***Anatomy of a Web Connection: a Brief Analysis***

**Author: Rodrigo França Lima**

**Date: 20/03/2022**

**Index**

[**Objectives**](#_heading=h.30j0zll) **2**

[**Connection establishment**](#_heading=h.1fob9te) **2**

[Using traceroute tool to trace hops between a connection establishment](#_heading=h.3dy6vkm) 2

[Universidade de Aveiro’s network](#_heading=h.7tk0905qbgq9) 3

[Home connection](#_heading=h.zc8d1aqx0sy9) 3

[Tryhackme’s network](#_heading=h.ca0xcb5vrjq4) 4

[Result analysis](#_heading=h.1t3h5sf) 4

[Universidade de Aveiro’s network analysis](#_heading=h.dnn3t0k2rrhj) 4

[Home connection analysis](#_heading=h.840fwnxtb6ra) 5

[Tryhackme’s network analysis](#_heading=h.1t5w2gjx4ve2) 6

[**Exchange of data between connected peers**](#_heading=h.fhaww7zeatdv) **7**

[**Putting all together**](#_heading=h.2s8eyo1) **9**

[**Sources**](#_heading=h.frygu0a39l5s) **10**

# Objectives

The objective of this paper is to provide a basic knowledge of the steps and processes required to establish a connection between peers in a network and processes to exchange and manage those connections.

# Connection establishment

To better understand the steps and processes required on a connection establishment between peers we are going to proceed with a connection between multiple sources and [www.google.com](https://www.google.com/). The sources used are the following:

1. Campus Area Network (CAN) (Universidade de Aveiro’s network)
2. Local Area Network (LAN) (Home network)
3. Virtual Private Network (VPN) (Connection via VPN to a remote machine hosted by [tryhackme.com](https://tryhackme.com/))

## Using traceroute tool to trace hops between a connection establishment

Traceroute is a diagnostics tool that enables a user to know the involved hops and system’s ips on a connection establishment via the Control Message Protocol (ICMP) packets and their TTL (time to live).

To get a better view of the hops established between a connection it used the TCP (Transmission Control Protocol) with various sources as previously mentioned. The results are as follows:

### Universidade de Aveiro’s network

| > sudo traceroute -T www.google.com  traceroute to www.google.com (142.250.184.4), 30 hops max, 60 byte packets  1 gt2-edu-alunos.core.ua.pt (192.168.63.253) 1.322 ms 2.533 ms 2.496 ms  2 10.1.0.118 (10.1.0.118) 1.921 ms 1.892 ms 1.862 ms  3 nx2-ibgp.core.ua.pt (10.0.34.1) 3.041 ms Router41.Porto.fccn.pt (193.136.4.26) 3.969 ms 3.939 ms  4 Router41.Porto.fccn.pt (193.136.4.26) 3.854 ms Router40.Porto.fccn.pt (194.210.7.208) 7.274 ms Router41.Porto.fccn.pt (193.136.4.26) 4.465 ms  5 Router60.Lisboa.fccn.pt (193.136.1.10) 8.645 ms 7.184 ms Router40.Porto.fccn.pt (194.210.7.208) 7.140 ms  6 Router60.Lisboa.fccn.pt (193.136.1.10) 8.553 ms Router6.Lisboa.fccn.pt (194.210.6.105) 6.748 ms Router30.Lisboa.fccn.pt (193.136.1.8) 8.139 ms  7 Google.AS15169.gigapix.pt (193.136.250.20) 7.100 ms Router6.Lisboa.fccn.pt (194.210.6.205) 7.032 ms Google.AS15169.gigapix.pt (193.136.250.20) 6.994 ms  8 Google.AS15169.gigapix.pt (193.136.250.20) 7.872 ms 74.125.245.83 (74.125.245.83) 6.579 ms Google.AS15169.gigapix.pt (193.136.250.20) 7.808 ms  9 142.251.55.185 (142.251.55.185) 14.375 ms 15.825 ms 74.125.245.101 (74.125.245.101) 6.797 ms  10 142.251.55.185 (142.251.55.185) 14.691 ms 15.729 ms 142.251.55.149 (142.251.55.149) 17.506 ms  11 142.250.214.43 (142.250.214.43) 15.648 ms 142.250.214.41 (142.250.214.41) 19.279 ms 16.800 ms  12 142.250.214.41 (142.250.214.41) 16.726 ms 16.331 ms 15.063 ms  13 mad41s10-in-f4.1e100.net (142.250.184.4) 16.231 ms 142.250.214.43 (142.250.214.43) 16.851 ms mad41s10-in-f4.1e100.net (142.250.184.4) 15.082 ms  > |
| --- |

