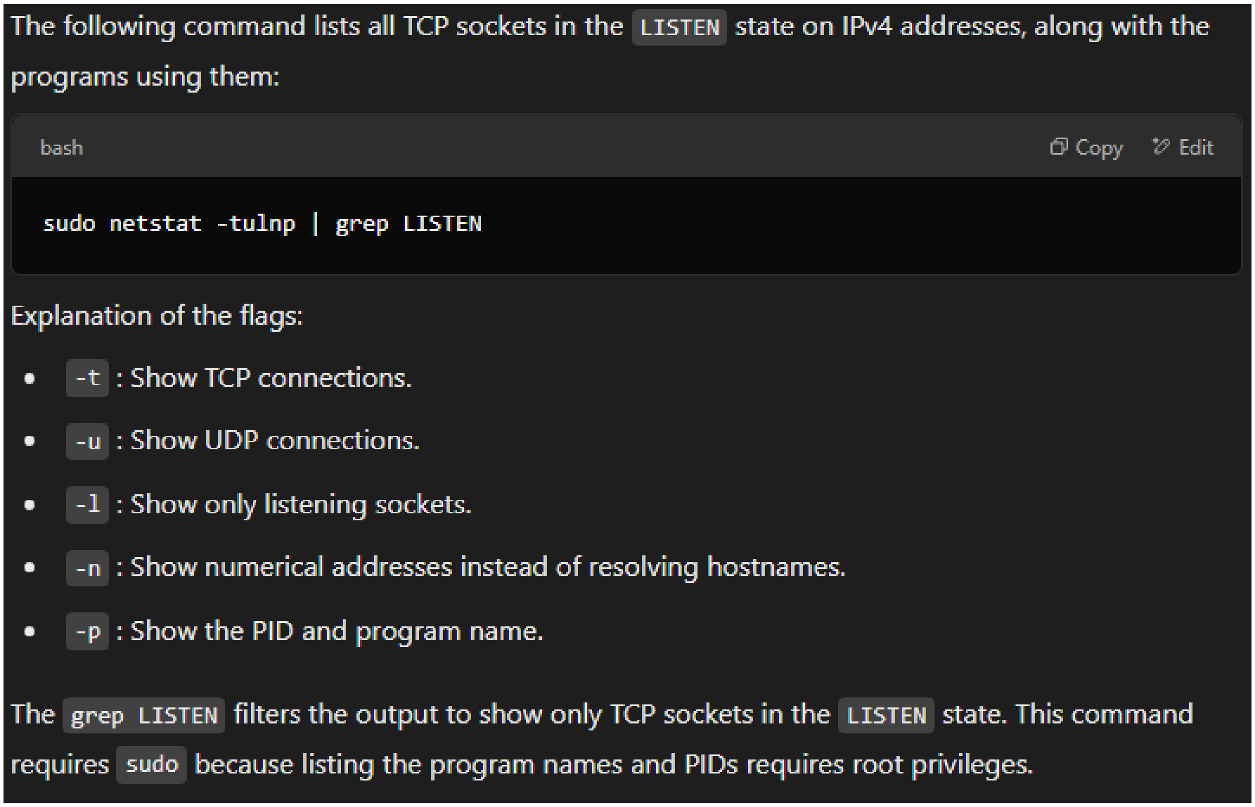
Erick Franco
1/18/25
CS430p

Lab 2 part 1

TCP 2.1.1

- Take a screenshot of the prompt and the command that ChatGPT generates



The following command lists all TCP sockets in the `LISTEN` state on IPv4 addresses, along with the programs using them:

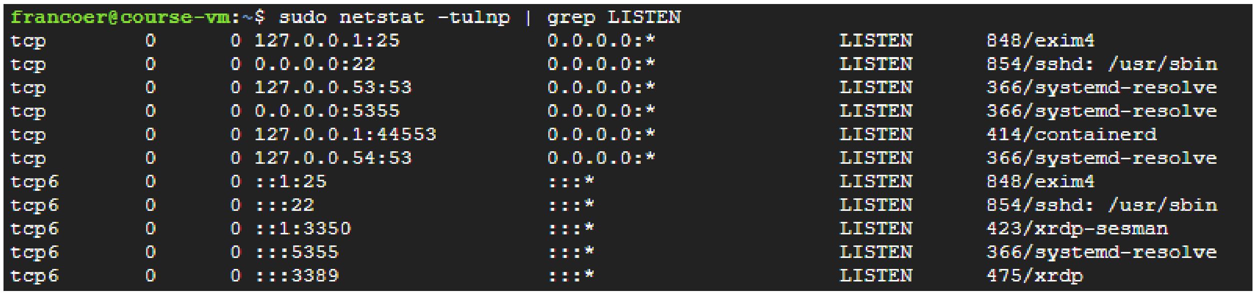
```bash
sudo netstat -tulnp | grep LISTEN
```

Explanation of the flags:

- `-t` : Show TCP connections.
- `-u` : Show UDP connections.
- `-l` : Show only listening sockets.
- `-n` : Show numerical addresses instead of resolving hostnames.
- `-p` : Show the PID and program name.

The `grep LISTEN` filters the output to show only TCP sockets in the `LISTEN` state. This command requires `sudo` because listing the program names and PIDs requires root privileges.

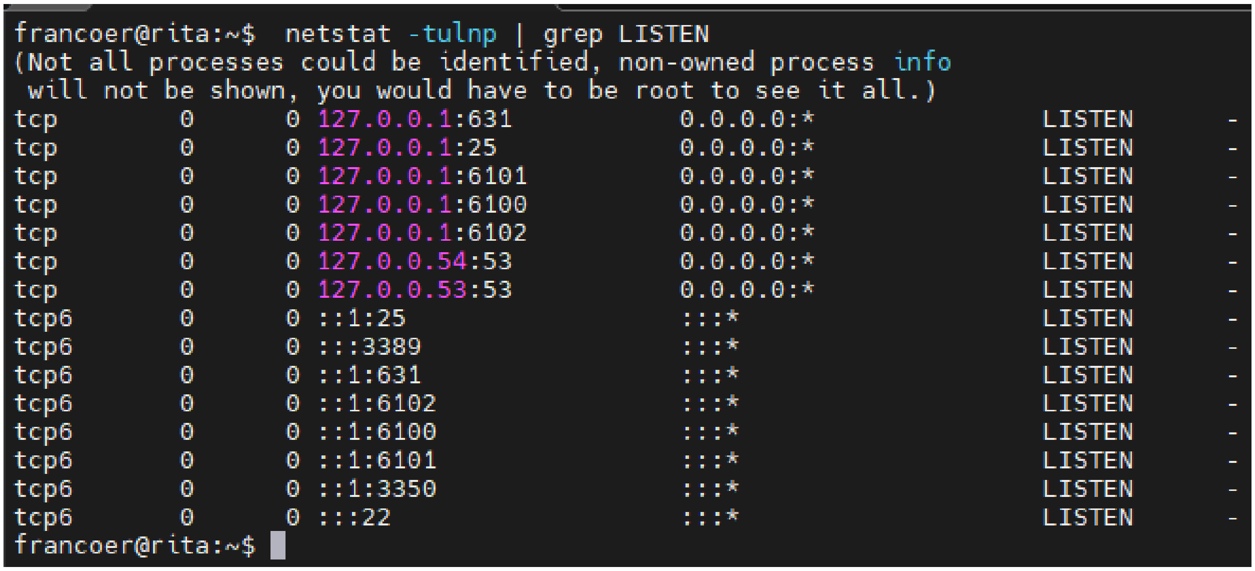
- **Run the command using `sudo` and take a screenshot of the output to include in your lab notebook.**



```
francoer@course-vm:~$ sudo netstat -tulnp | grep LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      848/exim4
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      854/sshd: /usr/sbin
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      366/systemd-resolve
tcp        0      0 0.0.0.0:5355            0.0.0.0:*               LISTEN      366/systemd-resolve
tcp        0      0 127.0.0.1:44553         0.0.0.0:*               LISTEN      414/containerd
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      366/systemd-resolve
tcp6       0      0 ::1:25                  :::*                    LISTEN      848/exim4
tcp6       0      0 :::22                   :::*                    LISTEN      854/sshd: /usr/sbin
tcp6       0      0 ::1:3350                :::*                    LISTEN      423/xrdp-sesman
tcp6       0      0 :::5355                 :::*                    LISTEN      366/systemd-resolve
tcp6       0      0 :::3389                 :::*                    LISTEN      475/xrdp
```

List a service that can be contacted from any interface on the machine. List a service that can only be contacted by local processes.
Any interface: port 854 on 0.0.0.0:22
Local interface: port 848 on 127.0.0.1:25

Run the command again, but do not use sudo as this is a machine managed by CAT. Include a screenshot of the output.

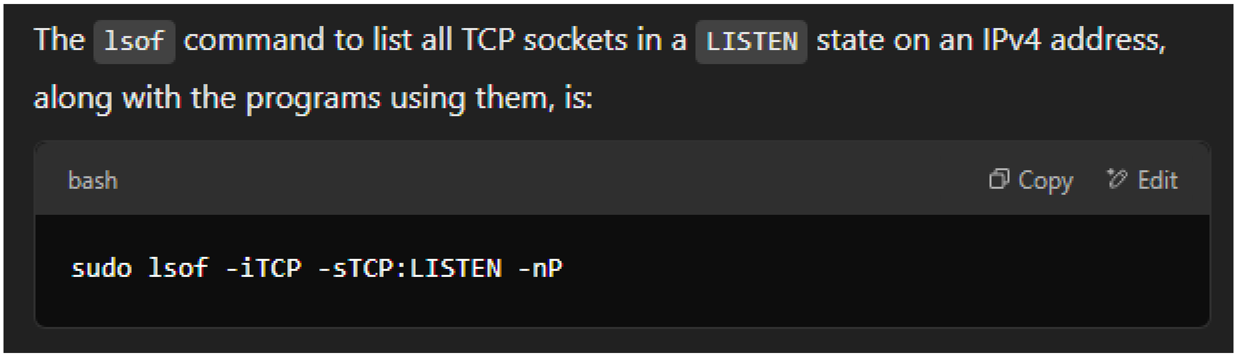


```
francoer@rita:~$  netstat -tulnp | grep LISTEN
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6101          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6100          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6102          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp6       0      0 ::1:25                  :::*                    LISTEN      -
tcp6       0      0 :::3389                 :::*                    LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 ::1:6102                :::*                    LISTEN      -
tcp6       0      0 ::1:6100                :::*                    LISTEN      -
tcp6       0      0 ::1:6101                :::*                    LISTEN      -
tcp6       0      0 ::1:3350                :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
francoer@rita:~$
```

List the services that this machine provides for external access
-:::22 :::* LISTEN -
-:::3389 :::* LISTEN -

TCP 2.1.2

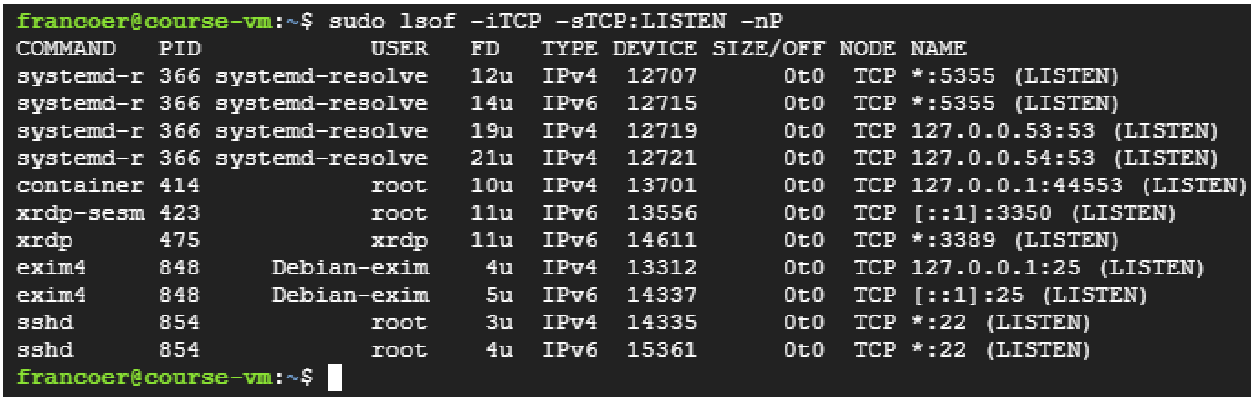
- Take a screenshot of the prompt and the command that ChatGPT generates



The `lsof` command to list all TCP sockets in a `LISTEN` state on an IPv4 address, along with the programs using them, is:

```bash
sudo lsof -iTCP -sTCP:LISTEN -nP
```

Run the command using sudo and take a screenshot of the output to include in your lab notebook.

