

Table of Contents

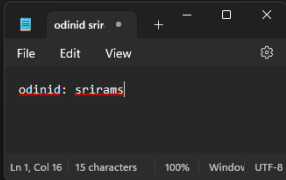
2.1: TCP, HTTP	1
2.1.1 TCP #1: Sockets	1
2.1.2 -	4
2.1.3 TCP #2: Throughput	4
2.1.4 -	5
2.1.5 HTTP #3: Requests	6
2.1.6 -	14
2.2: DNS, Recap	15
2.2.1 DNS reconnaissance #1 (dig)	15
2.2.2 Iterative DNS lookups	17
2.2.3 Reverse DNS lookups	18
2.2.4 Host enumeration	19
2.2.5 Geographic DNS #2	20
2.2.6 Wireshark Lab #3	25
2.2.7 -	26
2.2.8 Network Recap Lab #4	28
2.2.9. Collect trace	29

2.1: TCP, HTTP

2.1.1 TCP #1: Sockets

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

find a single command and its command-line flags that, when executed, lists all TCP sockets in a LISTEN state on an IPv4 address, showing the program that is using it.



To list all TCP sockets in a LISTEN state on an IPv4 address and show the program using it, you can use the following `ss` (socket statistics) command:

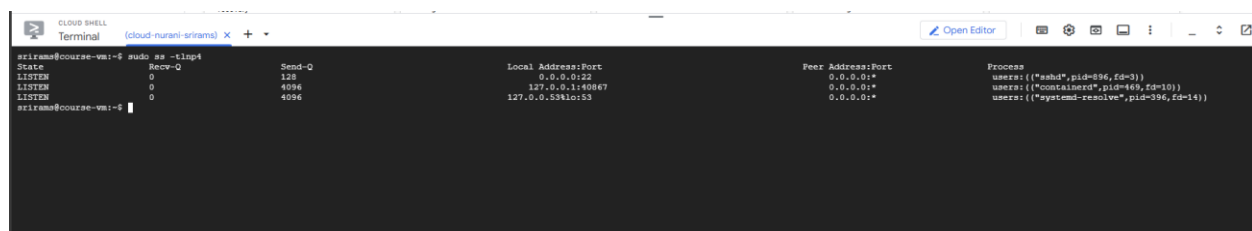
```
ss -tlnp4
```

Here's a breakdown of the command:

- `-t`: Show TCP sockets.
- `-l`: Show only listening sockets.
- `-n`: Show numerical addresses (don't resolve hostnames).
- `-p`: Show the process using the socket.
- `4`: Restrict to IPv4 sockets.

This command will list all the TCP sockets in the LISTEN state for IPv4, along with the process using each socket.

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



```

#sriams@course-ws1:~$ sudo ss -tlnp4
State     Recv-Q    Send-Q     Local Address:Port      Peer Address:Port      Process
LISTEN     0         128      0.0.0.0:22              0.0.0.0:*               users: (*sshd*,pid=896,fd=3)
LISTEN     0         4096     127.0.0.1:40847         0.0.0.0:*               users: (*containerd*,pid=469,fd=10)
LISTEN     0         4096     127.0.0.0:53            0.0.0.0:*               users: (*systemd-resolve*,pid=356,fd=14)

```

List a service that can be contacted from any interface on the machine. List a service that can only be contacted by local processes.

1. Service that can be contacted from any Interface on machine

Secure Shell (SSH)

IP Address : 0.0.0.0

Port: 22

2. Service Contactable Only by Local Processes

System DNS Resolver (systemd-resolve)

IP Address: 127.0.0.53

Port: 53

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

```
srirams@ada:~$ ss -tlnp4
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
LISTEN	0	4096	127.0.0.0:54153	0.0.0.0:*	
LISTEN	0	4096	127.0.0.1:46624	0.0.0.0:*	
LISTEN	0	1024	127.0.0.1:43577	0.0.0.0:*	
LISTEN	0	1024	127.0.0.1:43807	0.0.0.0:*	
LISTEN	0	1024	127.0.0.1:42759	0.0.0.0:*	
LISTEN	0	1024	127.0.0.1:39891	0.0.0.0:*	
LISTEN	0	1024	127.0.0.1:38155	0.0.0.0:*	
LISTEN	0	1024	127.0.0.1:34431	0.0.0.0:*	
LISTEN	0	1024	127.0.0.1:33919	0.0.0.0:*	
LISTEN	0	4096	127.0.0.53%lo:53	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6112	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6113	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6115	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6116	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6104	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6105	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6107	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6108	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6109	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6110	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6111	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6101	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6102	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:6103	0.0.0.0:*	
LISTEN	0	4096	127.0.0.1:631	0.0.0.0:*	
LISTEN	0	100	127.0.0.1:25	0.0.0.0:*	

```
srirams@ada:~$
```

List the services that this machine provides for external access

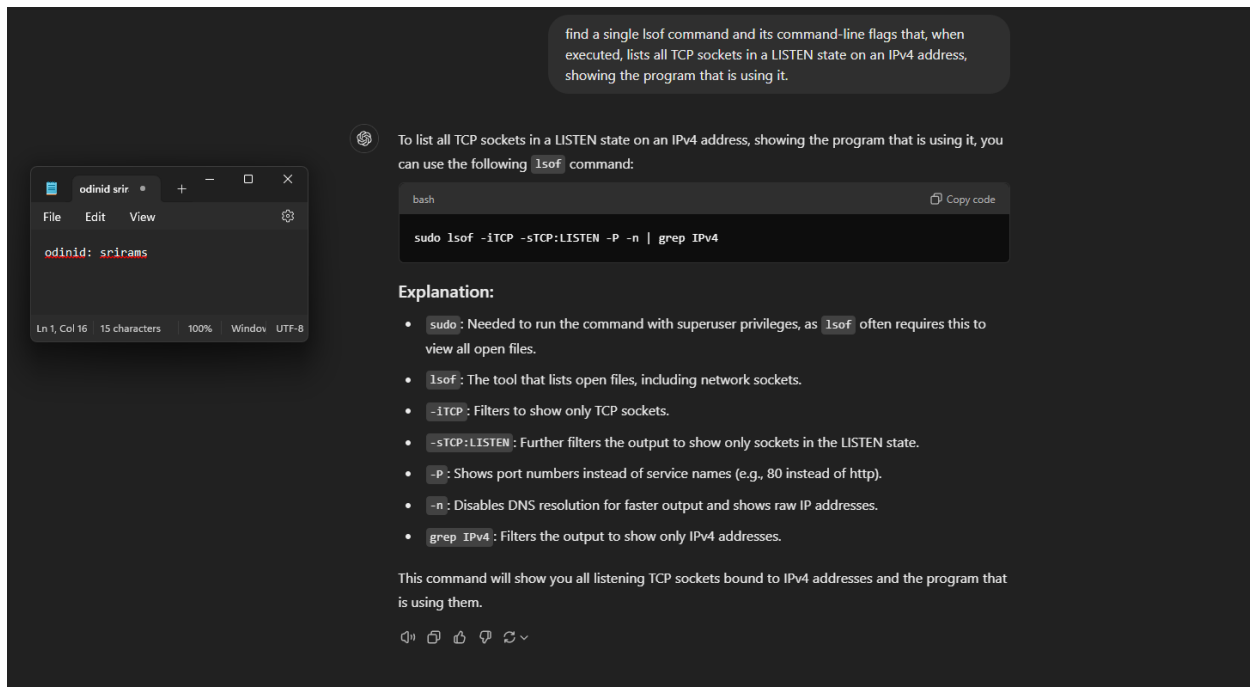
The machine provides **Secure Shell (SSH)** on **Port 22** and **Remote Desktop** on **port 3389** for external access.

udp	UNCONN	0	0	0.0.0.0:41175	0.0.0.0:*
udp	UNCONN	0	0	0.0.0.0:46476	0.0.0.0:*
udp	UNCONN	0	0	0.0.0.0:48441	0.0.0.0:*
udp	UNCONN	0	0	0.0.0.0:49242	0.0.0.0:*
udp	UNCONN	0	0	0.0.0.0:52077	0.0.0.0:*
udp	UNCONN	0	0	0.0.0.0:50339	0.0.0.0:*
udp	UNCONN	0	0	127.0.0.54:53	0.0.0.0:*
udp	UNCONN	0	0	127.0.0.53%lo:53	0.0.0.0:*
udp	UNCONN	0	0	131.252.208.103%ena3:68	0.0.0.0:*
tcp	LISTEN	0	4096	127.0.0.54:53	0.0.0.0:*
tcp	LISTEN	0	4096	127.0.0.1:46624	0.0.0.0:*
tcp	LISTEN	0	1024	127.0.0.1:43577	0.0.0.0:*
tcp	LISTEN	0	1024	127.0.0.1:43807	0.0.0.0:*
tcp	LISTEN	0	1024	127.0.0.1:43235	0.0.0.0:*
tcp	LISTEN	0	1024	127.0.0.1:39891	0.0.0.0:*
tcp	LISTEN	0	1024	127.0.0.1:38155	0.0.0.0:*
tcp	LISTEN	0	1024	127.0.0.1:34431	0.0.0.0:*
tcp	LISTEN	0	1024	127.0.0.1:33919	0.0.0.0:*
tcp	LISTEN	0	4096	127.0.0.53%lo:53	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6112	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6113	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6115	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6116	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6104	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6105	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6107	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6108	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6109	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6110	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6111	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6100	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6101	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6102	0.0.0.0:*
tcp	LISTEN	0	128	127.0.0.1:6103	0.0.0.0:*
tcp	LISTEN	0	4096	127.0.0.1:631	0.0.0.0:*
tcp	LISTEN	0	100	127.0.0.1:25	0.0.0.0:*
tcp	LISTEN	0	100	:::1:25	:::1:*
tcp	LISTEN	0	4096	:::1:631	:::1:*
tcp	LISTEN	0	2	:::1:3350	:::1:*
tcp	LISTEN	0	128	:::1:6102	:::1:*
tcp	LISTEN	0	128	:::1:6103	:::1:*
tcp	LISTEN	0	128	:::1:6100	:::1:*
tcp	LISTEN	0	128	:::1:6101	:::1:*
tcp	LISTEN	0	128	:::1:6110	:::1:*
tcp	LISTEN	0	128	:::1:6111	:::1:*
tcp	LISTEN	0	128	:::1:6108	:::1:*
tcp	LISTEN	0	128	:::1:6109	:::1:*
tcp	LISTEN	0	128	:::1:6107	:::1:*
tcp	LISTEN	0	128	:::1:6104	:::1:*
tcp	LISTEN	0	128	:::1:6105	:::1:*
tcp	LISTEN	0	128	:::1:6116	:::1:*
tcp	LISTEN	0	128	:::1:6115	:::1:*
tcp	LISTEN	0	128	:::1:6112	:::1:*
tcp	LISTEN	0	128	:::1:6113	:::1:*
tcp	LISTEN	0	4096	:::1:22	:::1:*
tcp	LISTEN	0	2	:::1:3389	:::1:*