### Home connection

| > sudo traceroute -T www.google.com  traceroute to www.google.com (142.250.178.164), 30 hops max, 60 byte packets  1 \_gateway (192.168.0.1) 3.800 ms \* \*  2 \* \* \*  3 pa1-84-91-1-137.netvisao.pt (84.91.1.137) 43.915 ms 44.058 ms 45.000 ms  4 pa1-84-91-0-198.netvisao.pt (84.91.0.198) 53.028 ms pa1-84-91-0-193.netvisao.pt (84.91.0.193) 45.188 ms 52.872 ms  5 pa1-84-91-0-137.netvisao.pt (84.91.0.137) 54.246 ms 53.433 ms 54.230 ms  6 74.125.245.67 (74.125.245.67) 61.558 ms 57.753 ms 74.125.245.117 (74.125.245.117) 51.813 ms  7 142.251.55.185 (142.251.55.185) 58.136 ms 142.251.55.149 (142.251.55.149) 35.769 ms 29.100 ms  8 108.170.253.241 (108.170.253.241) 29.151 ms 216.239.47.124 (216.239.47.124) 36.316 ms 108.170.253.225 (108.170.253.225) 29.500 ms  9 142.251.54.155 (142.251.54.155) 37.022 ms 108.170.253.225 (108.170.253.225) 36.365 ms 142.251.54.153 (142.251.54.153) 28.366 ms  10 142.251.54.153 (142.251.54.153) 36.394 ms 142.251.54.155 (142.251.54.155) 38.406 ms mad41s08-in-f4.1e100.net (142.250.178.164) 37.661 ms  > |
| --- |

### Tryhackme’s network

| > traceroute -T www.google.com  traceroute to www.google.com (74.125.193.103), 30 hops max, 60 byte packets  1 \* \* \*  2 \* \* \*  3 \* \* \*  4 \* 241.0.9.140 (241.0.9.140) 0.349 ms \*  5 241.0.9.137 (241.0.9.137) 0.365 ms 240.1.88.30 (240.1.88.30) 0.397 ms 241.0.9.128 (241.0.9.128) 0.444 ms  6 240.1.88.31 (240.1.88.31) 0.439 ms 240.1.88.23 (240.1.88.23) 0.297 ms 240.1.88.31 (240.1.88.31) 0.255 ms  7 100.95.18.159 (100.95.18.159) 1.306 ms 242.3.160.1 (242.3.160.1) 0.961 ms 242.3.160.17 (242.3.160.17) 0.253 ms  8 100.95.18.153 (100.95.18.153) 0.758 ms 100.95.18.147 (100.95.18.147) 1.430 ms 100.95.18.149 (100.95.18.149) 1.290 ms  9 100.100.16.68 (100.100.16.68) 0.380 ms 100.100.16.42 (100.100.16.42) 0.577 ms 100.100.16.26 (100.100.16.26) 7.880 ms  10 99.82.176.25 (99.82.176.25) 1.302 ms 216.239.43.3 (216.239.43.3) 1.456 ms 100.95.20.96 (100.95.20.96) 1.663 ms  11 100.100.2.12 (100.100.2.12) 1.051 ms 74.125.244.1 (74.125.244.1) 2.301 ms 74.125.243.248 (74.125.243.248) 1.528 ms  12 74.125.244.7 (74.125.244.7) 2.024 ms 74.125.243.216 (74.125.243.216) 1.920 ms 99.82.176.189 (99.82.176.189) 1.540 ms  13 216.239.46.157 (216.239.46.157) 2.443 ms 216.239.43.3 (216.239.43.3) 1.353 ms 74.125.244.7 (74.125.244.7) 1.976 ms  14 74.125.243.215 (74.125.243.215) 3.715 ms 74.125.243.232 (74.125.243.232) 1.936 ms 216.239.40.163 (216.239.40.163) 1.411 ms  15 209.85.240.77 (209.85.240.77) 1.615 ms 1.409 ms 216.239.46.157 (216.239.46.157) 2.259 ms  16 \* \* \*  17 \* \* \*  18 \* \* \*  19 \* \* \*  20 ig-in-f103.1e100.net (74.125.193.103) 1.665 ms \* \*  > |
| --- |