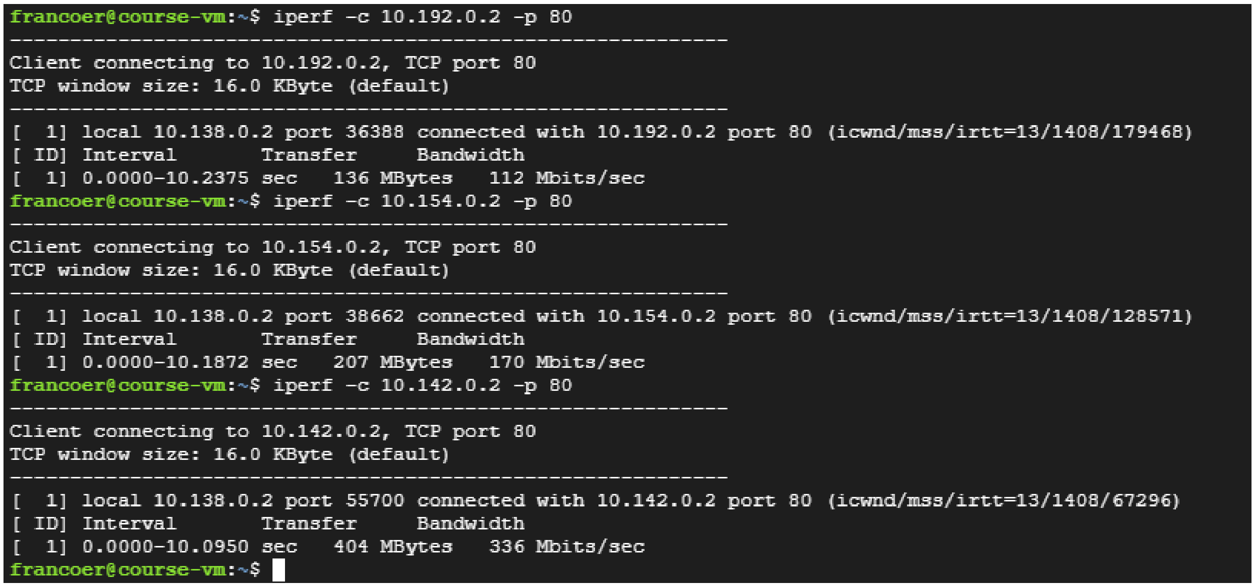


```
francoer@course-vm:~$ sudo lsof -iTCP -sTCP:LISTEN -nP
COMMAND   PID             USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd-r 366 systemd-resolve   12u  IPv4  12707      0t0  TCP *:5355 (LISTEN)
systemd-r 366 systemd-resolve   14u  IPv6  12715      0t0  TCP *:5355 (LISTEN)
systemd-r 366 systemd-resolve   19u  IPv4  12719      0t0  TCP 127.0.0.53:53 (LISTEN)
systemd-r 366 systemd-resolve   21u  IPv4  12721      0t0  TCP 127.0.0.54:53 (LISTEN)
container 414            root   10u  IPv4  13701      0t0  TCP 127.0.0.1:44553 (LISTEN)
xrdp-sesm 423            root   11u  IPv6  13556      0t0  TCP [::1]:3350 (LISTEN)
xrdp      475            xrdp   11u  IPv6  14611      0t0  TCP *:3389 (LISTEN)
exim4     848     Debian-exim    4u  IPv4  13312      0t0  TCP 127.0.0.1:25 (LISTEN)
exim4     848     Debian-exim    5u  IPv6  14337      0t0  TCP [::1]:25 (LISTEN)
sshd      854            root    3u  IPv4  14335      0t0  TCP *:22 (LISTEN)
sshd      854            root    4u  IPv6  15361      0t0  TCP *:22 (LISTEN)
francoer@course-vm:~$
```

TCP 2.1.4

Show a screenshot of the measured bandwidth available between your us-west1-b VM and each of the other Compute Engine VMs. Explain the relative differences (or lack thereof) in your results.

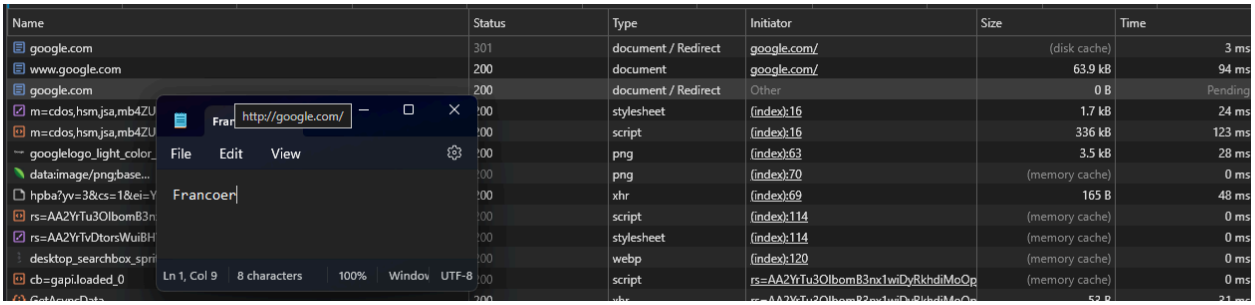
I think the difference would be distance as the closer you are to us-west the faster/or bandwidth there is because east has the highest bandwidth, than Europe than australia



```
francoer@course-vm:~$ iperf -c 10.192.0.2 -p 80
------------------------------------------------------------
Client connecting to 10.192.0.2, TCP port 80
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  1] local 10.138.0.2 port 36388 connected with 10.192.0.2 port 80 (icwnd/mss/irtt=13/1408/179468)
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.2375 sec   136 MBytes   112 Mbits/sec
francoer@course-vm:~$ iperf -c 10.154.0.2 -p 80
------------------------------------------------------------
Client connecting to 10.154.0.2, TCP port 80
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  1] local 10.138.0.2 port 38662 connected with 10.154.0.2 port 80 (icwnd/mss/irtt=13/1408/128571)
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.1872 sec   207 MBytes   170 Mbits/sec
francoer@course-vm:~$ iperf -c 10.142.0.2 -p 80
------------------------------------------------------------
Client connecting to 10.142.0.2, TCP port 80
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  1] local 10.138.0.2 port 55700 connected with 10.142.0.2 port 80 (icwnd/mss/irtt=13/1408/67296)
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0950 sec   404 MBytes   336 Mbits/sec
francoer@course-vm:~$
```

TCP 2.1.5

Take a screenshot of the initial requests for your lab notebook.

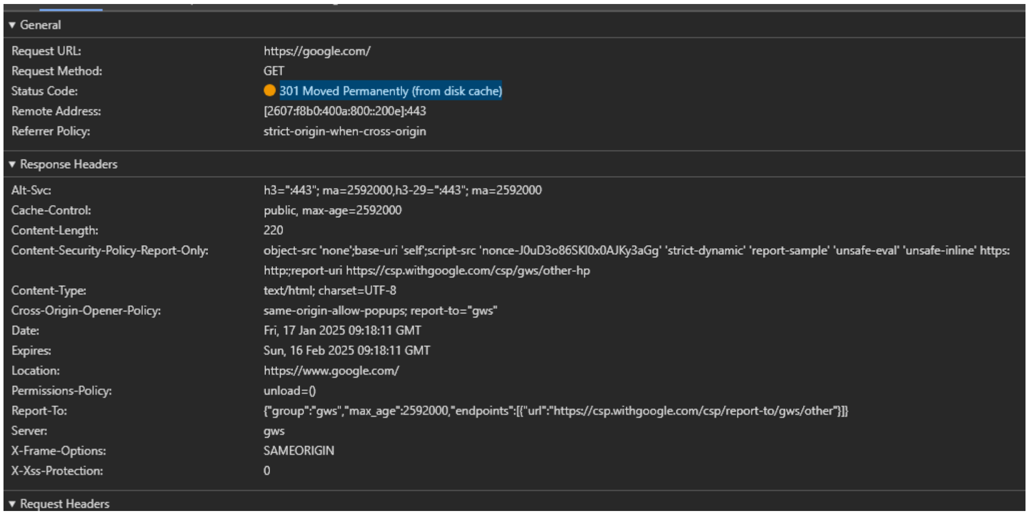


Name	Status	Type	Initiator	Size	Time
google.com	301	document / Redirect	google.com/	(disk cache)	3 ms
www.google.com	200	document	google.com/	63.9 kB	94 ms
google.com	200	document / Redirect	Other	0 B	Pending
m=cdos,hsm,jsa,mb4ZU	200	stylesheet	(index):16	1.7 kB	24 ms
m=cdos,hsm,jsa,mb4ZU	200	script	(index):16	336 kB	123 ms
googlelogo_light_color_	200	png	(index):63	3.5 kB	28 ms
data:image/png;base...	200	png	(index):70	(memory cache)	0 ms
hpba?yv=3&cs=1&ei=Y	200	xhr	(index):69	165 B	48 ms
rs=AA2YrTu3OlbomB3n	200	script	(index):114	(memory cache)	0 ms
rs=AA2YrTvDtorsWuiBH	200	stylesheet	(index):114	(memory cache)	0 ms
desktop_searchbox_spri	200	webp	(index):120	(memory cache)	0 ms
cb=gapi.loaded_0	200	script	rs=AA2YrTu3OlbomB3nx1wiDyRkhdiMoOp	(memory cache)	0 ms
GetAsyncData	200	xhr	rs=AA2YrTu3OlbomB3nx1wiDyRkhdiMoOp	53 B	31 ms