```
srirams@ada:~$
```

2.1.2 -

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



The screenshot shows a ChatGPT interface with a dark theme. At the top, a prompt box contains the text: "find a single lsof command and its command-line flags that, when executed, lists all TCP sockets in a LISTEN state on an IPv4 address, showing the program that is using it." Below the prompt, a response box shows a terminal window on the left and a code block on the right. The terminal window shows a prompt "odinid: srirams" and a cursor. The code block contains the command: `sudo lsof -iTCP -sTCP:LISTEN -P -n | grep IPv4`. Below the code block, an "Explanation:" section lists several bullet points explaining the command's components:

- `sudo`: Needed to run the command with superuser privileges, as `lsof` often requires this to view all open files.
- `lsof`: The tool that lists open files, including network sockets.
- `-iTCP`: Filters to show only TCP sockets.
- `-sTCP:LISTEN`: Further filters the output to show only sockets in the LISTEN state.
- `-P`: Shows port numbers instead of service names (e.g., 80 instead of http).
- `-n`: Disables DNS resolution for faster output and shows raw IP addresses.
- `grep IPv4`: Filters the output to show only IPv4 addresses.

At the bottom, a note states: "This command will show you all listening TCP sockets bound to IPv4 addresses and the program that is using them." Below the note are icons for voice, copy, share, and a dropdown menu.

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

```
srirams@course-vm:~$ sudo lsof -iTCP -sTCP:LISTEN -P -n | grep IPv4
systemd-r 396 systemd-resolve 14u IPv4 3985 0t0 TCP 127.0.0.53:53 (LISTEN)
container 469 root 10u IPv4 4683 0t0 TCP 127.0.0.1:40867 (LISTEN)
sshd 896 root 3u IPv4 5107 0t0 TCP *:22 (LISTEN)
srirams@course-vm:~$
```

2.1.3 TCP #2: Throughput

VM Instances:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input checked="" type="checkbox"/>	course-vm	us-west1-b			10.138.0.2 (nic0)	34.168.156.204 (nic0)	SSH
<input type="checkbox"/>	vm-australia-southeast1-b	australia-southeast1-b			10.152.0.2 (nic0)	35.197.186.72 (nic0)	SSH
<input type="checkbox"/>	vm-europe-west1-d	europe-west1-d			10.132.0.2 (nic0)	34.76.41.206 (nic0)	SSH
<input type="checkbox"/>	vm-us-east1-b	us-east1-b			10.142.0.3 (nic0)	35.231.205.219 (nic0)	SSH
<input type="checkbox"/>	vm-us-west1-b	us-west1-b			10.138.0.6 (nic0)	34.83.199.230 (nic0)	SSH

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.