## Result analysis

### Universidade de Aveiro’s network analysis

The traceroute made from the UA’s (Universidade de Aveiro) network can be interpreted as follows:

1. Firstly we try to check the DNS records of the given domain inside of the UA’s network first, this is made by the following hop: gt2-edu-alunos.core.ua.pt (192.168.63.253) .
2. After knowing that the request is for a peer outside of the UA’s network we are rerouted inside of UA’s network by the hop: 10.1.0.118 (10.1.0.118) .
3. In the next hop we are going to be routed outside UA’s network but still inside of the educational network provided by [fccn.pt](http://fccn.pt). This routing is provided by the following hop: nx2-ibgp.core.ua.pt (10.0.34.1) 3.041 ms Router41.Porto.fccn.pt (193.136.4.26).
4. After UP’s (Universidade do Porto) network we are going to be routed to UL’s (Universidade de Lisboa) network, this is made by the following hop: Router60.Lisboa.fccn.pt (193.136.1.10) Router40.Porto.fccn.pt (194.210.7.208) .
5. The following hop is just a hop inside UL’s network: Router60.Lisboa.fccn.pt (193.136.1.10) Router6.Lisboa.fccn.pt (194.210.6.105) Router30.Lisboa.fccn.pt (193.136.1.8) .
6. After the internal network hop we hit a hop that routes us to an exchange point, this exchange point is [gigapix.pt](http://gigapix.pt) and it provides a physical space for ISP (Internet Service Providers) and CDN (content delivery network) to connect with each other. The hop is the following: Google.AS15169.gigapix.pt (193.136.250.20) Router6.Lisboa.fccn.pt (194.210.6.205) Google.AS15169.gigapix.pt (193.136.250.20) .
7. Now that we hit an exchange point we are routed to a google’s router and the following hops are just internal google’s network hops. The hops are as follow: Google.AS15169.gigapix.pt (193.136.250.20) 74.125.245.83 (74.125.245.83) Google.AS15169.gigapix.pt (193.136.250.20) 142.251.55.185 (142.251.55.185) 74.125.245.101 (74.125.245.101) 142.251.55.185 (142.251.55.185) 142.251.55.149 (142.251.55.149) 142.250.214.43 (142.250.214.43) 142.250.214.41 (142.250.214.41) 142.250.214.41 (142.250.214.41) mad41s10-in-f4.1e100.net (142.250.184.4) 16.231 ms 142.250.214.43 (142.250.214.43) 16.851 ms mad41s10-in-f4.1e100.net (142.250.184.4)

### Home connection analysis

The traceroute made from the author’s home network can be interpreted as follows:

1. The request first goes through the gateway, this is described by the hop: \_gateway (192.168.0.1) .
2. The following hop doesn't have any response, this can be due to the priority the request has in the router's configuration, or the TTL (time to live) was exceeded due to some reason or if the router is configured to ignore these types of requests. This hop is represented as: \* \* \* .
3. The next hop is the routing to the ISP used by the author, and the following two hops are internal ISP’s network hops. The hops are the following: pa1-84-91-1-137.netvisao.pt (84.91.1.137) pa1-84-91-0-198.netvisao.pt (84.91.0.198) pa1-84-91-0-193.netvisao.pt (84.91.0.193) pa1-84-91-0-137.netvisao.pt (84.91.0.137) .
4. Finally in the next hops we enter google’s network and after some internal routings we get our connection: 74.125.245.67 (74.125.245.67) 74.125.245.117 (74.125.245.117) 142.251.55.185 (142.251.55.185) 142.251.55.149 (142.251.55.149) 108.170.253.241 (108.170.253.241) 216.239.47.124 (216.239.47.124) 108.170.253.225 (108.170.253.225) 142.251.54.155 (142.251.54.155) 108.170.253.225 (108.170.253.225) 142.251.54.153 (142.251.54.153) 142.251.54.153 (142.251.54.153) 142.251.54.155 (142.251.54.155) mad41s08-in-f4.1e100.net (142.250.178.164)