Francoer

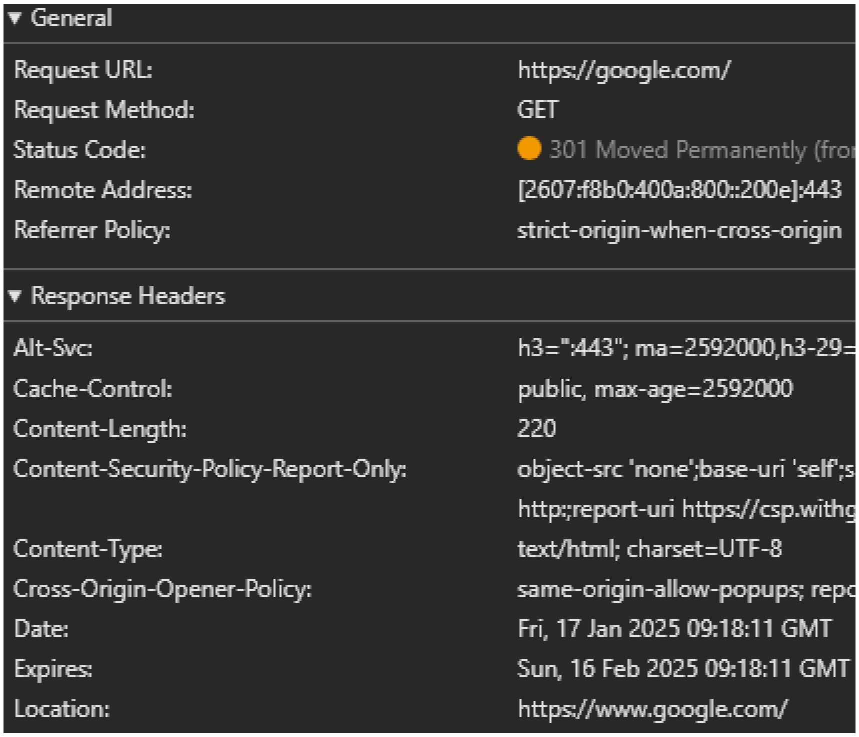
Ln 1, Col 9 | 8 characters | 100% | Windov | UTF-8

- What is the URL being requested?
    - https://google.com
- Explain the HTTP status code that is returned and what the code indicates
    - The status code means that the requested resource has been moved permanently to a new url, also meaning that the request was not to a server but rather to the local cache
- Take a screenshot indicating the version of the HTTP protocol that is used for each request. (Hint: look at the response status line and alt-svc: HTTP response headers indicating HTTP/2 or HTTP/3).



```
▾ General
Request URL:                        https://google.com/
Request Method:                     GET
Status Code:                        ● 301 Moved Permanently (from disk cache)
Remote Address:                     [2607:f8b0:400a:800::200e]:443
Referrer Policy:                    strict-origin-when-cross-origin
▾ Response Headers
Alt-Svc:                            h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
Cache-Control:                      public, max-age=2592000
Content-Length:                     220
Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-src 'nonce-J0uD3o86SKI0x0AJKy3aGg' 'strict-dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inline' https:
                                    http:;report-uri https://csp.withgoogle.com/csp/gws/other-hp
Content-Type:                       text/html; charset=UTF-8
Cross-Origin-Opener-Policy:         same-origin-allow-popups; report-to="gws"
Date:                               Fri, 17 Jan 2025 09:18:11 GMT
Expires:                            Sun, 16 Feb 2025 09:18:11 GMT
Location:                           https://www.google.com/
Permissions-Policy:                 unload=()
Report-To:                          {"group":"gws","max_age":2592000,"endpoints":[{"url":"https://csp.withgoogle.com/csp/report-to/gws/other"}]}
Server:                             gws
X-Frame-Options:                    SAMEORIGIN
X-Xss-Protection:                   0
▾ Request Headers
```

- Show the URLs the browser is redirected to via this header.



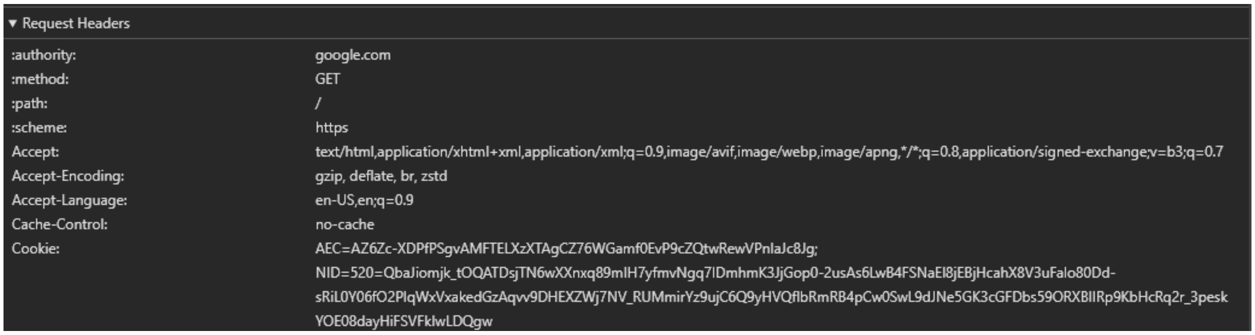
▼ General

Request URL:	https://google.com/
Request Method:	GET
Status Code:	● 301 Moved Permanently (fro
Remote Address:	[2607:f8b0:400a:800::200e]:443
Referrer Policy:	strict-origin-when-cross-origin

▼ Response Headers

Alt-Svc:	h3=":443"; ma=2592000,h3-29=
Cache-Control:	public, max-age=2592000
Content-Length:	220
Content-Security-Policy-Report-Only:	object-src 'none';base-uri 'self';s http:;report-uri https://csp.withg
Content-Type:	text/html; charset=UTF-8
Cross-Origin-Opener-Policy:	same-origin-allow-popups; repo
Date:	Fri, 17 Jan 2025 09:18:11 GMT
Expires:	Sun, 16 Feb 2025 09:18:11 GMT
Location:	https://www.google.com/

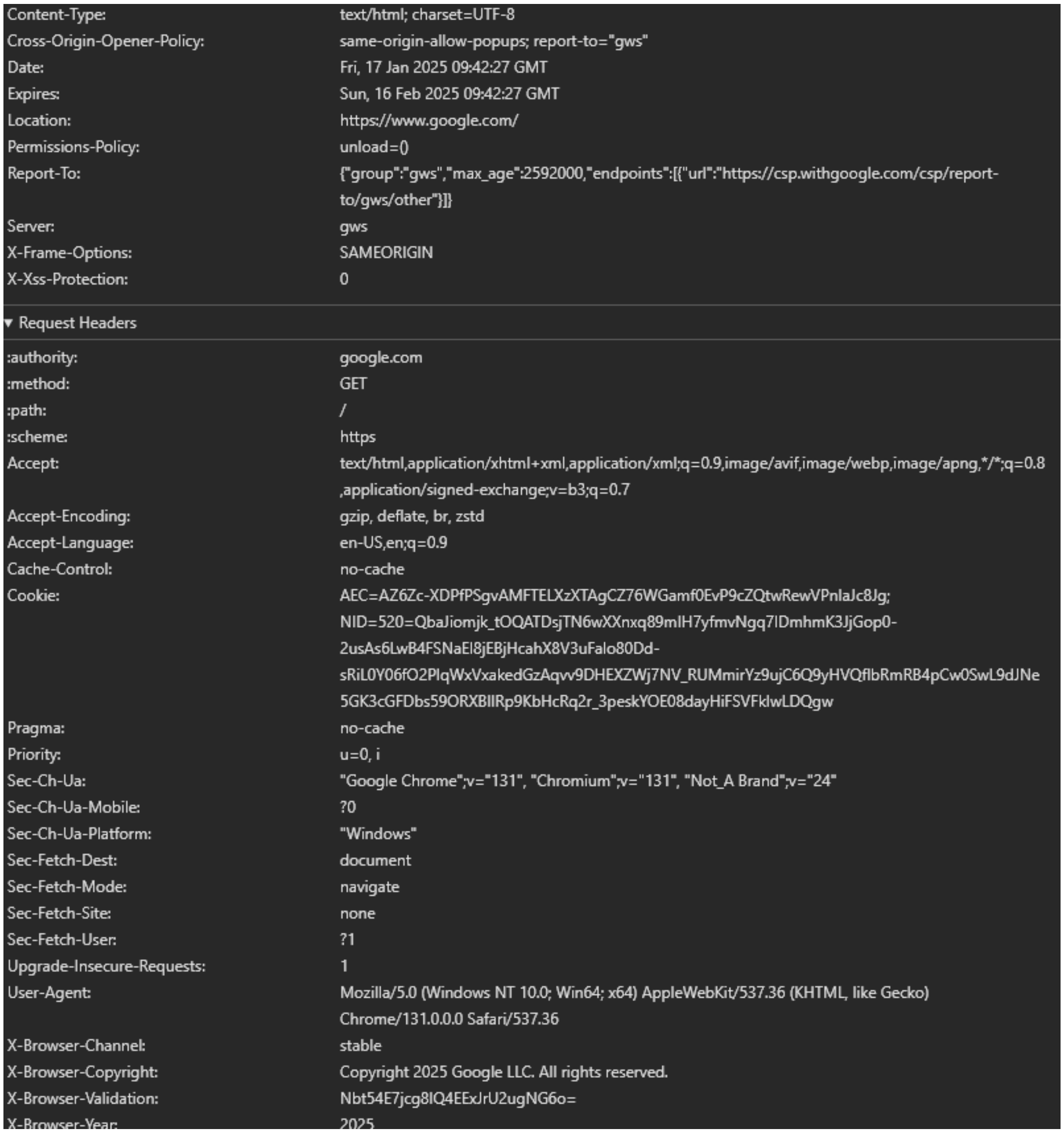
- Take a screenshot of when cookies are set via Set-Cookie:
- Take a screenshot of when cookies are attached via Cookie



▾ Request Headers

:authority:	google.com
:method:	GET
:path:	/
:scheme:	https
Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding:	gzip, deflate, br, zstd
Accept-Language:	en-US,en;q=0.9
Cache-Control:	no-cache

Cookie:	AEC=AZ6Zc-XDPfPSgvAMFTELXzXTAgCZ76WGamf0EvP9cZQtwRewVPnlaJc8Jg;
	NID=520=QbaJiomjk_tOQATDsjTN6wXXnxq89mlH7yfmvNgq7IDmhmK3JjGop0-2usAs6LwB4FSNaEl8jEBjHcahX8V3uFalo80Dd-sRiL0Y06fO2PlqWxVxakedGzAqvv9DHEXZWj7NV_RUMmirYz9ujC6Q9yHVQflbRmRB4pCw0SwL9dJNe5GK3cGFDbs59ORXBIIRp9KbHcRq2r_3peskYOE08dayHiFSVFklwLDQgw



Content-Type:	text/html; charset=UTF-8
Cross-Origin-Opener-Policy:	same-origin-allow-popups; report-to="gws"
Date:	Fri, 17 Jan 2025 09:42:27 GMT
Expires:	Sun, 16 Feb 2025 09:42:27 GMT
Location:	https://www.google.com/
Permissions-Policy:	unload=()
Report-To:	{"group":"gws","max_age":2592000,"endpoints":[{"url":"https://csp.withgoogle.com/csp/report-to/gws/other"}]}
Server:	gws
X-Frame-Options:	SAMEORIGIN
X-Xss-Protection:	0