```
sriams@vm-us-west1-b:~$ iperf -c 10.152.0.2 -p 80
-----
Client connecting to 10.152.0.2, TCP port 80
TCP window size: 85.0 KByte (default)
-----
[ 1] local 10.138.0.6 port 52774 connected with 10.152.0.2 port 80
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.2169 sec  175 MBytes  144 Mbits/sec
sriams@vm-us-west1-b:~$ iperf -c 10.132.0.2 -p 80
-----
Client connecting to 10.132.0.2, TCP port 80
TCP window size: 85.0 KByte (default)
-----
[ 1] local 10.138.0.6 port 48208 connected with 10.132.0.2 port 80
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.1752 sec  187 MBytes  154 Mbits/sec
sriams@vm-us-west1-b:~$ iperf -c 10.142.0.3 -p 80
-----
Client connecting to 10.142.0.3, TCP port 80
TCP window size: 85.0 KByte (default)
-----
[ 1] local 10.138.0.6 port 35894 connected with 10.142.0.3 port 80
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.1111 sec  373 MBytes  309 Mbits/sec
sriams@vm-us-west1-b:~$
```

Connection to 10.152.0.2 (vm-australia-southeast1-b) VM1:

Bandwidth: 144 Mbits/sec

Connection to 10.132.0.2 (vm-europe-west1-d) VM2:

Bandwidth: 154 Mbits/sec

Connection to 10.142.0.3 (vm-us-east1-b) VM3:

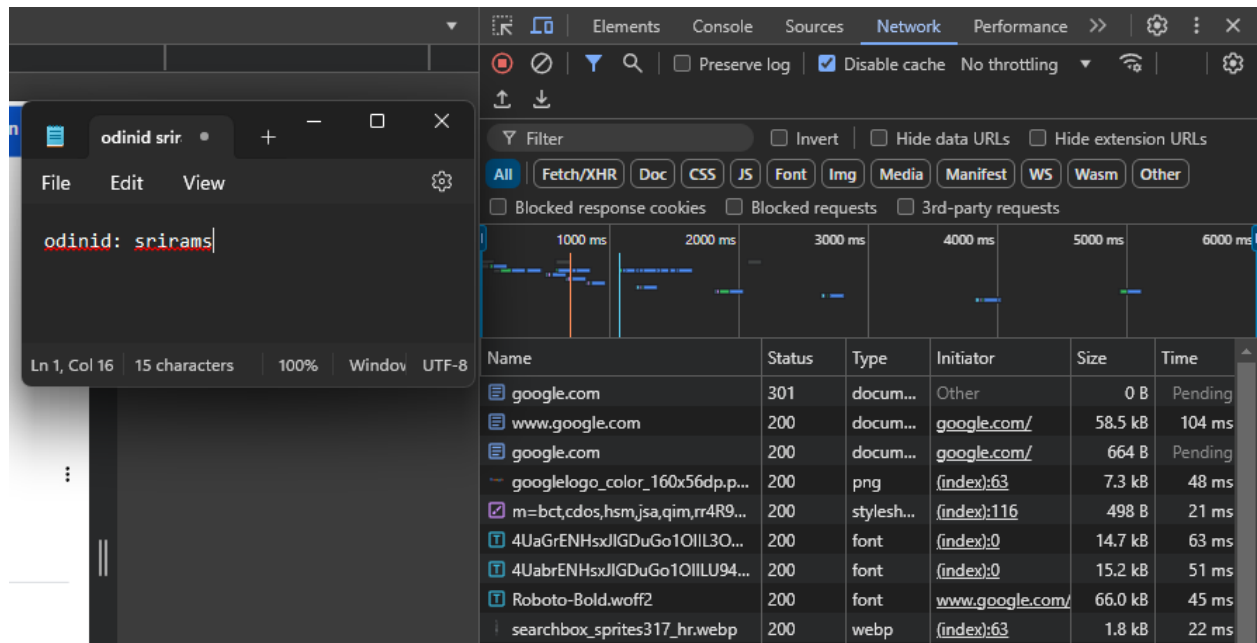
Bandwidth: 309 Mbits/sec

- The connection to **VM3** has the highest bandwidth at **309 Mbits/sec**, indicating the best network performance.
- The connection to **VM2** has a slightly better bandwidth of **154 Mbits/sec** compared to **VM1** but still lower than **VM3**.
- The connection to **VM1** shows the lowest bandwidth at **144 Mbits/sec**, indicating relatively poorer network performance.

These differences in bandwidth are likely due to factors such as latency, distance between VMs, and network congestion at the time of testing. This demonstrates a clear variation in network performance across the VMs.

2.1.5 HTTP #3: Requests

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

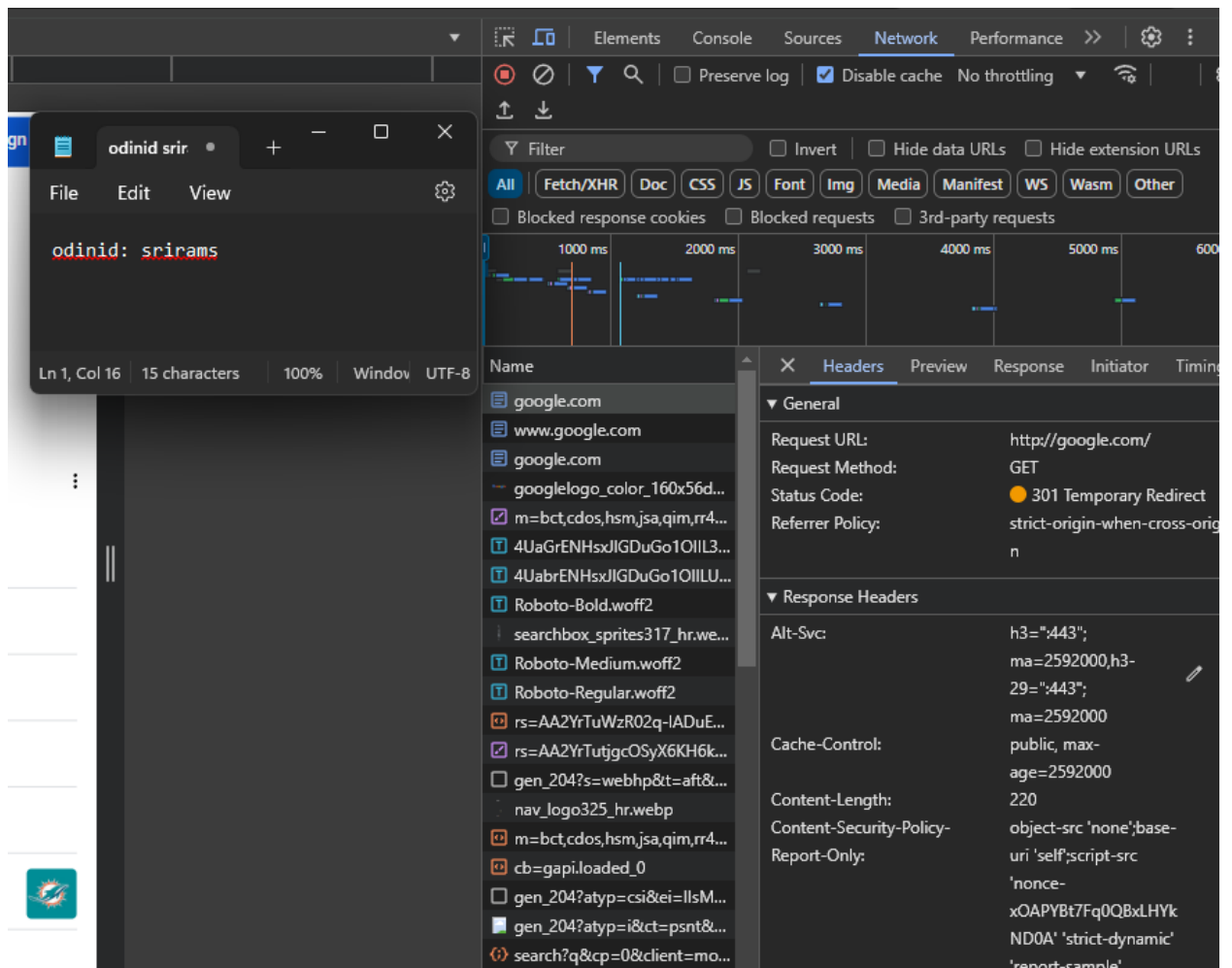


