

```
[ec2-user@ip-172-31-41-0 ~]$ ip a
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 noprefixroute
        valid_lft forever preferred_lft forever
2: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 0e:36:34:fc:e9:7f brd ff:ff:ff:ff:ff:ff
    altname enp0s5
    altname enx0e3634fce97f
    inet 172.31.41.0/20 brd 172.31.47.255 scope global dynamic noprefixroute ens5
        valid_lft 2092sec preferred_lft 2092sec
    inet6 fe80::c36:34ff:fefc:e97f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Loopback Interface (**lo**)

```
1: lo: <LOOPBACK,UP,LOWER_UP> ...
    link/loopback 00:00:00:00:00:00 ...
    inet 127.0.0.1/8 ...
    inet6 ::1/128 ...
```

- **Interface **lo****: This is the loopback interface, used for internal communication within the host.
- **IPv4 address**: **127.0.0.1/8** – the loopback IP (localhost).
- **IPv6 address**: **::1/128** – the IPv6 equivalent of localhost.
- **MAC address**: **00:00:00:00:00:00** (not used for loopback).
- **State**: **UNKNOWN** – this is normal for loopback.

Network Interface (**ens5**)

```
2: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> ...
    link/ether 0e:36:34:fc:e9:7f ...
    inet 172.31.41.0/20 ...
    inet6 fe80::c36:34ff:fefc:e97f/64 ...
```

- **Interface `ens5`:** This is your primary network interface (connected to the outside network).
 - **MAC address:** `0e:36:34:fc:e9:7f`
 - **IPv4 address:** `172.31.41.0/20` – a private IP, dynamically assigned.
 - This is part of AWS EC2's private IP range.
 - The `/20` indicates a subnet mask of `255.255.240.0`.
 - **Broadcast address:** `172.31.47.255`
 - **IPv6 address:** `fe80::c36:34ff:fe9c:e97f/64`
 - This is a **link-local** address, used only within the local network segment.
 - **State:** `UP` – interface is active and ready to send/receive data.
-

Summary

- The system has two network interfaces:
 1. **lo:** Internal loopback.
 2. **ens5:** External interface with an active private IPv4 and a link-local IPv6 address.
 - Your system is connected and has been assigned the IP `172.31.41.0`.
-

These `ping` command results show that your EC2 instance has **working internet connectivity** and **DNS resolution is functioning correctly**. Here's what each part means:

```
[ec2-user@ip-172-31-41-0 ~]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=0.890 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=0.904 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=0.905 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2075ms
rtt min/avg/max/mdev = 0.890/0.899/0.905/0.006 ms
[ec2-user@ip-172-31-41-0 ~]$ ping google.com
PING google.com (192.178.218.100) 56(84) bytes of data.
64 bytes from yuiadtq-in-f100.1e100.net (192.178.218.100): icmp_seq=1 ttl=106 time=1.94 ms
64 bytes from yuiadtq-in-f100.1e100.net (192.178.218.100): icmp_seq=2 ttl=106 time=1.98 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.941/1.961/1.982/0.020 ms
```

✓ ping 8.8.8.8

This tests **basic network connectivity** to Google's public DNS server (IP: 8.8.8.8), without involving DNS name resolution.

Output:

```
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=0.890 ms
...
3 packets transmitted, 3 received, 0% packet loss
rtt min/avg/max/mdev = 0.890/0.899/0.905/0.006 ms
```

What this tells you:

- Your instance is able to reach the internet.
 - No packet loss occurred.
 - Round-trip time (RTT) is very low (under 1ms), which is normal for a well-connected cloud server.
 - TTL = 117 indicates how many hops are left (not particularly important unless you're tracing routes).
-

✓ ping google.com

This tests both:

1. **DNS resolution** (turning `google.com` into an IP).
2. **Network connectivity** to the resolved IP.

Output:

```
PING google.com (192.178.218.100)  
64 bytes from yuiadtq-in-f100.1e100.net ...
```

What this tells you:

- Your system successfully resolved `google.com` to `192.178.218.100`, so DNS is working.
 - The actual server it reached is `yuiadtq-in-f100.1e100.net` (a Google frontend).
 - RTT is also very low (~2 ms), indicating a fast response and good network health.
 - Again, no packet loss.
-