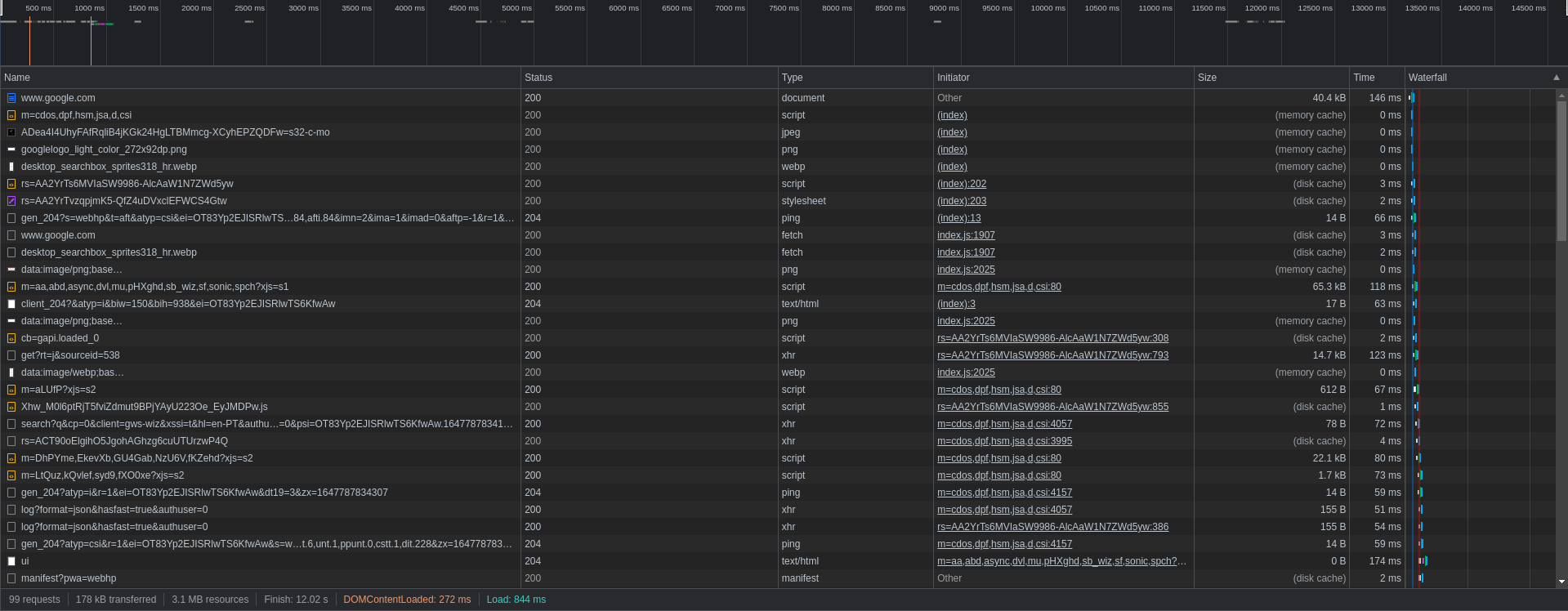
### Tryhackme’s network analysis

The VPN connection between the tryhackme’s machine and the author's machine won't be represented.