- What is the URL being requested?

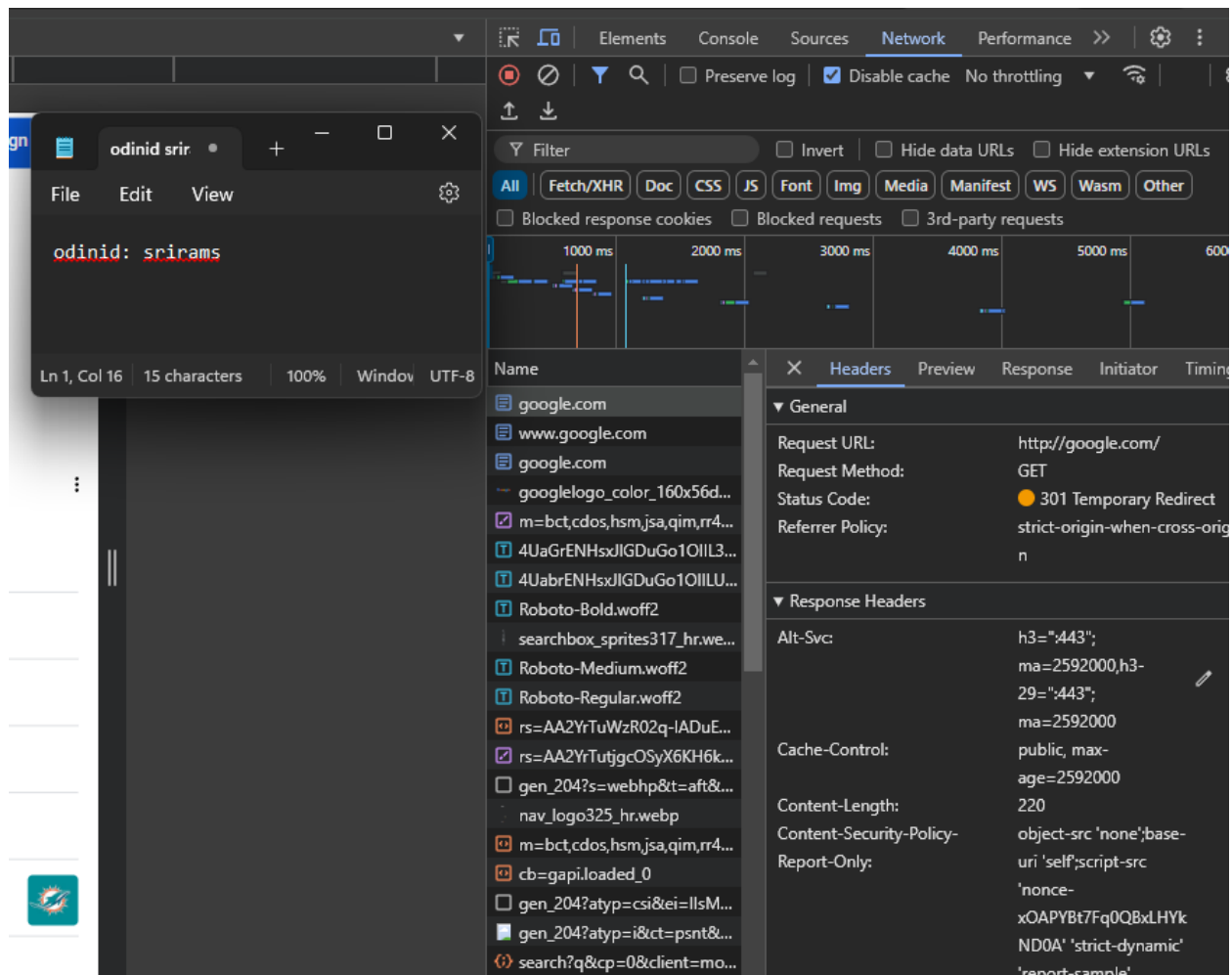
<http://google.com/>

- Explain the HTTP status code that is returned and what the code indicates

HTTP status code received is 301 and the HTTP status code 301 indicates that a requested resource has been permanently moved to a new URL.



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).

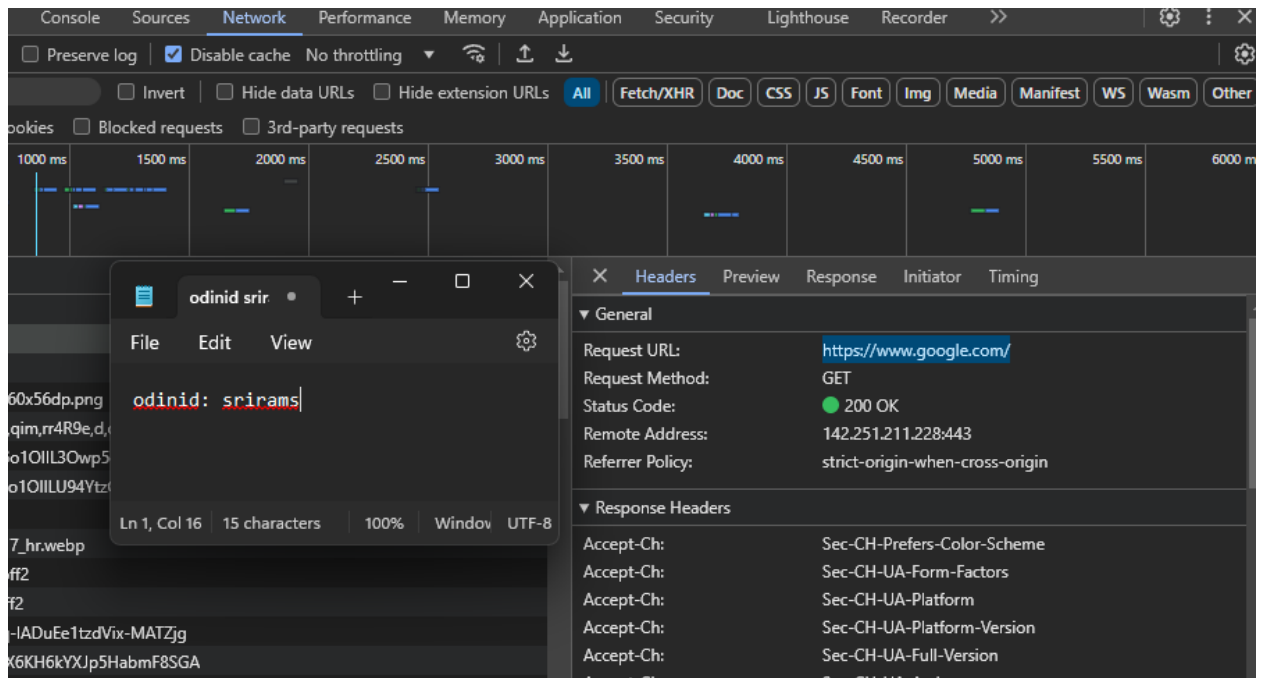


- **What is the URL being requested?**

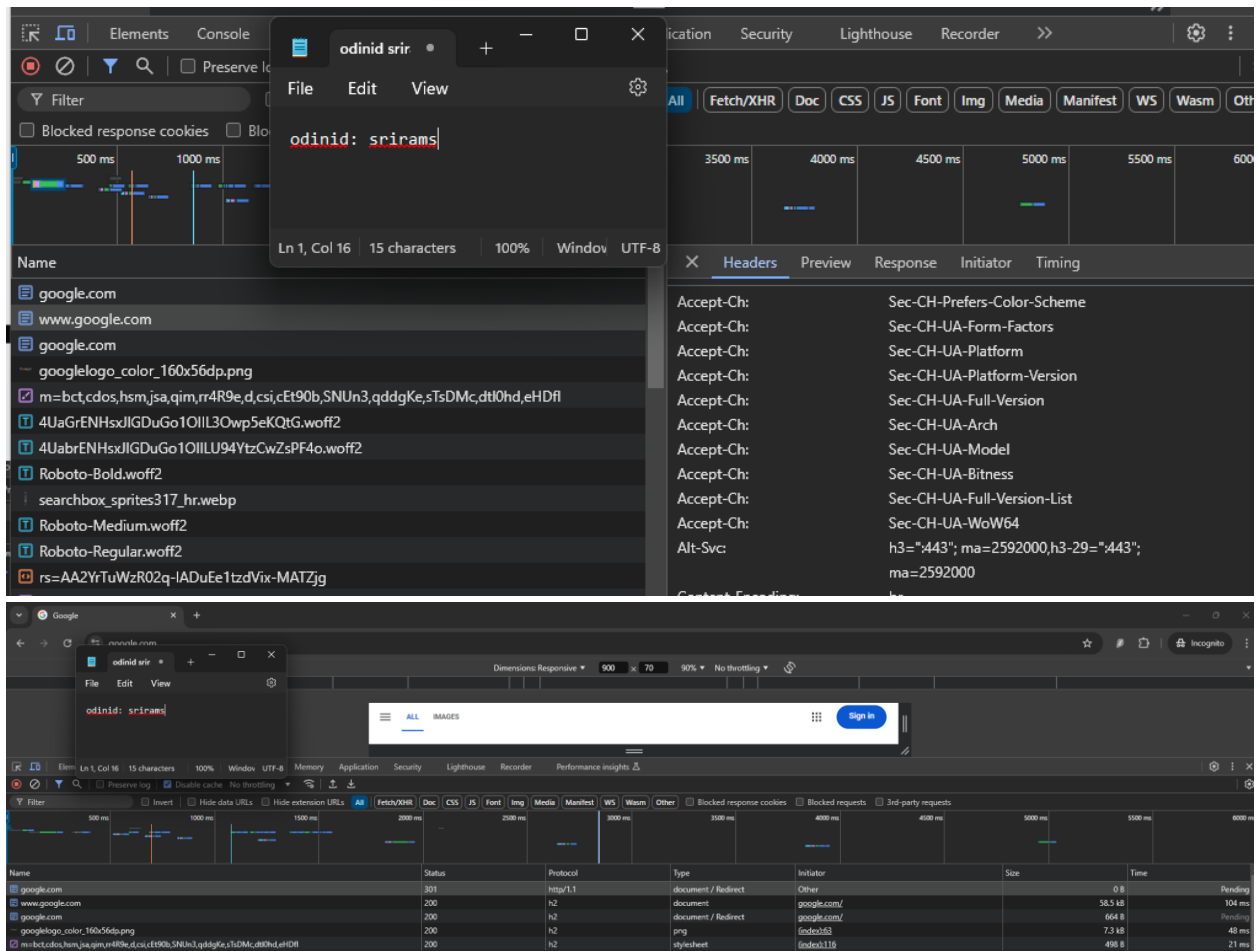
<https://www.google.com/>

- **Explain the HTTP status code that is returned and what the code indicates**

HTTP status code received is 200 indicates that a request was successful



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).

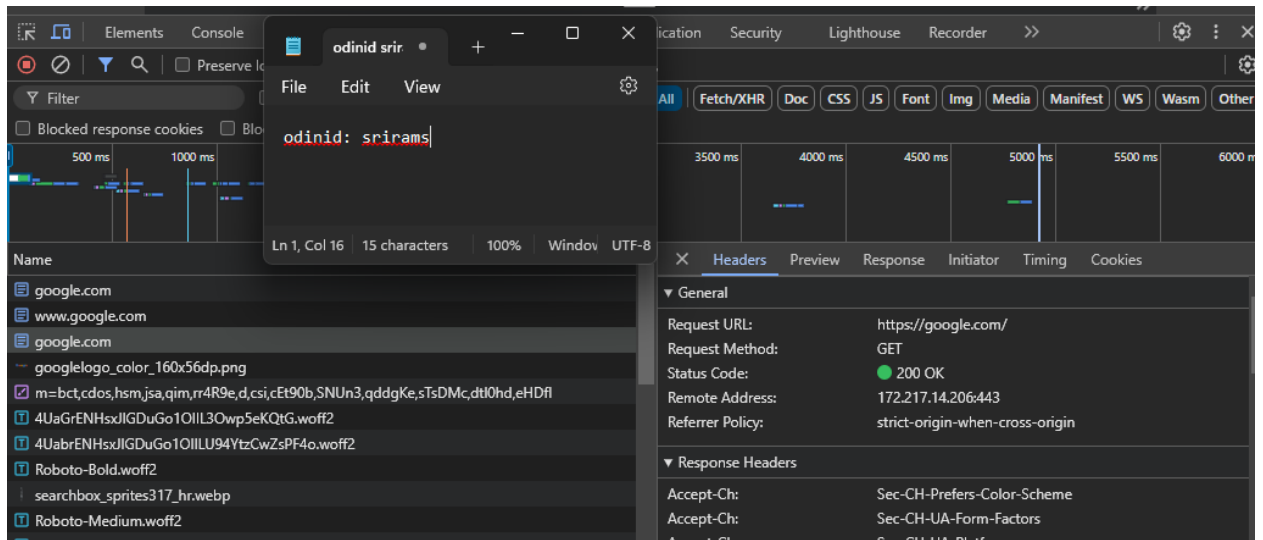


- What is the URL being requested?

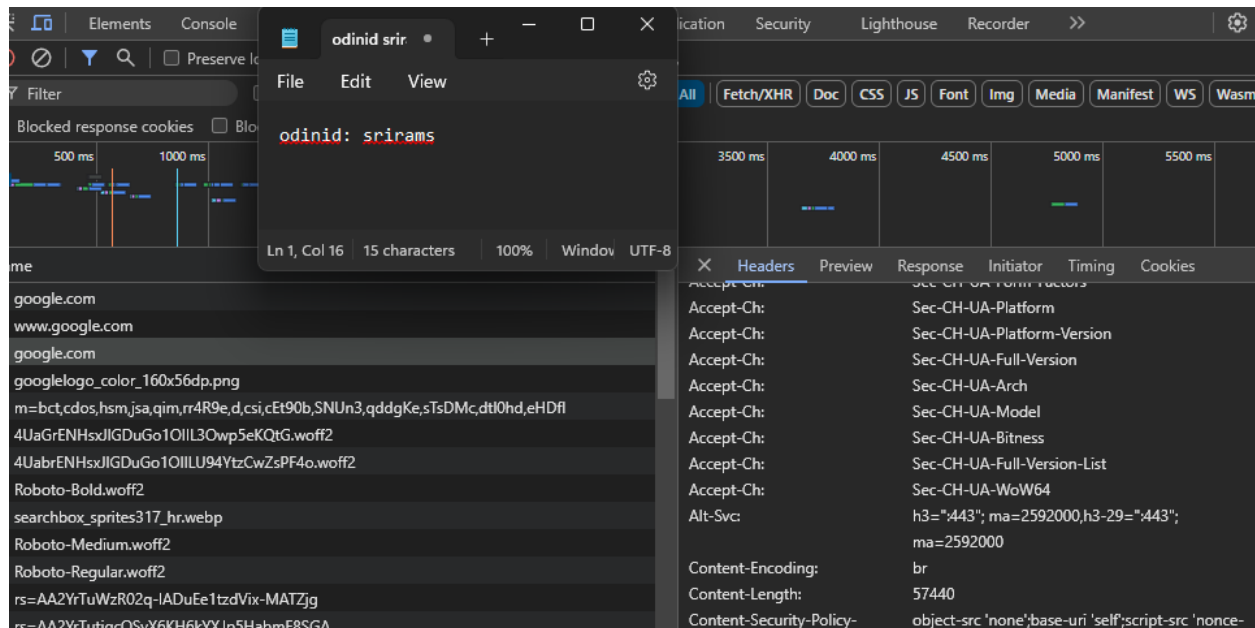
<https://google.com/>

- Explain the HTTP status code that is returned and what the code indicates

HTTP status code received is 200 indicates that a request was successful



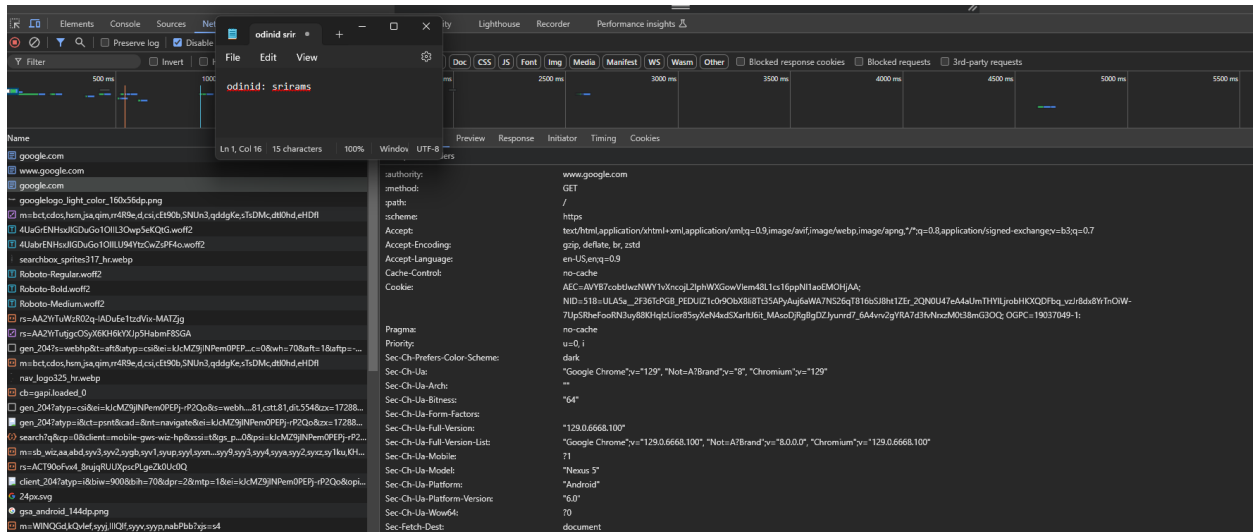
Take a screenshot indicating the version of the HTTP protocol that is used for each request. (Hint: look at the response status line and alt-svc: HTTP response headers indicating HTTP/2 or HTTP/3).



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

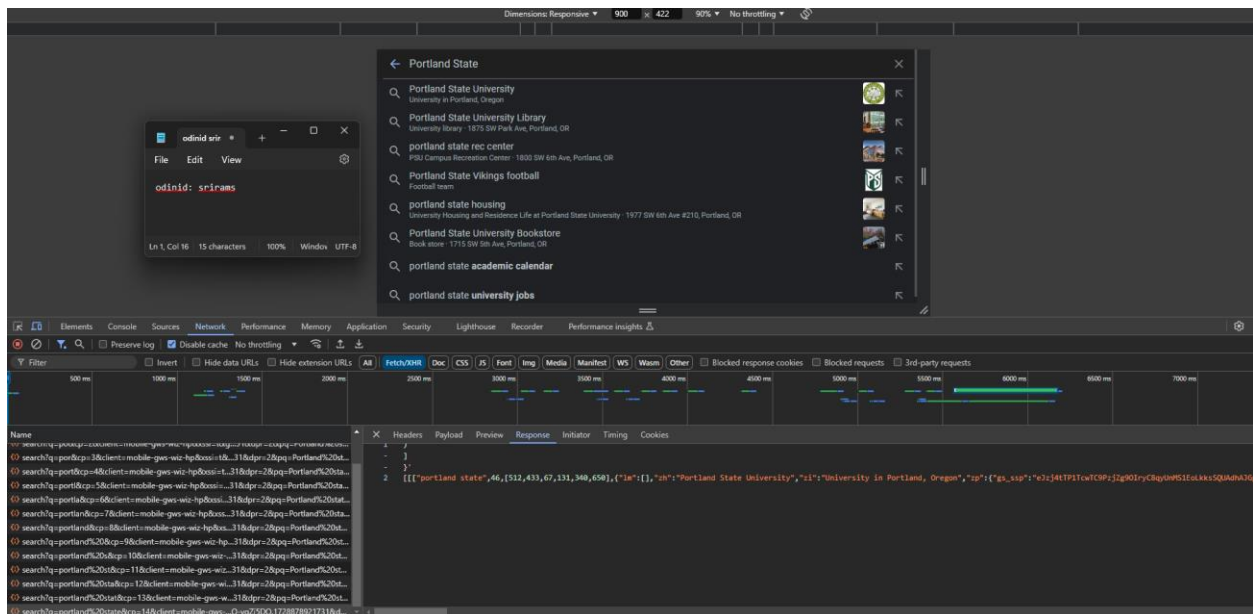
<https://www.google.com/>

<https://google.com/>



2.1.6–

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 "rabid"

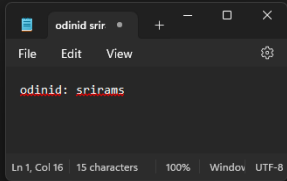


2.2: DNS, Recap

2.2.1 DNS reconnaissance #1 (dig)

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

Using ChatGPT, 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.



odinid: srirams

To use the `dig` command to query Portland State University's (PSU) local DNS server at `131.252.208.53` for the A record of `www.pdx.edu` using TCP, the following command can be used:

```
bash
```

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

Copy code

Explanation:

- `@131.252.208.53`: Specifies the DNS server to query, in this case, PSU's local DNS server.
- `www.pdx.edu`: The domain name you are querying for.
- `A`: Requests the A (address) record, which provides the IPv4 address of the domain.
- `+tcp`: Forces the query to use TCP instead of the default UDP.

This command will return the A record for `www.pdx.edu` using TCP from PSU's DNS server at `131.252.208.53`.

```
srirams@ada:~$ dig @131.252.208.53 www.pdx.edu A +tcp

; <<>> DiG 9.18.28-Ubuntu0.24.04.1-Ubuntu <<>> @131.252.208.53 www.pdx.edu A +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1553
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: c9f5cb621072522501000000670c9b0849682f4abbeb36ba (good)
;; QUESTION SECTION:
;www.pdx.edu.                IN      A

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

;; Query time: 78 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Sun Oct 13 21:16:08 PDT 2024
;; MSG SIZE rcvd: 132

srirams@ada:~$
```

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

```

srirams@ada:~$ dig @131.252.208.53 pdx.edu MX +tcp

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @131.252.208.53 pdx.edu MX +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56872
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 28b81fef9bd6c35701000000670c9b7a7441798d514312ed (good)
;; QUESTION SECTION:
;pdx.edu.                IN      MX

;; ANSWER SECTION:
pdx.edu.                67322   IN      MX      1 aspmx.l.google.com.
pdx.edu.                67322   IN      MX      5 alt2.aspmx.l.google.com.
pdx.edu.                67322   IN      MX      10 alt3.aspmx.l.google.com.
pdx.edu.                67322   IN      MX      10 alt4.aspmx.l.google.com.
pdx.edu.                67322   IN      MX      5 alt1.aspmx.l.google.com.

;; Query time: 1 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Sun Oct 13 21:18:02 PDT 2024
;; MSG SIZE rcvd: 182

srirams@ada:~$

```

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

Amazon Webservises (AWS)

- What cloud provider handles mail for pdx.edu?

Based on the Above MX screenshot taken for pdx.edu, We can infer it is using the Google Workspace.