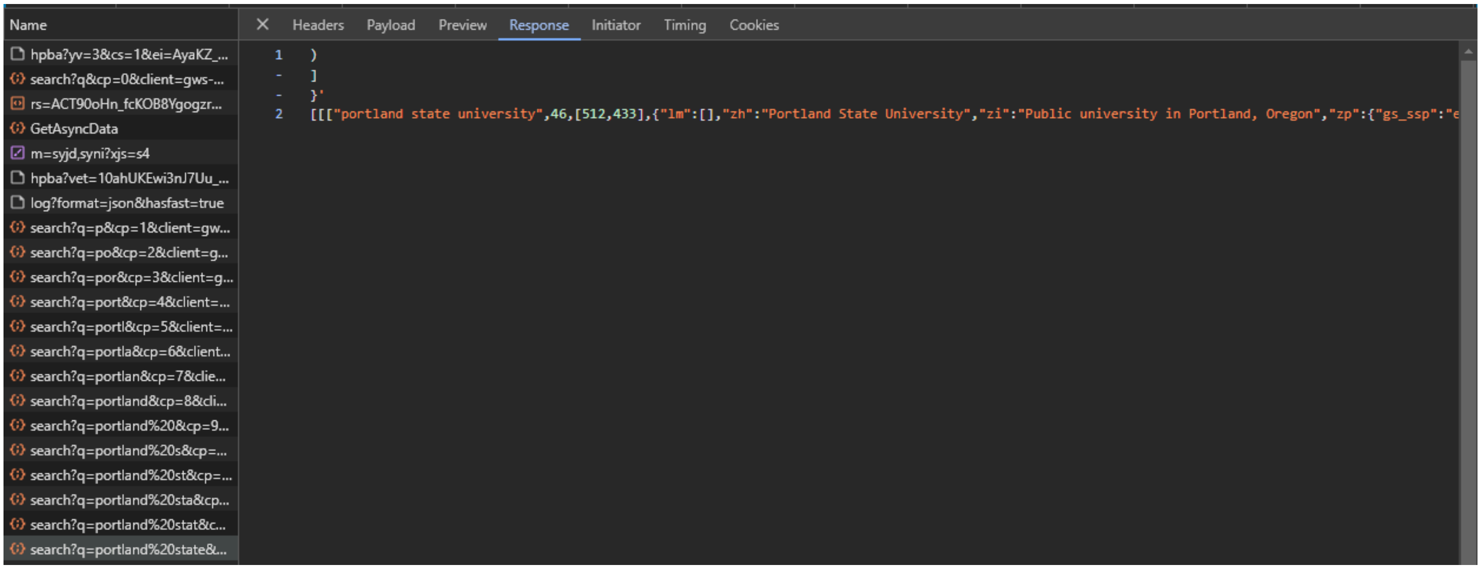
▾ Request Headers

:authority:	google.com
:method:	GET
:path:	/
:scheme:	https
Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding:	gzip, deflate, br, zstd
Accept-Language:	en-US,en;q=0.9
Cache-Control:	no-cache

Cookie:	AEC=AZ6Zc-XDPfPSgvAMFTELXzXTAgCZ76WGamf0EvP9cZQtwRewVPnlaJc8Jg;
	NID=520=QbaJiomjk_tOQATDsjTN6wXXnxq89mlH7yfmvNgq7IDmhmK3JjGop0-2usAs6LwB4FSNaEl8jEBjHcahX8V3uFalo80Dd-sRiL0Y06fO2PlqWxVxakedGzAqvv9DHEXZWj7NV_RUMmirYz9ujC6Q9yHVQflbRmRB4pCw0SwL9dJNe5GK3cGFDbs59ORXBIIRp9KbHcRq2r_3peskYOE08dayHiFSVFklwLDQgw

Pragma:	no-cache
Priority:	u=0, i
Sec-Ch-Ua:	"Google Chrome";v="131", "Chromium";v="131", "Not_A Brand";v="24"
Sec-Ch-Ua-Mobile:	?0
Sec-Ch-Ua-Platform:	"Windows"
Sec-Fetch-Dest:	document
Sec-Fetch-Mode:	navigate
Sec-Fetch-Site:	none
Sec-Fetch-User:	?1
Upgrade-Insecure-Requests:	1
User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
X-Browser-Channel:	stable
X-Browser-Copyright:	Copyright 2025 Google LLC. All rights reserved.
X-Browser-Validation:	Nbt54E7jcg8IQ4EExJrU2ugNG6o=
X-Browser-Year:	2025

- Show the requests and responses in the listing. Click on the last request sent, then click on the response to see that its payload has returned the data that is then rendered on the search page similar to what is shown below for "rabbid"



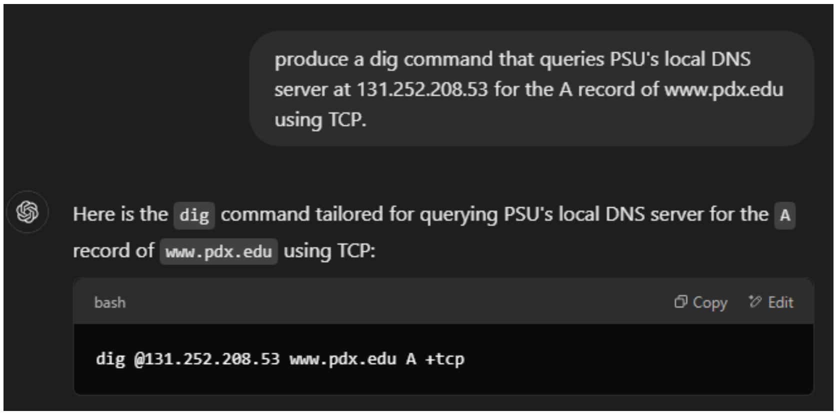
```
hpba?yv=3&cs=1&ei=AyaKZ_...
search?q&cp=0&client=gws-...
rs=ACT90oHn_fcKOB8Ygogzr...
GetAsyncData
m=syjd,syni?xjs=s4
hpba?vet=10ahUKEwi3nJ7Uu_...
log?format=json&hasfast=true
search?q=p&cp=1&client=gw...
search?q=po&cp=2&client=g...
search?q=por&cp=3&client=g...
search?q=port&cp=4&client=...
search?q=portl&cp=5&client=...
search?q=portla&cp=6&client...
search?q=portlan&cp=7&clie...
search?q=portland&cp=8&cli...
search?q=portland%20&cp=9...
search?q=portland%20s&cp=...
search?q=portland%20st&cp=...
search?q=portland%20sta&cp...
search?q=portland%20stat&c...
search?q=portland%20state&...
```

```
)
]
}'
[[["portland state university",46,[512,433],{"lm":[],"zh":"Portland State University","zi":"Public university in Portland, Oregon","zp":{"gs_ssp":"e
```

Lab 2 part 2

DNS 2.2.1

- Take a screenshot of the prompt and the dig command produced.

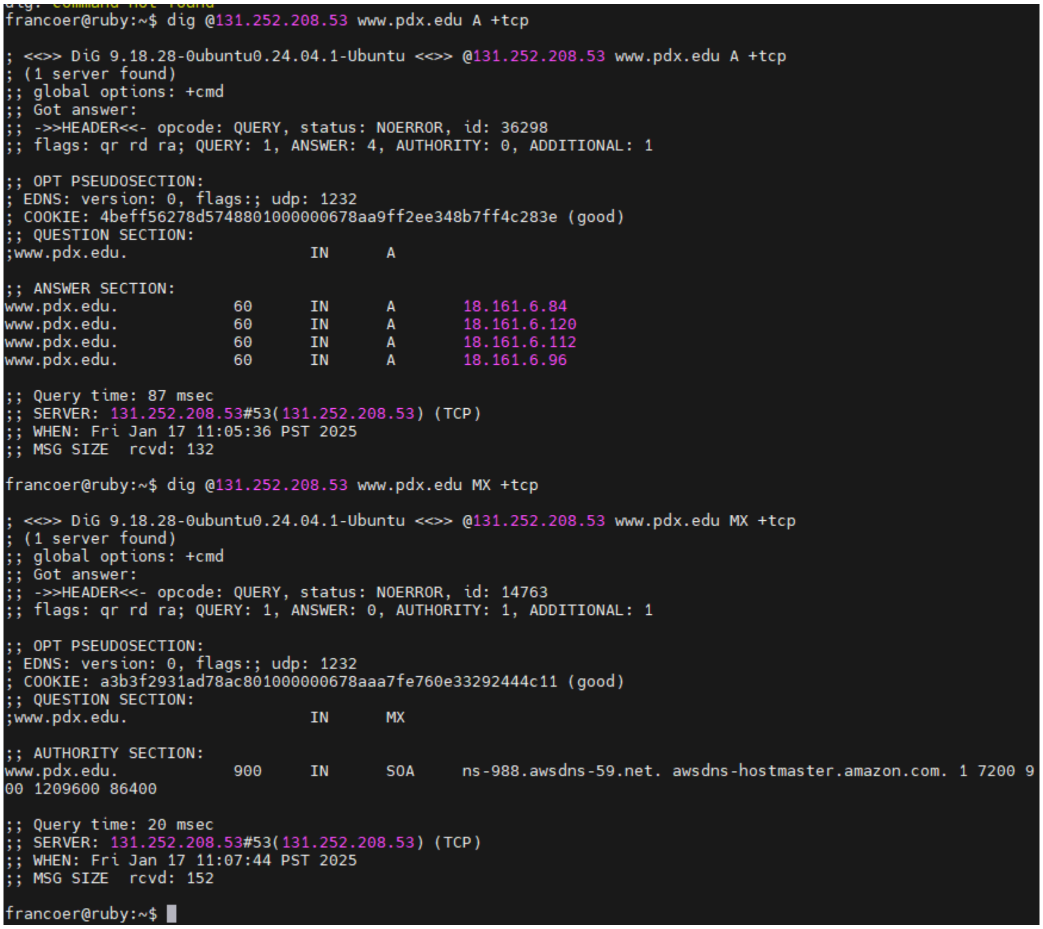


produce a dig command that queries PSU's local DNS server at 131.252.208.53 for the A record of www.pdx.edu using TCP.

Here is the `dig` command tailored for querying PSU's local DNS server for the `A` record of `www.pdx.edu` using TCP:

```bash
dig @131.252.208.53 www.pdx.edu A +tcp
```

Take a screenshot of the records returned for your lab notebook.



```
dig: command not found
francoer@ruby:~$ dig @131.252.208.53 www.pdx.edu A +tcp

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @131.252.208.53 www.pdx.edu A +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36298
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 4beff56278d5748801000000678aa9ff2ee348b7ff4c283e (good)
;; QUESTION SECTION:
;www.pdx.edu.                   IN      A

;; ANSWER SECTION:
www.pdx.edu.            60      IN      A       18.161.6.84
www.pdx.edu.            60      IN      A       18.161.6.120
www.pdx.edu.            60      IN      A       18.161.6.112
www.pdx.edu.            60      IN      A       18.161.6.96

;; Query time: 87 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Fri Jan 17 11:05:36 PST 2025
;; MSG SIZE  rcvd: 132

francoer@ruby:~$ dig @131.252.208.53 www.pdx.edu MX +tcp

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @131.252.208.53 www.pdx.edu MX +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14763
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: a3b3f2931ad78ac801000000678aaa7fe760e33292444c11 (good)
;; QUESTION SECTION:
;www.pdx.edu.                   IN      MX

;; AUTHORITY SECTION:
www.pdx.edu.            900     IN      SOA     ns-988.awsdns-59.net. awsdns-hostmaster.amazon.com. 1 7200 9
00 1209600 86400

;; Query time: 20 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Fri Jan 17 11:07:44 PST 2025
;; MSG SIZE  rcvd: 152

francoer@ruby:~$
```