1. The first three hops didn't respond to the request but later there were various hops inside tryhackme’s network, the hops are the following:  
   \* \* \*  
   \* \* \*  
   \* \* \*  
   \* 241.0.9.140 (241.0.9.140) 241.0.9.137 (241.0.9.137) 240.1.88.30 (240.1.88.30) 241.0.9.128 (241.0.9.128) 240.1.88.31 (240.1.88.31) 240.1.88.23 (240.1.88.23) 240.1.88.31 (240.1.88.31) 100.95.18.159 (100.95.18.159) 242.3.160.1 (242.3.160.1) 242.3.160.17 (242.3.160.17) 100.95.18.153 (100.95.18.153) 100.95.18.147 (100.95.18.147) 100.95.18.149 (100.95.18.149) 100.100.16.68 (100.100.16.68) 100.100.16.42 (100.100.16.42) 100.100.16.26 (100.100.16.26)`
2. After internal hops we arrive at amazon’s network, this is due to tryhackme’s network being hosted by AWS ([**Amazon Web Services**](https://aws.amazon.com/)). The hop is the following: 99.82.176.25 (99.82.176.25) 216.239.43.3 (216.239.43.3) 100.95.20.96 (100.95.20.96).
3. The following hop is just internal routing inside amazon’s network: 100.100.2.12 (100.100.2.12) 74.125.244.1 (74.125.244.1) 74.125.243.248 (74.125.243.248).
4. After internal routing we get routed into google’s network by the following hop: 74.125.244.7 (74.125.244.7) 74.125.243.216 (74.125.243.216) 99.82.176.189 (99.82.176.189).
5. The following hops are internal routings inside google’s network until we have a connections established: `216.239.46.157 (216.239.46.157) 216.239.43.3 (216.239.43.3) 74.125.244.7 (74.125.244.7) 74.125.243.215 (74.125.243.215) 74.125.243.232 (74.125.243.232) 216.239.40.163 (216.239.40.163) 209.85.240.77 (209.85.240.77) 216.239.46.157 (216.239.46.157)  
   \* \* \*  
   \* \* \*  
   \* \* \*   
   \* \* \*  
   [ig-in-f103.1e100.net](http://ig-in-f103.1e100.net) (74.125.193.103) \* \*

# Exchange of data between connected peers

Now that a connection is established the transition of information between two connected peers doesn't stopt. In the previous example the connection was between us and the webpage, [www.google.com](https://www.google.com/). When we have a connection established in a webpage we require to load multiple resources for the web page to work properly. Our browser will try to connect and request those resources in the same way we tried to connect to the webpage. The communications can be seen by a debugger. If we try to record the communications between our browser and the resources host’s for the webpage [www.google.com](https://www.google.com/) we get the following:



These resources can vary between essential ones or just trackers and communication methods between companies to share user information. These requests also rely on specific requests that may also relay more user info to the resource hosts.

We can inspect those requests also with the help of the debugger provided by the browser as seen below:

| Host: www.google.com  User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:98.0) Gecko/20100101 Firefox/98.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  Accept-Language: en-US,en;q=0.5  Accept-Encoding: gzip, deflate, br  DNT: 1  Alt-Used: www.google.com  Connection: keep-alive  Cookie: CONSENT=YES+srp.gws-20211006-0-RC1.pt-PT+FX+700; NID=511=HoKWdyyhDRHr0U0d4Gyu4vnDNUVWl\_x25LlubTCqigrmxGMIIU7z5IC\_Lf85m9XwLX8tBIc\_N0\_IfvEkpEt-bpEQ40jXrdzkcxnV54OWKSikdpPgjJlx0guViyJkX5RIc1dZuDZc\_FrOvsv5t\_SjIU8vBH\_lWGpqRU-wzE\_dyfa3e4pTBbgo95ozp5uk1yCOjB9tIioC-rLnfEGHGxZFkcOvBLlSf\_XVgkSLXyhh17ob4\_hDWmbSH4wjaEwVYytVWW\_rCw\_2HxpDyg; 1P\_JAR=2022-03-20-14; ANID=AHWqTUljymLmqfCRCwvxM3ZWJui3KOfTMcW4P3NP0FJc5Zx827nxym\_OS530N7HE; SID=HQjEGmePlAxx-ZOPh6Qs5tRV9jAh8HRw0nx1pfvTfzG2vIX6dVrCRjf3K13Jh\_uK5WSqRw.; \_\_Secure-1PSID=HQjEGmePlAxx-ZOPh6Qs5tRV9jAh8HRw0nx1pfvTfzG2vIX6JcBsI0MVyob01VyY\_9JCjg.; \_\_Secure-3PSID=HQjEGmePlAxx-ZOPh6Qs5tRV9jAh8HRw0nx1pfvTfzG2vIX69vQmYKXXlTMnM7Ythv1VGA.; HSID=ADZ0j-MALTVYWzJQP; SSID=A1iVUaEoG7n4Xj8pB; APISID=SDS2jd-Rt4ohY6Cp/ANq\_UpEMJlzXs-8tM; SAPISID=JZPpl\_coK7Nn-UJB/ADmbv6Np7vNMwrH4z; \_\_Secure-1PAPISID=JZPpl\_coK7Nn-UJB/ADmbv6Np7vNMwrH4z; \_\_Secure-3PAPISID=JZPpl\_coK7Nn-UJB/ADmbv6Np7vNMwrH4z; SIDCC=AJi4QfG9UJ-3COgeREtAGJ-IezUCYbwKWPuSRwJn4oYYnKpsVa9tof1M-gS7jCGMxLIncHbutw; \_\_Secure-3PSIDCC=AJi4QfF9qaKD9HBNqbkBqKk7Rf4FHrEqD-Zc7HaozOnwM6cRhHqWMS9gpOgISXkX6XxzNMMqQ2k; SEARCH\_SAMESITE=CgQI4ZMB; S=billing-ui-v3=-RgQQ5PY\_9H7exfCo\_zVk4WJAg9NCZwh:billing-ui-v3-efe=-RgQQ5PY\_9H7exfCo\_zVk4WJAg9NCZwh  Upgrade-Insecure-Requests: 1  Sec-Fetch-Dest: document  Sec-Fetch-Mode: navigate  Sec-Fetch-Site: none  Sec-Fetch-User: ?1  Cache-Control: max-age=0  TE: trailers |
| --- |

In the request shown previously we can observe that the request shares that our host machine is linux x86 64 bits and that we are using the firefox browser by the user-agent, more information can be shared via these requests, for example via cookies. The information shared via these requests enable companies to fingerprint the hosts machines, for example by the type of extensions a user has, the choices in settings or even the size of the screen a company can generate an unique id for each host and enable tracking even if tracking blockers are used.

Although companies can spy on the users, users can also take a peak to the companies technologies used. Either by checking static code or by analyzing the requests made by a webpage a user can discover multiple dependencies that enable them to know more about the technologies used by a certain webpage and a certain company. One simple way to verify those technologies is by using the [wappalyzer](http://wappalyzer.com/) browser extension. For example, by visiting the University of Aveiro’s [webpage](https://www.ua.pt/) we get the following results:



# Putting all together

With how a peer connects to another and how a connected peers requests resources with others let’s review what it takes to request a webpage:

1. Firstly the browser issues a request for the webpage tryhackme.com.



1. The host machine that requested the webpage checks the local cache for the IP Address.



1. If the IP Address isn’t found on the host’s cache, the check is made recursively to the host’s DNS Server for Address.



1. After a recursive check, the host queries the root server to find an authoritative DNS Server.



1. After an authoritative DNS Server is found the server advises the IP address for the website the host requested.



1. After the host has the IP address the request passes through a Web Application Firewall (WAF)



1. After the WAF the request is redirected through a Load Balancer so that the webpage host’s machine doesn’t get overwhelmed by requests.



1. After the request goes through the load balancer, we connect to the webserver on the port 80 or 443 (it may vary due to http or https), we can also connect to other ports but those will not be default ones.



1. The web server then receives the GET request for the webpage we requested.



1. The Web Application talks to the Database so that it can retrieve the required content.



1. Our browser renders the HTML requested by it.



# Sources

<https://tryhackme.com/>  
<https://www.geolocation.com/>