```

srirams@ada:~$ dig mashimaro.cs.pdx.edu NS +tcp

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> mashimaro.cs.pdx.edu NS +tcp
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59097
;; 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. 2024100200 600 300 1209600 300

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (TCP)
;; WHEN: Sun Oct 13 21:31:14 PDT 2024
;; MSG SIZE rcvd: 105

srirams@ada:~$

```

```

srirams@ada:~$ dig mashimaro.cs.pdx.edu A +norecurse @walt.ee.pdx.edu +tcp

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> mashimaro.cs.pdx.edu A +norecurse @walt.ee.pdx.edu +tcp
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46949
;; flags: qr aa ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 8f79a76ca003191a01000000670c9f18f7b523f0872e8557 (good)
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      A

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

;; AUTHORITY SECTION:
cs.pdx.edu.                    14400   IN      NS       adns1.cat.pdx.edu.
cs.pdx.edu.                    14400   IN      NS       adns3.cat.pdx.edu.
cs.pdx.edu.                    14400   IN      NS       adns2.cat.pdx.edu.

;; Query time: 1 msec
;; SERVER: 131.252.208.38#53(walt.ee.pdx.edu) (TCP)
;; WHEN: Sun Oct 13 21:33:28 PDT 2024
;; MSG SIZE rcvd: 164

srirams@ada:~$

```

2.2.2 Iterative DNS lookups

- List all of the iterative dig commands performed for the lookup

dig

dig @a.root-servers.net F.ROOT-SERVERS.NET A +tcp +norecurse

dig @192.5.5.241 com. NS +tcp +norecurse

dig @192.35.51.30 google.com NS +tcp +norecurse

dig @216.239.34.10 cloud.google.com NS +tcp +norecurse

dig @ns1.google.com console.cloud.google.com A +tcp +norecurse

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

```
Last login: Mon Oct 14 01:20:45 2024 from 34.145.112.130
srirams@course-vm:~$ dig @ns1.google.com console.cloud.google.com A +tcp +norecurse

; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> @ns1.google.com console.cloud.google.com A +tcp +norecurse
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 36073
;; flags: qr aa; QUERY: 1, ANSWER: 7, 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   www3.1.google.com.
www3.1.google.com.       300     IN      A       74.125.20.100
www3.1.google.com.       300     IN      A       74.125.20.113
www3.1.google.com.       300     IN      A       74.125.20.102
www3.1.google.com.       300     IN      A       74.125.20.139
www3.1.google.com.       300     IN      A       74.125.20.138
www3.1.google.com.       300     IN      A       74.125.20.101

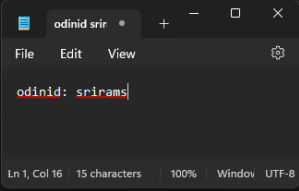
;; Query time: 17 msec
;; SERVER: 216.239.32.10#53(ns1.google.com) (TCP)
;; WHEN: Mon Oct 14 04:53:58 UTC 2024
;; MSG SIZE rcvd: 170

srirams@course-vm:~$
```

2.2.3 Reverse DNS lookups

- Take a screenshot of the prompt and the command produced

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 using `dig`, `egrep`, and `awk`:

```
bash
```

```
dig +short espn.go.com | egrep '^([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)$' | awk '{print $1}'
```

Copy code

Breakdown:

1. `dig +short espn.go.com`: This retrieves the DNS records for `espn.go.com` in a short format.
2. `egrep '^([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)$'`: This filters the output to include only lines that contain IPv4 addresses (ignoring other DNS records like IPv6).
3. `awk '{print $1}'`: This prints the first field, which is the IPv4 address.

🔊 📄 🌟 🔄 ↻

```
srirams@course-vm:~$ dig +short espn.go.com | egrep '^([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)$' | awk '{print $1}'
18.238.238.108
18.238.238.105
18.238.238.60
18.238.238.27
srirams@course-vm:~$
```

- Take a screenshot of the command and its results for your lab notebook

```
srirams@course-vm:~$ for ip in $(dig +short espn.go.com | egrep '^([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)$'); do
  dig -x $ip +short | egrep -v '^$' | awk '{print $1}';
done
server-18-238-238-60.sea90.r.cloudfront.net.
server-18-238-238-108.sea90.r.cloudfront.net.
server-18-238-238-27.sea90.r.cloudfront.net.
server-18-238-238-105.sea90.r.cloudfront.net.
srirams@course-vm:~$
```

2.2.4 Host enumeration

- Take a screenshot of the results in your lab notebook

```
srirams@ada:~$ 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.
srirams@ada:~$
```

2.2.5 Geographic DNS #2

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

131.252.208.53

Geolocation data from ipinfo.io		Product: API, real-time
IP ADDRESS: 131.252.208.53	ISP: Not available	
COUNTRY: United States	ORGANIZATION: AS6366 Portland State University	
REGION: Oregon	LATITUDE: 45.5234	
CITY: Portland	LONGITUDE: -122.6762	
Incorrect location? Contact ipinfo.io		view map

Geolocation data from DB-IP		Product: API, real-time
IP ADDRESS: 131.252.208.53	ISP: Portland State University	
COUNTRY: United States	ORGANIZATION: Portland State University	
REGION: Oregon	LATITUDE: 45.584	
CITY: Portland (North Portland)	LONGITUDE: -122.728	
Incorrect location? Contact DB-IP		view map



IP Address Articles

odinid srir

File Edit View

odinid: srirams

Ln 1, Col 16 15 characters 100% Window UTF-8

How to hide my IP address?

There are several ways to hide your IP address, and your geolocation. Hiding your IP address is concealing your "true" IP address with a different one. You may use a VPN, Proxy or Anonymous Browser to hide your IP address.

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

198.82.247.66

Geolocation data from ipinfo.io		Product: API, real-time
IP ADDRESS: 198.82.247.66	ISP: Not available	
COUNTRY: United States	ORGANIZATION: AS1312 Virginia Polytechnic Institute and State Univ.	
REGION: Virginia	LATITUDE: 37.2296	
CITY: Blacksburg	LONGITUDE: -80.4139	
Incorrect location? Contact ipinfo.io		view map

Geolocation data from DB-IP		Product: API, real-time
IP ADDRESS: 198.82.247.66	ISP: Virginia Polytechnic Institute and State Univ.	
COUNTRY: United States	ORGANIZATION: Virginia Polytechnic Institute and State Univ.	
REGION: Virginia	LATITUDE: 37.2037	
CITY: Blacksburg (Farmview - Ramble)	LONGITUDE: -80.4143	
Incorrect location? Contact DB-IP		view map

IP Address Articles

odinid srir

File Edit View

odinid: srirams

Ln 1, Col 16 15 characters 100% Window UTF-8

IP address is concealing your "true" IP address with a different one. You may use a VPN, Proxy or Anonymous Browser to hide your IP address.

[Learn more >](#)

Change your IP Address

- Record one address for www.google.com from each result for your lab notebook.

For 131.252.208.53

```

srirams@ada:~$ 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: 21116
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 513b9518345e922e01000000670d98387cd0b16ble56db75 (good)
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                 193     IN      A      142.250.217.100

;; Query time: 1 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (UDP)
;; WHEN: Mon Oct 14 15:16:24 PDT 2024
;; MSG SIZE rcvd: 87

srirams@ada:~$

```

For 198.82.247.66

```

srirams@ada:~$ 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: 16913
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 022baab2c79aa152366f1006670d98a59017331a89e4a136 (good)
;; QUESTION SECTION:
;www.google.com.                IN      A











;; ANSWER SECTION:
www.google.com.                 40      IN      A      142.251.167.105
www.google.com.                 40      IN      A      142.251.167.104
www.google.com.                 40      IN      A      142.251.167.99
www.google.com.                 40      IN      A      142.251.167.103
www.google.com.                 40      IN      A      142.251.167.106
www.google.com.                 40      IN      A      142.251.167.147











;; Query time: 69 msec
;; SERVER: 198.82.247.66#53(198.82.247.66) (UDP)
;; WHEN: Mon Oct 14 15:18:13 PDT 2024
;; MSG SIZE rcvd: 167











srirams@ada:~$


```

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





Geolocation data from		IP2Location	Product: DB6, 2024-9-15
 IP ADDRESS:	142.250.217.100	 ISP:	Google LLC
 COUNTRY:	United States 	 ORGANIZATION:	Not available
 REGION:	California	 LATITUDE:	37.4060
 CITY:	Mountain View	 LONGITUDE:	-122.0785
Incorrect location? Contact IP2Location		 view map	

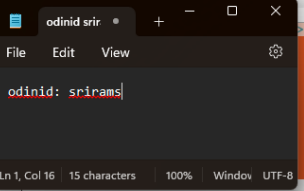
Geolocation data from		ipinfo.io	Product: API, real-time
 IP ADDRESS:	142.250.217.100	 ISP:	Not available
 COUNTRY:	United States 	 ORGANIZATION:	AS15169 Google LLC
 REGION:	Washington	 LATITUDE:	47.6062
 CITY:	Seattle	 LONGITUDE:	-122.3321
Incorrect location? Contact ipinfo.io		 view map	

Geolocation data from		DB-IP	Product: API, real-time
 IP ADDRESS:	142.250.217.100	 ISP:	Google LLC
 COUNTRY:	United States 	 ORGANIZATION:	Google LLC
 REGION:	Washington	 LATITUDE:	47.6061
 CITY:	Seattle	 LONGITUDE:	-122.333
Incorrect location? Contact DB-IP		 view map	




Popular Tools

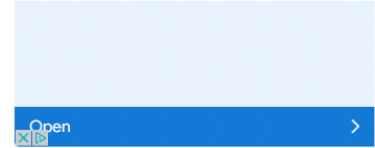
-  Domain Age Checker
-  Who is Hosting a Website
-  Is my website down?
-  Subnet Calculator







Ln 1, Col 16 15 characters 100% Window UTF-8










IP Address Articles



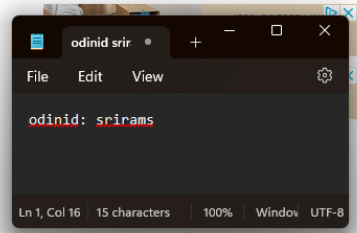








Popular Tools

-  Domain Age Checker
-  Who is Hosting a Website
-  Is my website down?
-  Subnet Calculator

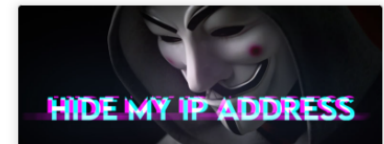
Geolocation data from		ipinfo.io	Product: API, real-time
	IP ADDRESS: 142.251.167.105		ISP: Not available
	COUNTRY: United States 		ORGANIZATION: AS15169 Google LLC
	REGION: Virginia		LATITUDE: 38.9687
	CITY: Reston		LONGITUDE: -77.3411

Advertisement



Geolocation data from		DB-IP	Product: API, real-time
	IP ADDRESS: 142.251.167.105		ISP: Google LLC
	COUNTRY: United States 		ORGANIZATION: Google LLC
	REGION: California		LATITUDE: 37.4225
	CITY: Mountain View		LONGITUDE: -122.085

IP Address Articles



- **Take a screenshot of the results for your lab notebook.**

```
srirams@ada:~$ 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)  0.770 ms  0.604 ms  0.589 ms
srirams@ada:~$
```