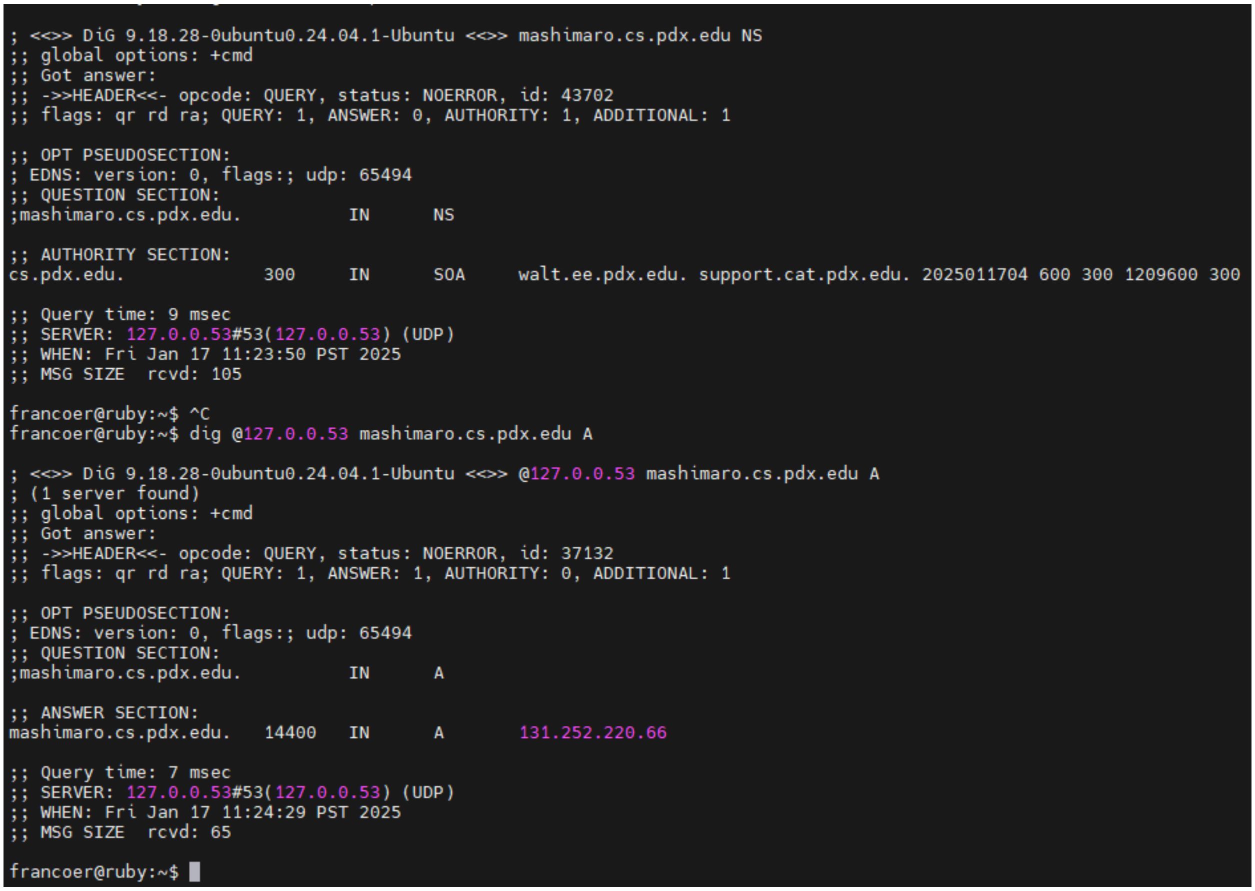
What cloud provider hosts the web site for www.pdx.edu?

From these below I assume that it is locally hosted but I put these in the **iplocation and got Amazon**

```
www.pdx.edu.        60    IN    A     18.161.6.84
www.pdx.edu.        60    IN    A     18.161.6.120
www.pdx.edu.        60    IN    A     18.161.6.112
www.pdx.edu.        60    IN    A     18.161.6.96
```

What cloud provider handles mail for pdx.edu?
awsdns-hostmaster.amazon.com.

Take a screenshot of the results for both records for your lab notebook.

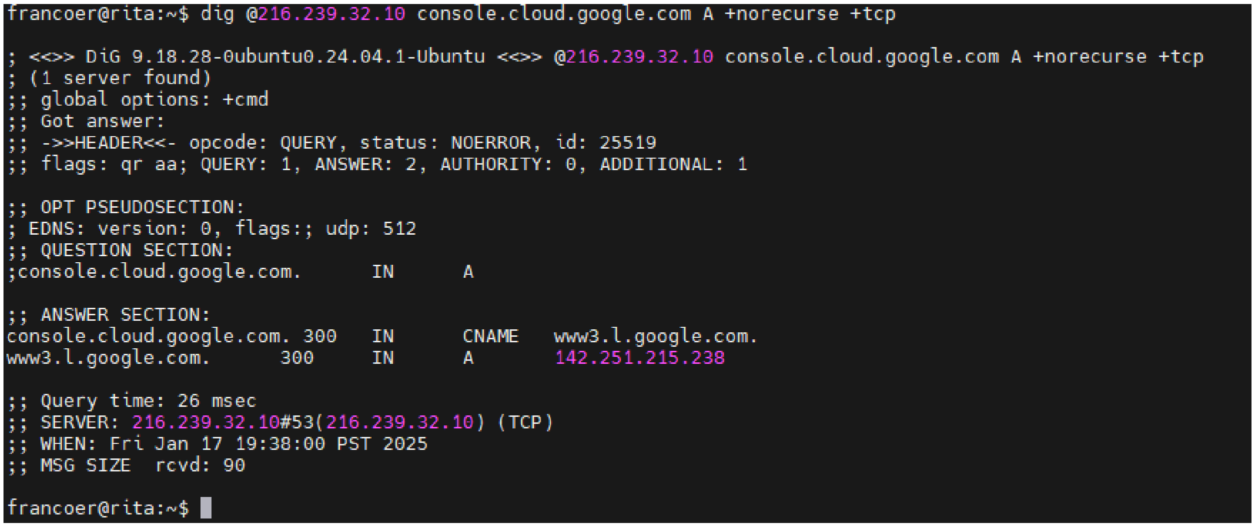


```
; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> mashimaro.cs.pdx.edu NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43702
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      NS

;; AUTHORITY SECTION:
cs.pdx.edu.             300     IN      SOA     walt.ee.pdx.edu. support.cat.pdx.edu. 2025011704 600 300 1209600 300

;; Query time: 9 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Fri Jan 17 11:23:50 PST 2025
;; MSG SIZE  rcvd: 105

francoer@ruby:~$ ^C
francoer@ruby:~$ dig @127.0.0.53 mashimaro.cs.pdx.edu A

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @127.0.0.53 mashimaro.cs.pdx.edu A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37132
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      A

;; ANSWER SECTION:
mashimaro.cs.pdx.edu.   14400   IN      A       131.252.220.66

;; Query time: 7 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Fri Jan 17 11:24:29 PST 2025
;; MSG SIZE  rcvd: 65

francoer@ruby:~$
```

DNS 2.2.2

List all of the iterative dig commands performed for the lookup
dig
dig @192.5.5.241 google.com NS +norecurse +tcp
dig @192.5.6.30 google.com NS +norecurse +tcp
dig @216.239.32.10 console.cloud.google.com A +norecurse +tcp

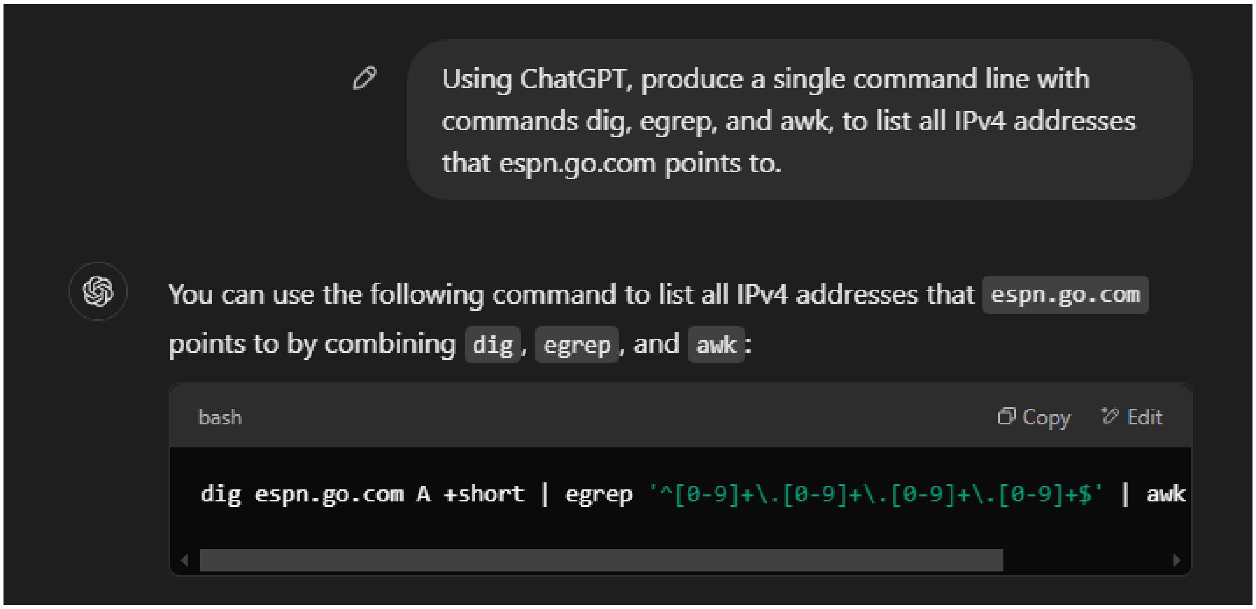
Take a screenshot of the results of the final query for your lab notebook.



```
francoer@rita:~$ dig @216.239.32.10 console.cloud.google.com A +norecurse +tcp

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @216.239.32.10 console.cloud.google.com A +norecurse +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25519
;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;console.cloud.google.com.      IN      A

;; ANSWER SECTION:
console.cloud.google.com. 300   IN      CNAME   www3.l.google.com.
www3.l.google.com.      300     IN      A       142.251.215.238

;; Query time: 26 msec
;; SERVER: 216.239.32.10#53(216.239.32.10) (TCP)
;; WHEN: Fri Jan 17 19:38:00 PST 2025
;; MSG SIZE  rcvd: 90

francoer@rita:~$
```

DNS 2.2.3

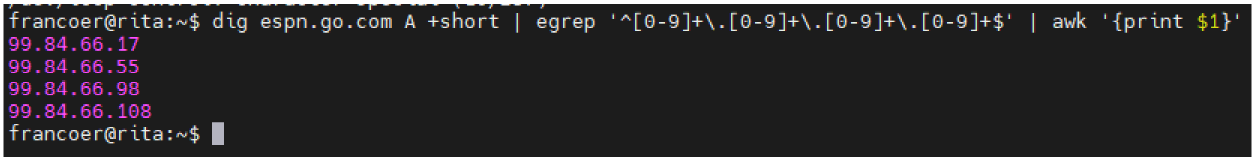
Using ChatGPT, produce a single command line with commands dig, egrep, and awk, to list all IPv4 addresses that espn.go.com points to.



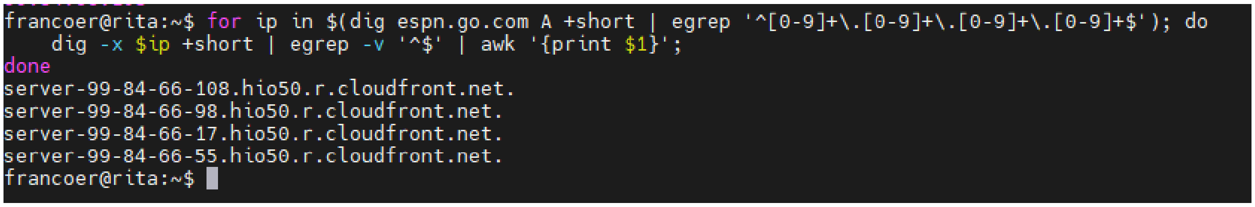
Using ChatGPT, produce a single command line with commands dig, egrep, and awk, to list all IPv4 addresses that espn.go.com points to.