```
[ec2-user@ip-172-31-41-0 ~]$ nmcli dev status
DEVICE  TYPE      STATE      CONNECTION
ens5    ethernet  connected  cloud-init ens5
lo      loopback  connected (externally)  lo
```

● What It Means:

1. **ens5**

- **TYPE:** **ethernet** – This is your main network interface (physical or virtual).
- **STATE:** **connected** – The interface is up and has a valid connection.
- **CONNECTION:** **cloud-init ens5** – This connection was created/managed by **cloud-init** (which is typical for cloud environments like AWS EC2).

2. **lo**

- **TYPE:** **loopback** – Internal interface used by the system to talk to itself.
 - **STATE:** **connected (externally)** – It's connected but not managed by NetworkManager; it's part of the system by default.
 - **CONNECTION:** **lo** – Name of the loopback connection.
-

```
[ec2-user@ip-172-31-41-0 ~]$ ip route
default via 172.31.32.1 dev ens5 proto dhcp src 172.31.41.0 metric 100
172.31.32.0/20 dev ens5 proto kernel scope link src 172.31.41.0 metric 100
```

1. Default Route (Internet Access)

```
default via 172.31.32.1 dev ens5 proto dhcp src 172.31.41.0 metric 100
```

- **default**: This is the default route (used for all destinations not explicitly listed elsewhere).
- **via 172.31.32.1**: This is the **gateway IP** — traffic to outside networks (like the internet) is sent here first.
- **dev ens5**: This route uses the **ens5** network interface.
- **proto dhcp**: This route was provided by DHCP (dynamic IP configuration).
- **src 172.31.41.0**: This is your instance's IP used as the source when sending traffic.
- **metric 100**: A value used to determine route priority (lower is preferred; 100 is standard for DHCP).

2. Local Subnet Route

```
172.31.32.0/20 dev ens5 proto kernel scope link src 172.31.41.0 metric 100
```

- **172.31.32.0/20**: This is your **local subnet** — includes IPs from **172.31.32.0** to **172.31.47.255**.
- **dev ens5**: Traffic to this subnet is sent via the same **ens5** interface.
- **proto kernel**: Added automatically by the Linux kernel.
- **scope link**: Route is directly reachable (no gateway needed).
- **src 172.31.41.0**: Your source IP when sending to this subnet.

```
[ec2-user@ip-172-31-41-0 ~]$ ss -tln
Netid      State      Recv-Q      Send-Q      Local Address:Port
udp        UNCONN     0            0            127.0.0.1:323
udp        UNCONN     0            0            [::1]:323
tcp        LISTEN     0            128          0.0.0.0:22
tcp        LISTEN     0            128          [::]:22
```

Netid: Protocol used (**tcp** or **udp**)

State: Connection state (**LISTEN** for TCP servers, **UNCONN** for UDP)

Recv-Q / Send-Q: Queues for received/sent data (usually 0 if idle)

Local Address:Port: The IP and port the service is bound to

Port **22 (SSH)** is open and listening for connections on **both IPv4 and IPv6**.

Port **323 (UDP)** is used by the NTP daemon (**chronyd**) for time synchronization, listening only on localhost (**127.0.0.1** and **::1**).

```
[ec2-user@ip-172-31-41-0 ~]$ ifconfig
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.41.0 netmask 255.255.240.0 broadcast 172.31.47.255
    inet6 fe80::c36:34ff:fe9c:e97f prefixlen 64 scopeid 0x20<link>
    ether 0e:36:34:fc:e9:7f txqueuelen 1000 (Ethernet)
    RX packets 27446 bytes 24310389 (23.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11564 bytes 1271691 (1.2 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 32 bytes 2616 (2.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 32 bytes 2616 (2.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Interface **ens5** (Your main network interface)

- **Flags:** **UP**, **RUNNING**, **BROADCAST**, **MULTICAST**
 - The interface is active and ready to send/receive packets.
- **MTU:** 9001
 - This is a "jumbo frame" setting, common in AWS (default is 1500).
- **IPv4 Address:** **172.31.41.0**
 - This is your EC2 instance's private IP.
- **Netmask:** **255.255.240.0**
 - This is a **/20** subnet, covering **172.31.32.0** to **172.31.47.255**.
- **Broadcast:** **172.31.47.255**
 - Used for sending packets to all devices in the subnet.
- **IPv6 Address:** **fe80::c36:34ff:fe9c:e97f**
 - A link-local address used within the local network.