```

srirams@ada:~$ 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)  4.951 ms  4.887 ms  3.754 ms
 2 router.seas.pdx.edu (10.208.91.1)  0.120 ms  0.136 ms  0.091 ms
 3 CORE1.net.pdx.edu (131.252.5.142)  5.084 ms  5.022 ms  4.899 ms
 4 131.252.5.213 (131.252.5.213)  0.629 ms  0.546 ms  0.484 ms
 5 * e0-28.switch4.pdx1.he.net (216.218.230.89)  1.839 ms  1.835 ms
 6 port-channel2.core2.seal.he.net (184.105.64.137)  4.028 ms * *
 7 * * *
 8 * * *
 9 eqix-ash.vt.edu (206.126.236.139)  63.282 ms  64.206 ms  63.275 ms
10 192.70.187.20 (192.70.187.20)  68.907 ms  69.388 ms  68.754 ms
11 isb-core.xe-7-0-0.0.cns.vt.edu (128.173.0.202)  69.525 ms  69.536 ms  69.474 ms
12 cas-core.lo0.2000.cns.vt.edu (198.82.1.143)  69.585 ms  69.279 ms  69.214 ms
13 jeru.cns.vt.edu (198.82.247.66)  68.931 ms  68.954 ms  68.974 ms

```



```

srirams@ada:~$ traceroute 142.250.217.100
traceroute to 142.250.217.100 (142.250.217.100), 30 hops max, 60 byte packets
 1 glados.cat.pdx.edu (131.252.208.21) 22.429 ms 22.277 ms 10.252 ms
 2 router.seas.pdx.edu (10.208.91.1) 0.170 ms 0.103 ms 0.100 ms
 3 CORE1.net.pdx.edu (131.252.5.142) 4.027 ms 3.990 ms 3.929 ms
 4 131.252.5.213 (131.252.5.213) 0.507 ms 0.481 ms 0.657 ms
 5 google.nwax.net (198.32.195.34) 4.333 ms 9.782 ms 3.830 ms
 6 192.178.105.35 (192.178.105.35) 4.214 ms 4.173 ms 4.157 ms
 7 142.251.55.201 (142.251.55.201) 4.362 ms 4.139 ms 4.077 ms
 8 sea09s30-in-f4.1e100.net (142.250.217.100) 4.526 ms 4.603 ms 4.253 ms
srirams@ada:~$

```

```

srirams@ada:~$ traceroute 142.251.167.105
traceroute to 142.251.167.105 (142.251.167.105), 30 hops max, 60 byte packets
 1 glados.cat.pdx.edu (131.252.208.21) 8.531 ms 8.425 ms 8.357 ms
 2 router.seas.pdx.edu (10.208.91.1) 0.198 ms 0.102 ms 0.098 ms
 3 CORE1.net.pdx.edu (131.252.5.142) 3.514 ms 3.468 ms 3.374 ms
 4 131.252.5.213 (131.252.5.213) 0.686 ms 0.618 ms 0.551 ms
 5 google.nwax.net (198.32.195.34) 4.011 ms 3.737 ms 4.255 ms
 6 192.178.105.35 (192.178.105.35) 3.848 ms 108.170.255.175 (108.170.255.175) 5.414 ms 5.312 ms
 7 108.170.255.132 (108.170.255.132) 4.412 ms 108.170.255.194 (108.170.255.194) 5.596 ms 108.170.255.128 (108.170.255.128) 4.660 ms
 8 142.251.225.38 (142.251.225.38) 9.945 ms 216.239.41.34 (216.239.41.34) 10.775 ms 216.239.57.168 (216.239.57.168) 11.986 ms
 9 142.251.226.163 (142.251.226.163) 52.303 ms 142.251.226.161 (142.251.226.161) 52.528 ms 142.250.213.61 (142.250.213.61) 51.275 ms
10 192.178.81.224 (192.178.81.224) 64.657 ms 192.178.81.228 (192.178.81.228) 64.858 ms 192.178.81.238 (192.178.81.238) 65.507 ms
11 172.253.65.78 (172.253.65.78) 64.498 ms 142.250.238.219 (142.250.238.219) 66.215 ms 172.253.65.78 (172.253.65.78) 64.443 ms
12 142.250.238.7 (142.250.238.7) 65.681 ms 142.250.211.33 (142.250.211.33) 63.749 ms 142.250.238.5 (142.250.238.5) 65.490 ms
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 * * *
22 ww-in-f105.1e100.net (142.251.167.105) 63.554 ms * 63.912 ms
srirams@ada:~$

```

2.2.6 Wireshark Lab #3

In a terminal, using commands from prior labs, find the addresses and interfaces on the VM. Make a note of:

- The IP address of the VM – 10.138.0.2
- The name of the local virtual ethernet interface – ens4
- The IP address of the default router – 10.138.0.1

```

srirams@course-vm:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc mq state UP group default qlen 1000
    link/ether 42:01:0a:8a:00:02 brd ff:ff:ff:ff:ff:ff
    inet 10.138.0.2/32 metric 100 scope global dynamic ens4
        valid_lft 84029sec preferred_lft 84029sec
    inet6 fe80::4001:aff:fe8a:2/64 scope link
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:e7:99:0b:0b brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
srirams@course-vm:~$

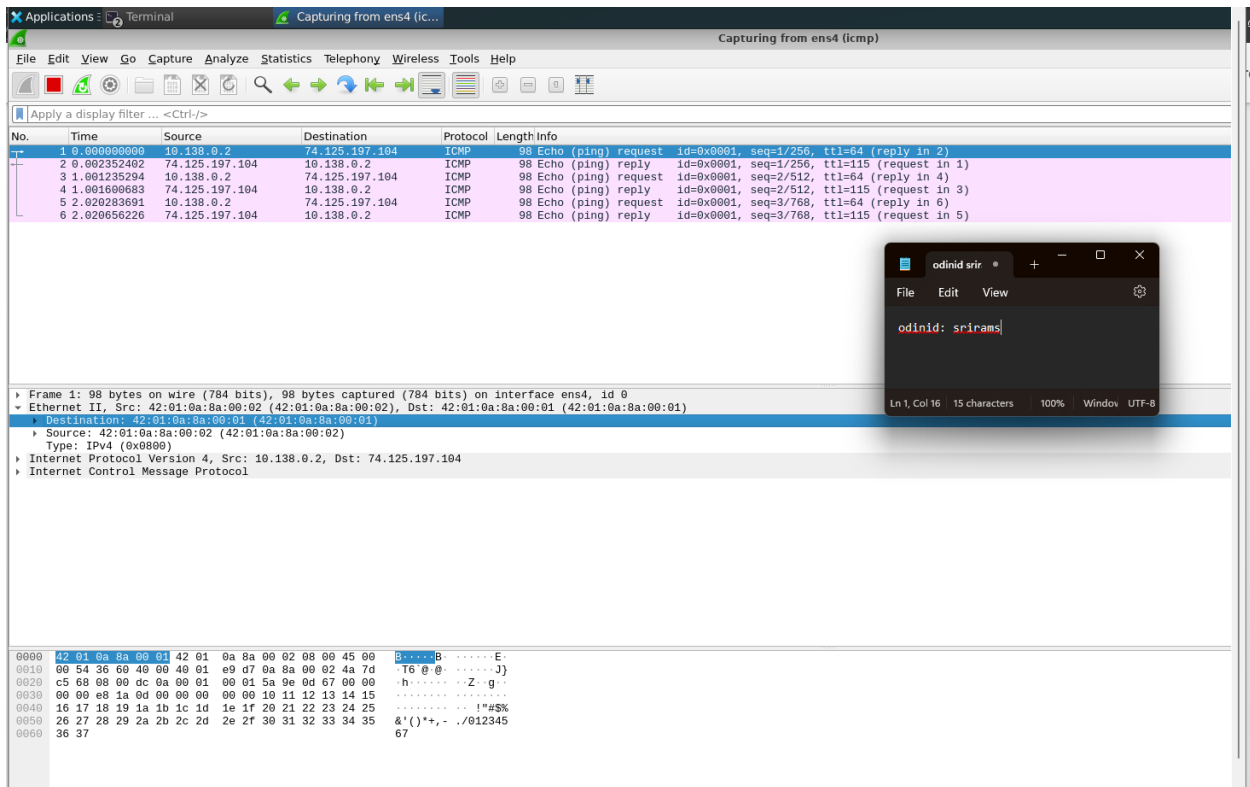
```

```

srirams@course-vm:~$ netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask         Flags         MSS Window  irtt Iface
0.0.0.0          10.138.0.1      0.0.0.0         UG            0 0        0 ens4
10.138.0.1       0.0.0.0         255.255.255.255 UH            0 0        0 ens4
169.254.169.254  10.138.0.1      255.255.255.255 UGH           0 0        0 ens4
172.17.0.0       0.0.0.0         255.255.0.0     U             0 0        0 docker0
srirams@course-vm:~$

```

2.2.7 -



- 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?

Yes the MAC address (42:01:0a:8a:00:01) corresponds to an interface on

Google's Website

Click on the next packet in the trace.

- 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?

The image shows a Wireshark packet capture of ICMP Echo (ping) traffic. The packet list shows several requests and replies between source 74.125.197.104 and destination 10.138.0.2. The packet details pane for the selected packet (No. 2) shows the following information:

- Frame 2: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface ens4, id 0
- Ethernet II, Src: 42:01:0a:8a:00:01 (42:01:0a:8a:00:01), Dst: 42:01:0a:8a:00:02 (42:01:0a:8a:00:02)
- Destination: 42:01:0a:8a:00:02 (42:01:0a:8a:00:02)
- Source: 42:01:0a:8a:00:01 (42:01:0a:8a:00:01)
- Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 74.125.197.104, Dst: 10.138.0.2
- 0100 = Version: 4
- 0101 = Header Length: 20 bytes (5)
- Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 84
- Identification: 0x0000 (0)
- Flags: 0x00
- ...0 0000 0000 0000 = Fragment Offset: 0
- Time to Live: 115
- Protocol: ICMP (1)
- Header Checksum: 0x2d38 [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 74.125.197.104
- Destination Address: 10.138.0.2
- Internet Control Message Protocol

The packet bytes pane shows the raw data in hexadecimal and ASCII. The ASCII column shows the string "srirams" in the payload, which is the OdinId used in the challenge.

The destination MAC address(42:01:0a:8a:00:02) corresponds to an interface on the VM “ens4

2.2.8 Network Recap Lab #4

- Find the IP address of <OdinId>.oregonctf.org, replacing <OdinId> with your OdinId

```
srirams@course-vm:~$ dig srirams.oregonctf.org

; <<>> DiG 9.18.28-0ubuntu0.22.04.1-Ubuntu <<>> srirams.oregonctf.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 13687
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;srirams.oregonctf.org.      IN      A

;; ANSWER SECTION:
srirams.oregonctf.org.  3600    IN      A      35.233.233.233

;; Query time: 87 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Oct 14 23:37:37 UTC 2024
;; MSG SIZE rcvd: 66

srirams@course-vm:~$ █
```

```
srirams@course-vm:~$ arp -an
? (10.138.0.1) at 42:01:0a:8a:00:01 [ether] on ens4
srirams@course-vm:~$ █
```

```
srirams@course-vm:~$ sudo arp -d 10.138.0.1; arp -an
srirams@course-vm:~$ arp -an
? (10.138.0.1) at 42:01:0a:8a:00:01 [ether] on ens4
srirams@course-vm:~$
```

2.2.9. Collect trace

```

srirams@course-vm:~$ sudo tcpdump -w network_trace
tcpdump: listening on ens4, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C854 packets captured
860 packets received by filter
0 packets dropped by kernel
srirams@course-vm:~$

```