You can use the following command to list all IPv4 addresses that `espn.go.com` points to by combining `dig`, `egrep`, and `awk`:

```bash
dig espn.go.com A +short | egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$' | awk
```



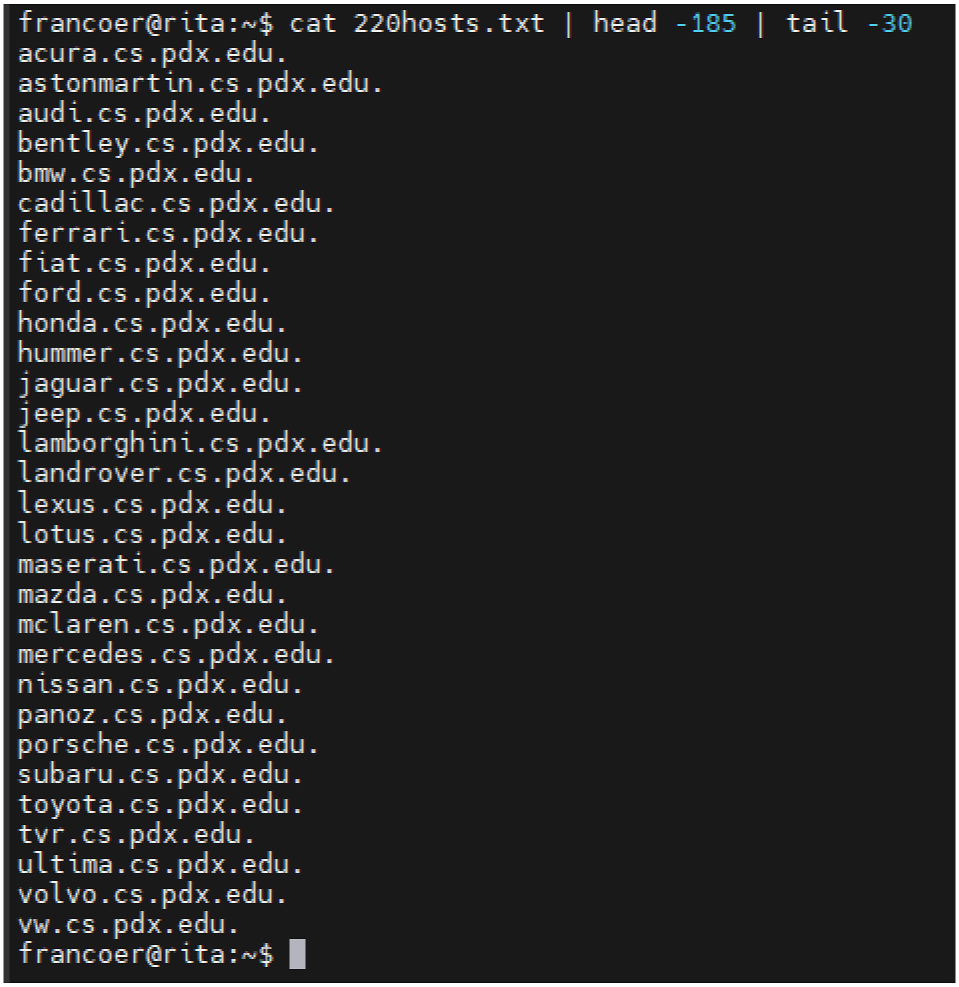
```
francoer@rita:~$ dig espn.go.com A +short | egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$' | awk '{print $1}'
99.84.66.17
99.84.66.55
99.84.66.98
99.84.66.108
francoer@rita:~$ 
```



```
francoer@rita:~$ for ip in $(dig espn.go.com A +short | egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$'); do
    dig -x $ip +short | egrep -v '^$' | awk '{print $1}';
done
server-99-84-66-108.hio50.r.cloudfront.net.
server-99-84-66-98.hio50.r.cloudfront.net.
server-99-84-66-17.hio50.r.cloudfront.net.
server-99-84-66-55.hio50.r.cloudfront.net.
francoer@rita:~$ 
```

DNS 2.2.4

156-185



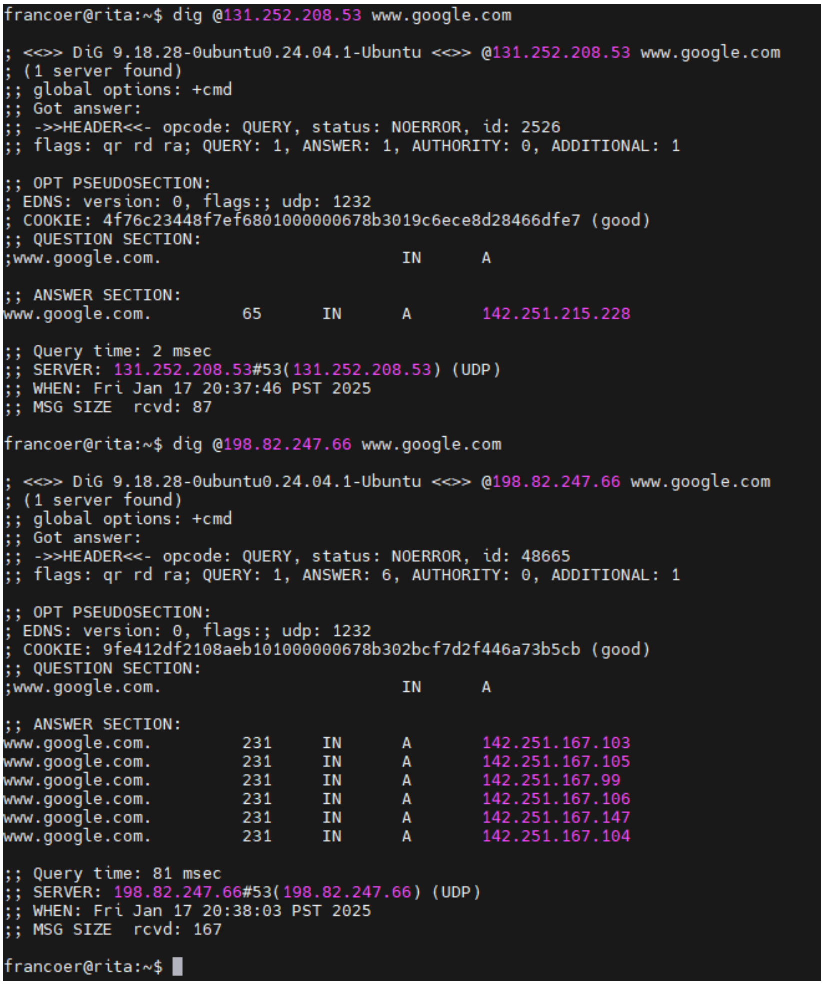
```
francoer@rita:~$ cat 220hosts.txt | head -185 | tail -30
acura.cs.pdx.edu.
astonmartin.cs.pdx.edu.
audi.cs.pdx.edu.
bentley.cs.pdx.edu.
bmw.cs.pdx.edu.
cadillac.cs.pdx.edu.
ferrari.cs.pdx.edu.
fiat.cs.pdx.edu.
ford.cs.pdx.edu.
honda.cs.pdx.edu.
hummer.cs.pdx.edu.
jaguar.cs.pdx.edu.
jeep.cs.pdx.edu.
lamborghini.cs.pdx.edu.
landrover.cs.pdx.edu.
lexus.cs.pdx.edu.
lotus.cs.pdx.edu.
maserati.cs.pdx.edu.
mazda.cs.pdx.edu.
mclaren.cs.pdx.edu.
mercedes.cs.pdx.edu.
nissan.cs.pdx.edu.
panoz.cs.pdx.edu.
porsche.cs.pdx.edu.
subaru.cs.pdx.edu.
toyota.cs.pdx.edu.
tvr.cs.pdx.edu.
ultima.cs.pdx.edu.
volvo.cs.pdx.edu.
vw.cs.pdx.edu.
francoer@rita:~$
```

DNS 2.2.5

- What geographic locations do ipinfo.io and DB-IP return?

For the PSU IP I get the same location on both, Region: Oregon, and City Portland, although DB is a bit more accurate saying in North Portland and having a ISP.

While for the Virginia the cities are different, along with latitude and longitude.



```
francoer@rita:~$ dig @131.252.208.53 www.google.com

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @131.252.208.53 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2526
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 4f76c23448f7ef6801000000678b3019c6ece8d28466dfe7 (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         65      IN      A       142.251.215.228

;; Query time: 2 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (UDP)
;; WHEN: Fri Jan 17 20:37:46 PST 2025
;; MSG SIZE  rcvd: 87

francoer@rita:~$ dig @198.82.247.66 www.google.com

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @198.82.247.66 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48665
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 9fe412df2108aeb101000000678b302bcf7d2f446a73b5cb (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         231     IN      A       142.251.167.103
www.google.com.         231     IN      A       142.251.167.105
www.google.com.         231     IN      A       142.251.167.99
www.google.com.         231     IN      A       142.251.167.106
www.google.com.         231     IN      A       142.251.167.147
www.google.com.         231     IN      A       142.251.167.104

;; Query time: 81 msec
;; SERVER: 198.82.247.66#53(198.82.247.66) (UDP)
;; WHEN: Fri Jan 17 20:38:03 PST 2025
;; MSG SIZE  rcvd: 167

francoer@rita:~$ █
```

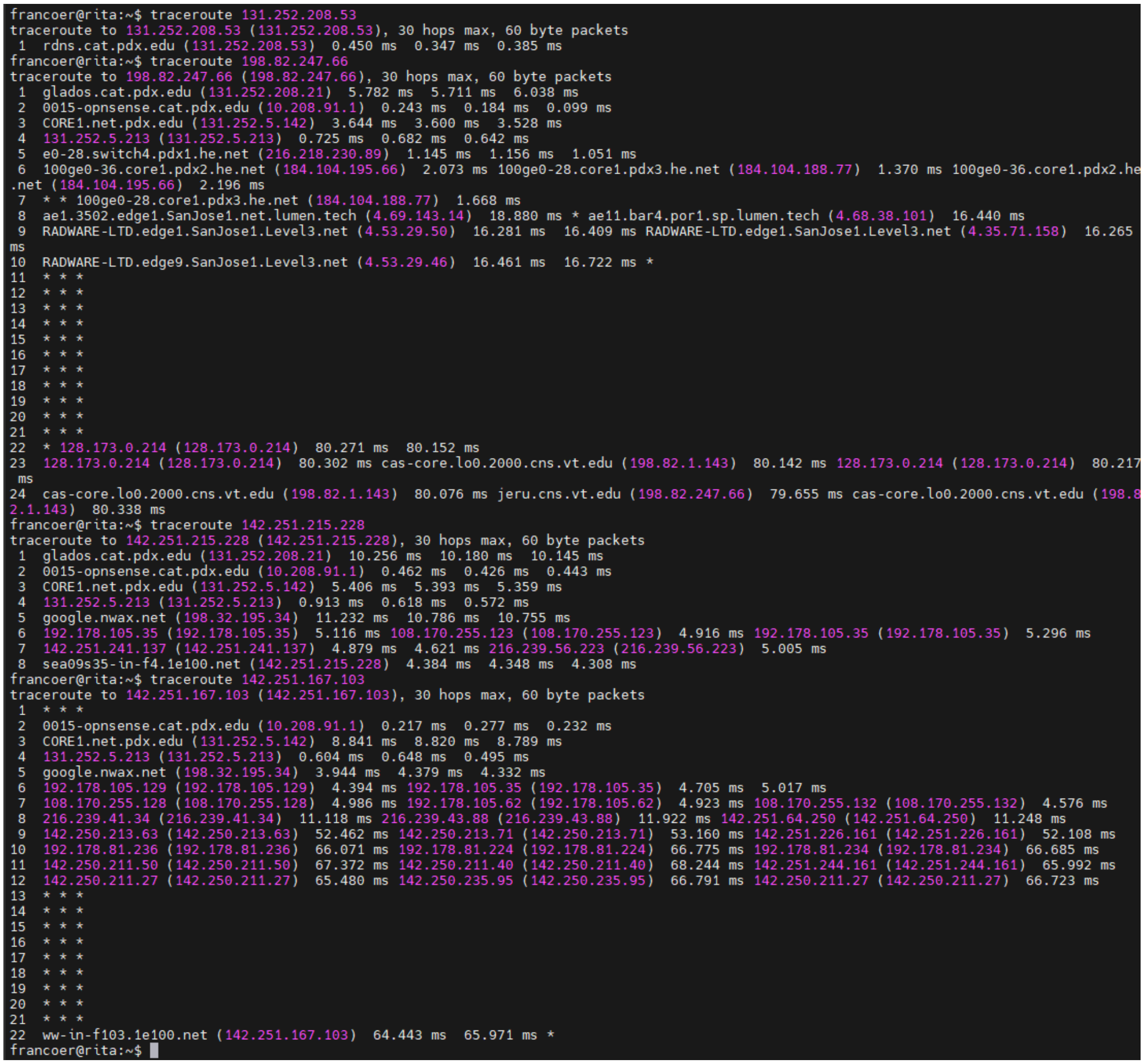
What are the geographic coordinates of each DNS server and the IP address it resolves for www.google.com?

PSU
Latitude:45.5234

Longitude:-122.6762
142.251.215.228

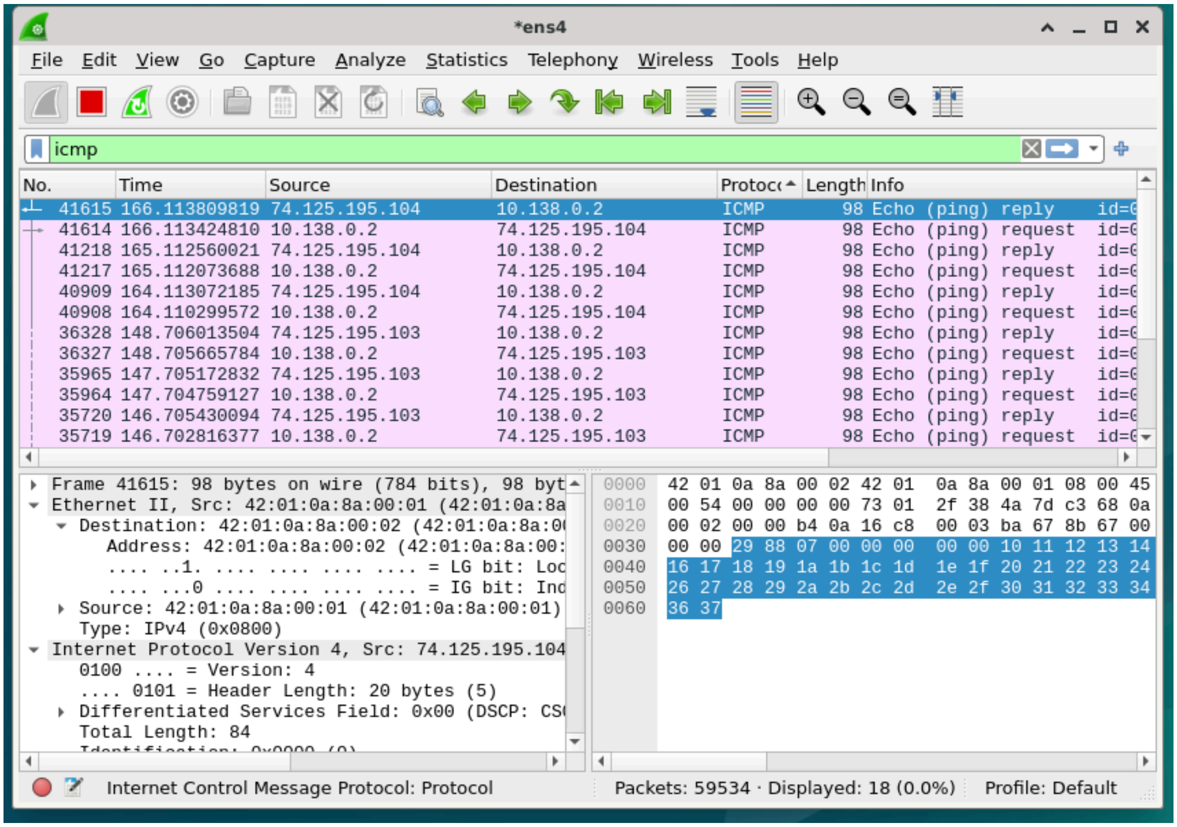
VPI and SU

Latitude:37.2296
Longitude:-80.4139
142.251.167.103



```
francoer@rita:~$ traceroute 131.252.208.53
traceroute to 131.252.208.53 (131.252.208.53), 30 hops max, 60 byte packets
 1  rdns.cat.pdx.edu (131.252.208.53)  0.450 ms  0.347 ms  0.385 ms
francoer@rita:~$ traceroute 198.82.247.66
traceroute to 198.82.247.66 (198.82.247.66), 30 hops max, 60 byte packets
 1  glados.cat.pdx.edu (131.252.208.21)  5.782 ms  5.711 ms  6.038 ms
 2  0015-opnsense.cat.pdx.edu (10.208.91.1)  0.243 ms  0.184 ms  0.099 ms
 3  CORE1.net.pdx.edu (131.252.5.142)  3.644 ms  3.600 ms  3.528 ms
 4  131.252.5.213 (131.252.5.213)  0.725 ms  0.682 ms  0.642 ms
 5  e0-28.switch4.pdx1.he.net (216.218.230.89)  1.145 ms  1.156 ms  1.051 ms
 6  100ge0-36.core1.pdx2.he.net (184.104.195.66)  2.073 ms 100ge0-28.core1.pdx3.he.net (184.104.188.77)  1.370 ms 100ge0-36.core1.pdx2.he
.net (184.104.195.66)  2.196 ms
 7  * * 100ge0-28.core1.pdx3.he.net (184.104.188.77)  1.668 ms
 8  ae1.3502.edge1.SanJose1.net.lumen.tech (4.69.143.14)  18.880 ms * ae11.bar4.por1.sp.lumen.tech (4.68.38.101)  16.440 ms
 9  RADWARE-LTD.edge1.SanJose1.Level3.net (4.53.29.50)  16.281 ms  16.409 ms RADWARE-LTD.edge1.SanJose1.Level3.net (4.35.71.158)  16.265
ms
10  RADWARE-LTD.edge9.SanJose1.Level3.net (4.53.29.46)  16.461 ms  16.722 ms *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * 128.173.0.214 (128.173.0.214)  80.271 ms  80.152 ms
23  128.173.0.214 (128.173.0.214)  80.302 ms cas-core.lo0.2000.cns.vt.edu (198.82.1.143)  80.142 ms 128.173.0.214 (128.173.0.214)  80.217
 ms
24  cas-core.lo0.2000.cns.vt.edu (198.82.1.143)  80.076 ms jeru.cns.vt.edu (198.82.247.66)  79.655 ms cas-core.lo0.2000.cns.vt.edu (198.8
2.1.143)  80.338 ms
francoer@rita:~$ traceroute 142.251.215.228
traceroute to 142.251.215.228 (142.251.215.228), 30 hops max, 60 byte packets
 1  glados.cat.pdx.edu (131.252.208.21)  10.256 ms  10.180 ms  10.145 ms
 2  0015-opnsense.cat.pdx.edu (10.208.91.1)  0.462 ms  0.426 ms  0.443 ms
 3  CORE1.net.pdx.edu (131.252.5.142)  5.406 ms  5.393 ms  5.359 ms
 4  131.252.5.213 (131.252.5.213)  0.913 ms  0.618 ms  0.572 ms
 5  google.nwax.net (198.32.195.34)  11.232 ms  10.786 ms  10.755 ms
 6  192.178.105.35 (192.178.105.35)  5.116 ms 108.170.255.123 (108.170.255.123)  4.916 ms 192.178.105.35 (192.178.105.35)  5.296 ms
 7  142.251.241.137 (142.251.241.137)  4.879 ms  4.621 ms 216.239.56.223 (216.239.56.223)  5.005 ms
 8  sea09s35-in-f4.1e100.net (142.251.215.228)  4.384 ms  4.348 ms  4.308 ms
francoer@rita:~$ traceroute 142.251.167.103
traceroute to 142.251.167.103 (142.251.167.103), 30 hops max, 60 byte packets
 1  * * *
 2  0015-opnsense.cat.pdx.edu (10.208.91.1)  0.217 ms  0.277 ms  0.232 ms
 3  CORE1.net.pdx.edu (131.252.5.142)  8.841 ms  8.820 ms  8.789 ms
 4  131.252.5.213 (131.252.5.213)  0.604 ms  0.648 ms  0.495 ms
 5  google.nwax.net (198.32.195.34)  3.944 ms  4.379 ms  4.332 ms
 6  192.178.105.129 (192.178.105.129)  4.394 ms 192.178.105.35 (192.178.105.35)  4.705 ms  5.017 ms
 7  108.170.255.128 (108.170.255.128)  4.986 ms 192.178.105.62 (192.178.105.62)  4.923 ms 108.170.255.132 (108.170.255.132)  4.576 ms
 8  216.239.41.34 (216.239.41.34)  11.118 ms 216.239.43.88 (216.239.43.88)  11.922 ms 142.251.64.250 (142.251.64.250)  11.248 ms
 9  142.250.213.63 (142.250.213.63)  52.462 ms 142.250.213.71 (142.250.213.71)  53.160 ms 142.251.226.161 (142.251.226.161)  52.108 ms
10  192.178.81.236 (192.178.81.236)  66.071 ms 192.178.81.224 (192.178.81.224)  66.775 ms 192.178.81.234 (192.178.81.234)  66.685 ms
11  142.250.211.50 (142.250.211.50)  67.372 ms 142.250.211.40 (142.250.211.40)  68.244 ms 142.251.244.161 (142.251.244.161)  65.992 ms
12  142.250.211.27 (142.250.211.27)  65.480 ms 142.250.235.95 (142.250.235.95)  66.791 ms 142.250.211.27 (142.250.211.27)  66.723 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  ww-in-f103.1e100.net (142.251.167.103)  64.443 ms  65.971 ms *
francoer@rita:~$ 
```

Wireshark 2.2.6


*ens4
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
icmp
```

```
No.   Time           Source          Destination     Protoc  Length Info
41615 166.113809819 74.125.195.104  10.138.0.2      ICMP        98 Echo (ping) reply    id=
41614 166.113424810 10.138.0.2      74.125.195.104  ICMP        98 Echo (ping) request  id=
41218 165.112560021 74.125.195.104  10.138.0.2      ICMP        98 Echo (ping) reply    id=
41217 165.112073688 10.138.0.2      74.125.195.104  ICMP        98 Echo (ping) request  id=
40909 164.113072185 74.125.195.104  10.138.0.2      ICMP        98 Echo (ping) reply    id=
40908 164.110299572 10.138.0.2      74.125.195.104  ICMP        98 Echo (ping) request  id=
36328 148.706013504 74.125.195.103  10.138.0.2      ICMP        98 Echo (ping) reply    id=
36327 148.705665784 10.138.0.2      74.125.195.103  ICMP        98 Echo (ping) request  id=
35965 147.705172832 74.125.195.103  10.138.0.2      ICMP        98 Echo (ping) reply    id=
35964 147.704759127 10.138.0.2      74.125.195.103  ICMP        98 Echo (ping) request  id=
35720 146.705430094 74.125.195.103  10.138.0.2      ICMP        98 Echo (ping) reply    id=
35719 146.702816377 10.138.0.2      74.125.195.103  ICMP        98 Echo (ping) request  id=
```