```

srirams@course-vm:~$ sudo arp -d 10.138.0.1; wget http://srirams.oregonctf.org
--2024-10-14 23:54:45-- http://srirams.oregonctf.org/
Resolving srirams.oregonctf.org (srirams.oregonctf.org)... 35.233.233.233
Connecting to srirams.oregonctf.org (srirams.oregonctf.org)[35.233.233.233]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7524 (7.3K) [text/html]
Saving to: 'index.html.1'

index.html.1                                     100%[=====] 7.35K  --.-KB/s
2024-10-14 23:54:45 (545 MB/s) - 'index.html.1' saved [7524/7524]

srirams@course-vm:~$

```

10. Analyze trace

- Take a screenshot of the all of the packets returned within Wireshark that includes their packet numbers

No.	Time	Source	Destination	Protocol	Length	Info
35	2.347954	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
36	2.347973	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
213	8.347450	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
214	8.347473	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
281	14.347212	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
282	14.347228	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
411	20.347468	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
412	20.347485	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
572	26.347969	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
573	26.347927	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
640	32.348047	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
641	32.348065	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
647	33.156842	10.138.0.2	169.254.169.254	DNS	112	Standard query 0xe2fa AAAA course-vm.c.cloud-nurani-srirams.internal OPT
648	33.156951	10.138.0.2	169.254.169.254	DNS	112	Standard query response 0xe2fa AAAA course-vm.c.cloud-nurani-srirams.internal OPT
649	33.162127	169.254.169.254	10.138.0.2	DNS	128	Standard query response 0xee9c A course-vm.c.cloud-nurani-srirams.internal A 10.138.0.2 OPT
650	33.162127	169.254.169.254	10.138.0.2	DNS	201	Standard query response 0xe2fa AAAA course-vm.c.cloud-nurani-srirams.internal SOA ns.global.gcedns-prod.internal OPT
651	33.180313	42:01:0a:8a:00:02	Broadcast	ARP	42	Who has 10.138.0.1? Tell 10.138.0.2
652	33.182064	42:01:0a:8a:00:01	42:01:0a:8a:00:02	ARP	42	10.138.0.1 is at 42:01:0a:8a:00:01
653	33.182080	10.138.0.2	169.254.169.254	DNS	92	Standard query 0xbada AAAA srirams.oregonctf.org OPT
659	33.252401	169.254.169.254	10.138.0.2	DNS	174	Standard query response 0xbada AAAA srirams.oregonctf.org SOA ns-cloud-d1.googledomains.com OPT
660	33.252887	10.138.0.2	35.233.233.233	TCP	74	36876 -> 80 [SYN] Seq=0 Win=65320 Len=0 MSS=1420 SACK_PERM=1 TSval=2353199759 TSecr=0 WS=128
661	33.257811	35.233.233.233	10.138.0.2	TCP	74	80 -> 36876 [SYN, ACK] Seq=0 Ack=1 Win=64768 Len=0 MSS=1420 SACK_PERM=1 TSval=1129669199 TSecr=2353199759 WS=128
662	33.257858	10.138.0.2	35.233.233.233	TCP	66	36876 -> 80 [ACK] Seq=1 Ack=1 Win=65408 Len=0 TSval=2353199764 TSecr=1129669199
663	33.257985	10.138.0.2	35.233.233.233	HTTP	202	GET / HTTP/1.1
664	33.258662	35.233.233.233	10.138.0.2	TCP	66	80 -> 36876 [ACK] Seq=1 Ack=137 Win=64640 Len=0 TSval=1129669202 TSecr=2353199764
665	33.259107	35.233.233.233	10.138.0.2	TCP	5698	80 -> 36876 [ACK] Seq=1 Ack=137 Win=64640 Len=5632 TSval=1129669202 TSecr=2353199764 [TCP segment of a reassembled PDU]
666	33.259107	35.233.233.233	10.138.0.2	TCP	1474	80 -> 36876 [PSH, ACK] Seq=5633 Ack=137 Win=64640 Len=1408 TSval=1129669202 TSecr=2353199764 [TCP segment of a reassembled PDU]
667	33.259107	35.233.233.233	10.138.0.2	HTTP	799	HTTP/1.1 200 OK (text/html)
668	33.259139	10.138.0.2	35.233.233.233	TCP	66	36876 -> 80 [ACK] Seq=137 Ack=5633 Win=64128 Len=0 TSval=2353199765 TSecr=1129669202
669	33.259151	10.138.0.2	35.233.233.233	TCP	66	36876 -> 80 [ACK] Seq=137 Ack=7041 Win=64128 Len=0 TSval=2353199765 TSecr=1129669202
670	33.259156	10.138.0.2	35.233.233.233	TCP	66	36876 -> 80 [ACK] Seq=137 Ack=7774 Win=64128 Len=0 TSval=2353199765 TSecr=1129669202
672	33.260291	10.138.0.2	35.233.233.233	TCP	66	36876 -> 80 [FIN, ACK] Seq=137 Ack=7774 Win=64128 Len=0 TSval=2353199766 TSecr=1129669202
673	33.260691	35.233.233.233	10.138.0.2	TCP	66	80 -> 36876 [FIN, ACK] Seq=7774 Ack=138 Win=64640 Len=0 TSval=1129669204 TSecr=2353199766
674	33.260713	10.138.0.2	35.233.233.233	TCP	66	36876 -> 80 [ACK] Seq=138 Ack=7775 Win=64128 Len=0 TSval=2353199767 TSecr=1129669204
840	37.985388	10.138.0.2	169.254.169.254	HTTP	2254	HTTP/1.1 200 OK (text/html)
850	38.348060	42:01:0a:8a:00:01	Broadcast	ARP	281	GET /computeMetadata/v1/?recursive=true&wait_for_change=true&last_etag=86401fa6101f00f4&timeout_sec=60 HTTP/1.1
851	38.348078	42:01:0a:8a:00:02	42:01:0a:8a:00:01	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
				ARP	42	10.138.0.2 is at 42:01:0a:8a:00:02

ARP

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

35, 36

No.	Time	Source	Destination	Protocol	Length	Info
35	2.347954	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
36	2.347973	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
213	8.347450	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
214	8.347473	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
281	14.347212	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
282	14.347228	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
411	20.347468	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
412	20.347485	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
572	26.347969	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
573	26.347927	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
640	32.348047	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
641	32.348065	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

- What is this hardware address?

42:01:0a:8a:00:02

DNS

- What packet numbers in the trace correspond to the DNS request for the web site?

647, 648

[arp or dns or http or ip.addr]==35.233.233.233						
No.	Time	Source	Destination	Protocol	Length	Info
35	2.347954	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
36	2.347973	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
213	8.347450	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
214	8.347473	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
281	14.347212	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
282	14.347228	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
411	20.347468	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
412	20.347485	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
572	26.347909	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
573	26.347927	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
640	32.348047	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
641	32.348065	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
647	33.156842	10.138.0.2	169.254.169.254	DNS	112	Standard query 0xe9c A course-vm.c.cloud-nurani-srirams.internal OPT
648	33.156951	10.138.0.2	169.254.169.254	DNS	112	Standard query response 0xe2fa AAAA course-vm.c.cloud-nurani-srirams.internal OPT
649	33.162127	169.254.169.254	10.138.0.2	DNS	128	Standard query response 0xe9c A course-vm.c.cloud-nurani-srirams.internal A 10.138.0.2 OPT
650	33.162127	169.254.169.254	10.138.0.2	DNS	201	Standard query response 0xe2fa AAAA course-vm.c.cloud-nurani-srirams.internal SOA ns.global.gcedns-prod.internal OPT
651	33.189313	42:01:0a:8a:00:02	Broadcast	ARP	42	Who has 10.138.0.1? Tell 10.138.0.2
652	33.182064	42:01:0a:8a:00:01	42:01:0a:8a:00:02	ARP	42	10.138.0.1 is at 42:01:0a:8a:00:01
653	33.162080	10.138.0.2	169.254.169.254	DNS	92	Standard query 0x8ada AAAA srirams.oregonctf.org OPT
654	33.252401	169.254.169.254	10.138.0.2	DNS	174	Standard query response 0xe9c AAAA srirams.oregonctf.org SOA ns-cloud-dl.googleddomains.com OPT
660	33.252807	10.138.0.2	35.233.233.233	TCP	74	60876 -> 80 [SYN] Seq=0 Win=65520 Len=0 MSS=1420 SACK_PERM=1 TSval=2353199759 TSecr=0 WS=128
661	33.257811	35.233.233.233	10.138.0.2	TCP	74	80 -> 60876 [SYN, ACK] Seq=0 Ack=1 Win=64768 Len=0 MSS=1420 SACK_PERM=1 TSval=2353199759 TSecr=2353199759 WS=128
662	33.257858	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [ACK] Seq=1 Ack=1 Win=65408 Len=0 TSval=2353199764 TSecr=1129609199
663	33.257905	10.138.0.2	35.233.233.233	HTTP	202	GET / HTTP/1.1
664	33.258003	35.233.233.233	10.138.0.2	TCP	66	80 -> 60876 [ACK] Seq=1 Ack=137 Win=64640 Len=0 TSval=1129609202 TSecr=2353199764
665	33.259107	35.233.233.233	10.138.0.2	TCP	5698	80 -> 60876 [ACK] Seq=1 Ack=137 Win=64640 Len=5632 TSval=1129609202 TSecr=2353199764 [TCP segment of a reassembled PDU]
666	33.259107	35.233.233.233	10.138.0.2	TCP	1474	80 -> 60876 [PSH, ACK] Seq=5633 Ack=137 Win=64640 Len=1408 TSval=1129609202 TSecr=2353199764 [TCP segment of a reassembled PDU]
667	33.259107	35.233.233.233	10.138.0.2	HTTP	799	HTTP/1.1 200 OK (text/html)
668	33.259139	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [ACK] Seq=137 Ack=5633 Win=64128 Len=0 TSval=2353199765 TSecr=1129609202
669	33.259151	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [ACK] Seq=137 Ack=7041 Win=64128 Len=0 TSval=2353199765 TSecr=1129609202
670	33.259156	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [ACK] Seq=137 Ack=7774 Win=64128 Len=0 TSval=2353199765 TSecr=1129609202
672	33.260291	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [FIN, ACK] Seq=137 Ack=7774 Win=64128 Len=0 TSval=2353199766 TSecr=1129609202
673	33.260691	35.233.233.233	10.138.0.2	TCP	66	80 -> 60876 [FIN, ACK] Seq=7774 Ack=138 Win=64640 Len=0 TSval=1129609204 TSecr=2353199766
674	33.260713	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [ACK] Seq=138 Ack=7775 Win=64128 Len=0 TSval=2353199767 TSecr=1129609204
678	37.064040	169.254.169.254	10.138.0.2	HTTP/1.1	2257	[TCP ACKed unseen segment] HTTP/1.1 200 OK - JavaScript Object Notation (application/json)
840	37.985308	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?recursive=true&json&wait_for_change=true&last_etag=B6401fa6101f06f4&timeout_sec=60 HTTP/1.1
850	38.348060	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
851	38.348078	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

- 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?

660, 661, 662

[arp or dns or http or ip.addr]==35.233.233.233						
No.	Time	Source	Destination	Protocol	Length	Info
35	2.347954	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
36	2.347973	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
213	8.347450	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
214	8.347473	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
281	14.347212	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
282	14.347228	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
411	20.347468	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
412	20.347485	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
572	26.347909	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
573	26.347927	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
640	32.348047	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
641	32.348065	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
647	33.156842	10.138.0.2	169.254.169.254	DNS	112	Standard query 0xe9c A course-vm.c.cloud-nurani-srirams.internal OPT
648	33.156951	10.138.0.2	169.254.169.254	DNS	112	Standard query response 0xe2fa AAAA course-vm.c.cloud-nurani-srirams.internal OPT
649	33.162127	169.254.169.254	10.138.0.2	DNS	128	Standard query response 0xe9c A course-vm.c.cloud-nurani-srirams.internal A 10.138.0.2 OPT
650	33.162127	169.254.169.254	10.138.0.2	DNS	201	Standard query response 0xe2fa AAAA course-vm.c.cloud-nurani-srirams.internal SOA ns.global.gcedns-prod.internal OPT
651	33.189313	42:01:0a:8a:00:02	Broadcast	ARP	42	Who has 10.138.0.1? Tell 10.138.0.2
652	33.182064	42:01:0a:8a:00:01	42:01:0a:8a:00:02	ARP	42	10.138.0.1 is at 42:01:0a:8a:00:01
653	33.162080	10.138.0.2	169.254.169.254	DNS	92	Standard query 0x8ada AAAA srirams.oregonctf.org OPT
659	33.252401	169.254.169.254	10.138.0.2	DNS	174	Standard query response 0x8ada AAAA srirams.oregonctf.org SOA ns-cloud-dl.googleddomains.com OPT
660	33.252807	10.138.0.2	35.233.233.233	TCP	74	60876 -> 80 [SYN] Seq=0 Win=65520 Len=0 MSS=1420 SACK_PERM=1 TSval=2353199759 TSecr=0 WS=128
661	33.257811	35.233.233.233	10.138.0.2	TCP	74	80 -> 60876 [SYN, ACK] Seq=0 Ack=1 Win=64768 Len=0 MSS=1420 SACK_PERM=1 TSval=2353199759 TSecr=2353199759 WS=128
662	33.257858	10.138.0.2	35.233.233.233	HTTP	66	60876 -> 80 [ACK] Seq=1 Ack=1 Win=65408 Len=0 TSval=2353199764 TSecr=1129609199
663	33.257905	10.138.0.2	35.233.233.233	TCP	202	GET / HTTP/1.1
664	33.258003	35.233.233.233	10.138.0.2	TCP	66	80 -> 60876 [ACK] Seq=1 Ack=137 Win=64640 Len=0 TSval=1129609202 TSecr=2353199764
665	33.259107	35.233.233.233	10.138.0.2	TCP	5698	80 -> 60876 [ACK] Seq=1 Ack=137 Win=64640 Len=5632 TSval=1129609202 TSecr=2353199764 [TCP segment of a reassembled PDU]
666	33.259107	35.233.233.233	10.138.0.2	TCP	1474	80 -> 60876 [PSH, ACK] Seq=5633 Ack=137 Win=64640 Len=1408 TSval=1129609202 TSecr=2353199764 [TCP segment of a reassembled PDU]
667	33.259107	35.233.233.233	10.138.0.2	HTTP	799	HTTP/1.1 200 OK (text/html)
668	33.259139	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [ACK] Seq=137 Ack=5633 Win=64128 Len=0 TSval=2353199765 TSecr=1129609202
669	33.259151	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [ACK] Seq=137 Ack=7041 Win=64128 Len=0 TSval=2353199765 TSecr=1129609202
670	33.259156	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [ACK] Seq=137 Ack=7774 Win=64128 Len=0 TSval=2353199765 TSecr=1129609202
672	33.260291	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [FIN, ACK] Seq=137 Ack=7774 Win=64128 Len=0 TSval=2353199766 TSecr=1129609202
673	33.260691	35.233.233.233	10.138.0.2	TCP	66	80 -> 60876 [FIN, ACK] Seq=7774 Ack=138 Win=64640 Len=0 TSval=1129609204 TSecr=2353199766
674	33.260713	10.138.0.2	35.233.233.233	TCP	66	60876 -> 80 [ACK] Seq=138 Ack=7775 Win=64128 Len=0 TSval=2353199767 TSecr=1129609204
678	37.064040	169.254.169.254	10.138.0.2	HTTP/1.1	2257	[TCP ACKed unseen segment] HTTP/1.1 200 OK - JavaScript Object Notation (application/json)
840	37.985308	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?recursive=true&json&wait_for_change=true&last_etag=B6401fa6101f06f4&timeout_sec=60 HTTP/1.1
850	38.348060	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
851	38.348078	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

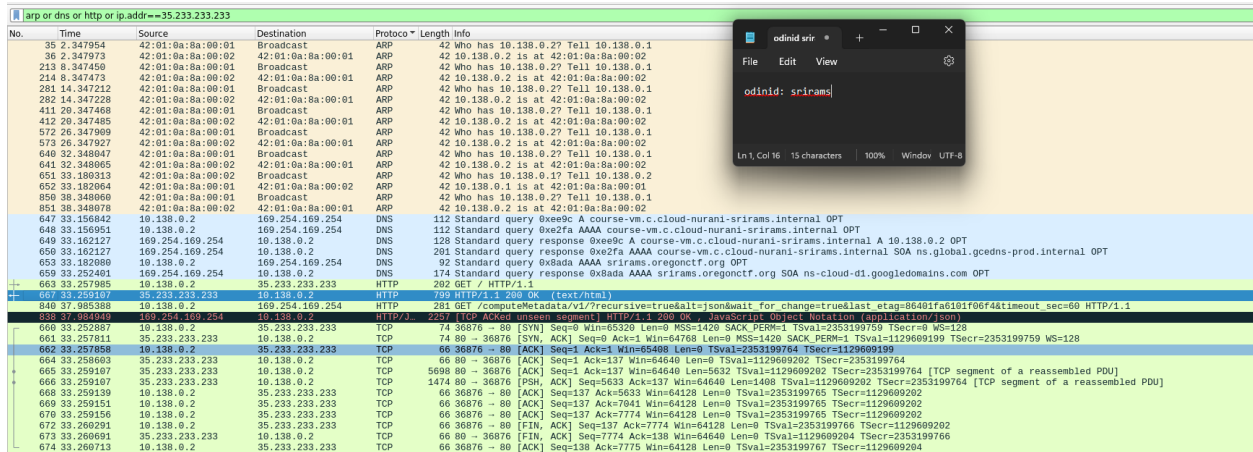
- How long does it take to perform the initial TCP handshake?

0.004971 seconds

HTTP

- What packet numbers in the trace correspond to the actual HTTP request and response?

663 (request), 667 (response)



No.	Time	Source	Destination	Protocol	Length	Info
35	2.347954	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
36	2.347973	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
213	8.347450	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
214	8.347473	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
281	14.347212	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
282	14.347228	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
411	20.347468	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
412	20.347485	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
572	26.347999	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
573	26.347927	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
640	32.348047	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
641	32.348005	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
651	33.180313	42:01:0a:8a:00:02	Broadcast	ARP	42	Who has 10.138.0.1? Tell 10.138.0.2
652	33.182064	42:01:0a:8a:00:01	42:01:0a:8a:00:02	ARP	42	10.138.0.1 is at 42:01:0a:8a:00:01
856	38.348060	42:01:0a:8a:00:01	Broadcast	ARP	42	Who has 10.138.0.2? Tell 10.138.0.1
851	38.348078	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
647	33.156842	10.138.0.2	169.254.169.254	DNS	112	Standard query 0xee0c A course-vm.c.cloud-nurani-srirams.internal OPT
648	33.156951	10.138.0.2	169.254.169.254	DNS	112	Standard query 0xe2fa AAAA course-vm.c.cloud-nurani-srirams.internal OPT
649	33.162127	169.254.169.254	10.138.0.2	DNS	128	Standard query response 0xee0c A course-vm.c.cloud-nurani-srirams.internal A 10.138.0.2 OPT
650	33.162127	169.254.169.254	10.138.0.2	DNS	201	Standard query response 0xe2fa AAAA course-vm.c.cloud-nurani-srirams.internal SOA ns.global.gcedns-prod.internal OPT
653	33.182080	10.138.0.2	169.254.169.254	DNS	92	Standard query 0xbada AAAA srirams.oregonctf.org OPT
659	33.252401	169.254.169.254	10.138.0.2	DNS	174	Standard query response 0xbada AAAA srirams.oregonctf.org SOA ns-cloud-d1.googledomains.com OPT
663	33.257965	10.138.0.2	35.233.233.233	HTTP	202	GET / HTTP/1.1
667	33.262917	35.233.233.233	10.138.0.2	HTTP	709	HTTP/1.1 200 OK (text/html)
840	37.985388	10.138.0.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?recursive=true&alt=json&wait_for_change=true&last_etag=86481fa6101f0674&timeout_sec=60 HTTP/1.1
888	37.985447	169.254.169.254	10.138.0.2	HTTP	2207	HTTP/1.1 200 OK (application/javascript)
660	33.252957	10.138.0.2	35.233.233.233	TCP	74	80 → 36876 [SYN] Seq=0 Win=0 Len=0 MSS=1420 SACK_PERM=1 TSval=2353199759 TSecr=0 WS=128
661	33.257811	35.233.233.233	10.138.0.2	TCP	74	80 → 36876 [SYN, ACK] Seq=0 Ack=1 Win=64768 Len=0 MSS=1420 SACK_PERM=1 TSval=2353199759 TSecr=2353199759 WS=128
662	33.257858	10.138.0.2	35.233.233.233	TCP	66	36876 → 80 [ACK] Seq=1 Ack=1 Win=64640 Len=0 TSval=2353199764 TSecr=1129609199
664	33.258663	35.233.233.233	10.138.0.2	TCP	66	80 → 36876 [ACK] Seq=1 Ack=137 Win=64640 Len=0 TSval=1129609202 TSecr=2353199764
665	33.259107	35.233.233.233	10.138.0.2	TCP	5698	80 → 36876 [ACK] Seq=1 Ack=137 Win=64640 Len=5632 TSval=1129609202 TSecr=2353199764 [TCP segment of a reassembled PDU]
666	33.259107	35.233.233.233	10.138.0.2	TCP	1474	80 → 36876 [PSH, ACK] Seq=5633 Ack=137 Win=64640 Len=1408 TSval=1129609202 TSecr=2353199764 [TCP segment of a reassembled PDU]
668	33.259139	10.138.0.2	35.233.233.233	TCP	66	36876 → 80 [ACK] Seq=137 Ack=5633 Win=64128 Len=0 TSval=2353199765 TSecr=1129609202
669	33.259151	10.138.0.2	35.233.233.233	TCP	66	36876 → 80 [ACK] Seq=137 Ack=7041 Win=64128 Len=0 TSval=2353199765 TSecr=1129609202
670	33.259156	10.138.0.2	35.233.233.233	TCP	66	36876 → 80 [ACK] Seq=137 Ack=7774 Win=64128 Len=0 TSval=2353199765 TSecr=1129609202
672	33.260201	10.138.0.2	35.233.233.233	TCP	66	36876 → 80 [FIN, ACK] Seq=137 Ack=7774 Win=64128 Len=0 TSval=2353199766 TSecr=1129609202
673	33.260691	35.233.233.233	10.138.0.2	TCP	66	80 → 36876 [FIN, ACK] Seq=7774 Ack=138 Win=64640 Len=0 TSval=1129609204 TSecr=2353199766
674	33.260713	10.138.0.2	35.233.233.233	TCP	66	36876 → 80 [ACK] Seq=138 Ack=7775 Win=64128 Len=0 TSval=2353199767 TSecr=1129609204

- How long does it take to process the HTTP request after the handshake?

0.001044 seconds