```
▸ Frame 41615: 98 bytes on wire (784 bits), 98 byt
▾ Ethernet II, Src: 42:01:0a:8a:00:01 (42:01:0a:8a
  ▾ Destination: 42:01:0a:8a:00:02 (42:01:0a:8a:0
      Address: 42:01:0a:8a:00:02 (42:01:0a:8a:00:
      .... ..1. .... .... .... .... = LG bit: Loc
      .... ...0 .... .... .... .... = IG bit: Ind
  ▸ Source: 42:01:0a:8a:00:01 (42:01:0a:8a:00:01)
    Type: IPv4 (0x0800)
▾ Internet Protocol Version 4, Src: 74.125.195.104
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS
    Total Length: 84
```

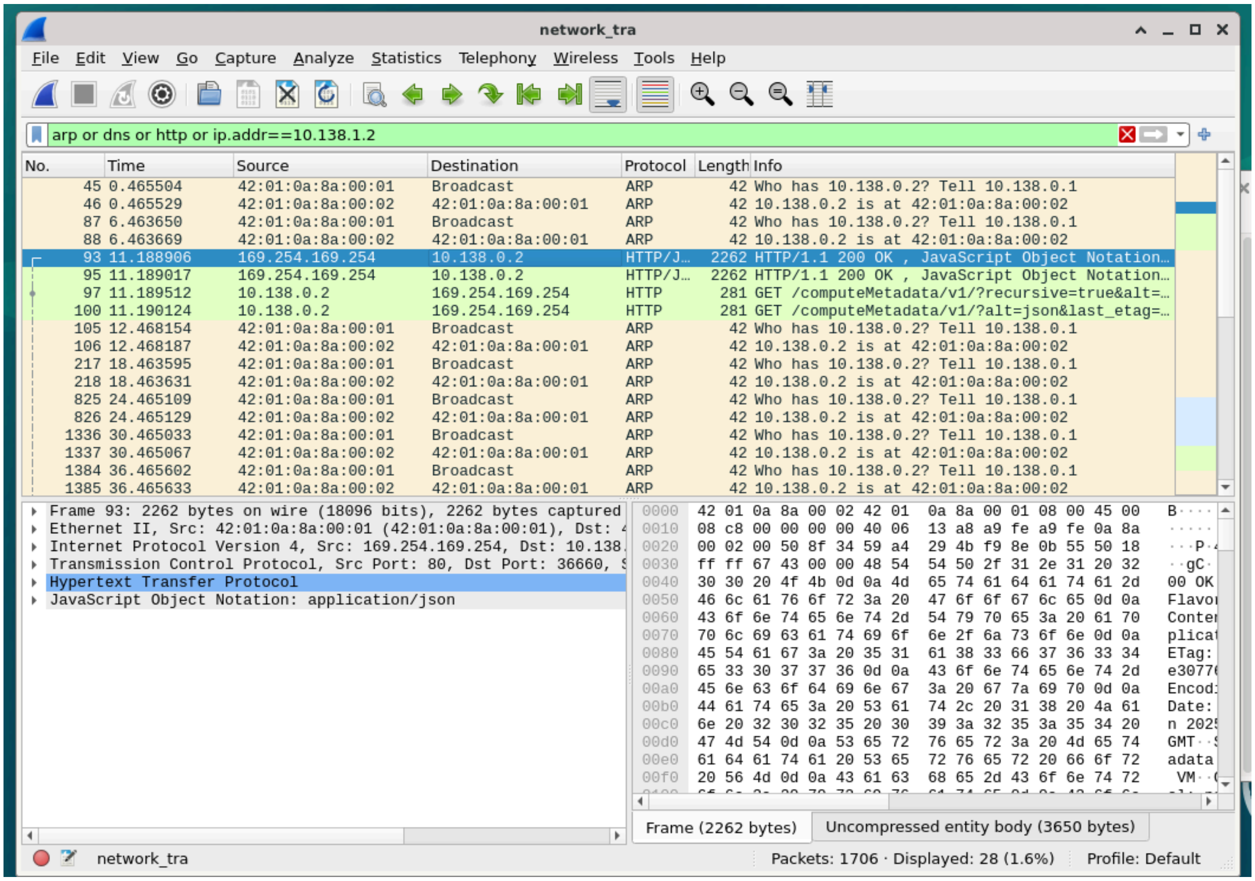
```
0000  42 01 0a 8a 00 02 42 01  0a 8a 00 01 08 00 45
0010  00 54 00 00 00 00 73 01  2f 38 4a 7d c3 68 0a
0020  00 02 00 00 b4 0a 16 c8  00 03 ba 67 8b 67 00
0030  00 00 29 88 07 00 00 00  00 00 10 11 12 13 14
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34
0060  36 37
```

```
Internet Control Message Protocol: Protocol    Packets: 59534 · Displayed: 18 (0.0%)    Profile: Default
```

- **Does the destination MAC address correspond to an interface on the VM, an interface on the default router or an interface on Google's web site?**
  -This should be the the interface on the default router

- **Does the destination MAC address correspond to an interface on the VM, an interface on the default router or an interface on Google's web site?**
  -This should be the interface on the vm

Wireshard 2.2.10


network_tra
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
arp or dns or http or ip.addr==10.138.1.2
```

```
No.   Time         Source              Destination         Protocol  Length Info
   45 0.465504     42:01:0a:8a:00:01   Broadcast           ARP         42 Who has 10.138.0.2? Tell 10.138.0.1
   46 0.465529     42:01:0a:8a:00:02   42:01:0a:8a:00:01   ARP         42 10.138.0.2 is at 42:01:0a:8a:00:02
   87 6.463650     42:01:0a:8a:00:01   Broadcast           ARP         42 Who has 10.138.0.2? Tell 10.138.0.1
   88 6.463669     42:01:0a:8a:00:02   42:01:0a:8a:00:01   ARP         42 10.138.0.2 is at 42:01:0a:8a:00:02
   93 11.188906    169.254.169.254     10.138.0.2          HTTP/J…   2262 HTTP/1.1 200 OK , JavaScript Object Notation…
   95 11.189017    169.254.169.254     10.138.0.2          HTTP/J…   2262 HTTP/1.1 200 OK , JavaScript Object Notation…
   97 11.189512    10.138.0.2          169.254.169.254     HTTP       281 GET /computeMetadata/v1/?recursive=true&alt=…
  100 11.190124    10.138.0.2          169.254.169.254     HTTP       281 GET /computeMetadata/v1/?alt=json&last_etag=…
  105 12.468154    42:01:0a:8a:00:01   Broadcast           ARP         42 Who has 10.138.0.2? Tell 10.138.0.1
  106 12.468187    42:01:0a:8a:00:02   42:01:0a:8a:00:01   ARP         42 10.138.0.2 is at 42:01:0a:8a:00:02
  217 18.463595    42:01:0a:8a:00:01   Broadcast           ARP         42 Who has 10.138.0.2? Tell 10.138.0.1
  218 18.463631    42:01:0a:8a:00:02   42:01:0a:8a:00:01   ARP         42 10.138.0.2 is at 42:01:0a:8a:00:02
  825 24.465109    42:01:0a:8a:00:01   Broadcast           ARP         42 Who has 10.138.0.2? Tell 10.138.0.1
  826 24.465129    42:01:0a:8a:00:02   42:01:0a:8a:00:01   ARP         42 10.138.0.2 is at 42:01:0a:8a:00:02
 1336 30.465033    42:01:0a:8a:00:01   Broadcast           ARP         42 Who has 10.138.0.2? Tell 10.138.0.1
 1337 30.465067    42:01:0a:8a:00:02   42:01:0a:8a:00:01   ARP         42 10.138.0.2 is at 42:01:0a:8a:00:02
 1384 36.465602    42:01:0a:8a:00:01   Broadcast           ARP         42 Who has 10.138.0.2? Tell 10.138.0.1
 1385 36.465633    42:01:0a:8a:00:02   42:01:0a:8a:00:01   ARP         42 10.138.0.2 is at 42:01:0a:8a:00:02
```

```
▸ Frame 93: 2262 bytes on wire (18096 bits), 2262 bytes captured
▸ Ethernet II, Src: 42:01:0a:8a:00:01 (42:01:0a:8a:00:01), Dst: 4
▸ Internet Protocol Version 4, Src: 169.254.169.254, Dst: 10.138.
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 36660, S
▸ Hypertext Transfer Protocol
▸ JavaScript Object Notation: application/json
```

```
0000  42 01 0a 8a 00 02 42 01  0a 8a 00 01 08 00 45 00   B····
0010  08 c8 00 00 00 00 40 06  13 a8 a9 fe a9 fe 0a 8a   ·····
0020  00 02 00 50 8f 34 59 a4  29 4b f9 8e 0b 55 50 18   ···P·
0030  ff ff 67 43 00 00 48 54  54 50 2f 31 2e 31 20 32   ··gC·
0040  30 30 20 4f 4b 0d 0a 4d  65 74 61 64 61 74 61 2d   00 OK
0050  46 6c 61 76 6f 72 3a 20  47 6f 6f 67 6c 65 0d 0a   Flavo
0060  43 6f 6e 74 65 6e 74 2d  54 79 70 65 3a 20 61 70   Conte
0070  70 6c 69 63 61 74 69 6f  6e 2f 6a 73 6f 6e 0d 0a   plica
0080  45 54 61 67 3a 20 35 31  61 38 33 66 37 36 33 34   ETag:
0090  65 33 30 37 37 36 0d 0a  43 6f 6e 74 65 6e 74 2d   e3077
00a0  45 6e 63 6f 64 69 6e 67  3a 20 67 7a 69 70 0d 0a   Encod
00b0  44 61 74 65 3a 20 53 61  74 2c 20 31 38 20 4a 61   Date:
00c0  6e 20 32 30 32 35 20 30  39 3a 32 35 3a 35 34 20   n 202
00d0  47 4d 54 0d 0a 53 65 72  76 65 72 3a 20 4d 65 74   GMT··
00e0  61 64 61 74 61 20 53 65  72 76 65 72 20 66 6f 72   adata
00f0  20 56 4d 0d 0a 43 61 63  68 65 2d 43 6f 6e 74 72    VM··
```

Frame (2262 bytes)
Uncompressed entity body (3650 bytes)
network_tra
Packets: 1706 · Displayed: 28 (1.6%)
Profile: Default

- **What packet numbers in the trace are the result of the VM attempting to get the hardware address of the default router?**

  Packets 45, 87, 105, 217, 825, 1336, 1384

- **What is this hardware address?**
  42:01:0a:8a:00:02

What packet numbers in the trace correspond to the DNS request for the web site?
Packets 1417, 1426

What is the IP address of the local DNS server being queried?
169.254.169.254

TCP
What packet numbers in the trace correspond to the initial TCP handshake for the web request?

Packet 1427

How long does it take to perform the initial TCP handshake?
41.14 ms maybe seconds

For the tcp I found this my changing what it will show and set to the flags for the handshake and only got one result back and besides that nothing popped up

HTTP
What packet numbers in the trace correspond to the actual HTTP request and response?
Packet 1431, 1435
How long does it take to process the HTTP request after the handshake?
41 seconds/maybe ms idk the units