- **MAC Address:** `0e:36:34:fc:e9:7f`
 - **Traffic Stats:**
 - **RX (Received):** 27,446 packets, ~23.1 MiB
 - **TX (Transmitted):** 11,564 packets, ~1.2 MiB
 - **No errors, no dropped packets, and no collisions** — this is good.
-

Interface `lo` (Loopback interface)

- **IPv4:** `127.0.0.1` – Localhost
 - **Netmask:** `255.0.0.0`
 - **IPv6:** `::1` – IPv6 localhost
 - **Traffic Stats:**
 - Very minimal RX/TX activity (normal)
 - No errors
-

```
[ec2-user@ip-172-31-41-0 ~]$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      504 ip-172-31-41-0.ec2.:ssh 203.81.241.120:11292    ESTABLISHED
udp        0      0 ip-172-31-41-0.e:bootpc ip-172-31-32-1.e:bootps ESTABLISHED
```

`tcp 0 504 ip-172-31-41-0.ec2.:ssh 203.81.241.120:11292 ESTABLISHED`

- Protocol: TCP
- Local Port: ssh (port 22)
- Remote Address: 203.81.241.120 (likely your own IP or whoever is connected)
- State: ESTABLISHED – this means an active SSH session is open (someone is logged into the EC2 instance).

`udp 0 0 ip-172-31-41-0.e:bootpc ip-172-31-32-1.e:bootps ESTABLISHED`

- bootpc → bootps: This is part of the DHCP communication (port 68 → 67).
- Indicates that your instance has established a DHCP lease via UDP.

UNIX Domain Sockets (UDS) are a form of **inter-process communication (IPC)** used in Unix-like operating systems (like Linux). They allow **local processes** (programs running on the same machine) to **communicate with each other** efficiently — kind of like a local network connection, but entirely inside the system.

They are faster and more secure than using network sockets over `localhost`.

Types of UNIX Domain Sockets

1. **Stream (STREAM)**: Like TCP — provides a reliable, ordered connection between two processes.
2. **Datagram (DGRAM)**: Like UDP — messages are sent without establishing a connection.

You'll often see them under paths like:

- `/run/dbus/system_bus_socket` → used by D-Bus for messaging between system components
- `/run/systemd/journal/socket` → used by journald for logging
`/run/chrony/chronyd.sock` → for time synchronization
- `/run/systemd/notify` → for internal communication for systemd

```
[ec2-user@ip-172-31-41-0 ~]$ netstat -rn
Kernel IP routing table
Destination      Gateway         Genmask         Flags         MSS Window  irtt Iface
0.0.0.0          172.31.32.1    0.0.0.0         UG            0  0           0 ens5
172.31.32.0      0.0.0.0        255.255.240.0   U             0  0           0 ens5
```

1. Default Route (Internet Gateway)

- **Destination:** `0.0.0.0` – This is the **default route**, meaning "any destination not explicitly listed."
- **Gateway:** `172.31.32.1` – This is the next hop (likely your **VPC's internet gateway**).
- **Genmask:** `0.0.0.0` – Matches any IP address.
- **Flags:**
 - `U` – Route is up
 - `G` – Route goes through a gateway
- **Iface:** `ens5` – This route uses the `ens5` network interface.

2. Local Subnet Route

- **Destination:** 172.31.32.0
- **Genmask:** 255.255.240.0 – A /20 subnet (range: 172.31.32.0 – 172.31.47.255)
- **Gateway:** 0.0.0.0 – Means it's a **directly connected** route (no gateway needed).
Flags: U – Route is up
- **Iface:** ens5

Your EC2 instance routes **internet-bound traffic** through 172.31.32.1. It also has a direct route to its local VPC subnet (172.31.32.0/20). All routing is handled through the interface ens5.

```
[ec2-user@ip-172-31-41-0 ~]$ traceroute google.com
traceroute to google.com (172.253.122.101), 30 hops max, 60 byte packets
 1 100.100.32.62 (100.100.32.62)  1.416 ms 100.100.8.44 (100.100.8.44)  0.915 ms 100.100.8.80 (100.100.8.80)  0.863 ms
 2 240.0.184.33 (240.0.184.33)  1.181 ms 240.0.184.32 (240.0.184.32)  1.267 ms 240.0.184.34 (240.0.184.34)  1.150 ms
 3 100.100.34.110 (100.100.34.110)  1.283 ms 100.100.36.102 (100.100.36.102)  8.579 ms 100.100.36.108 (100.100.36.108)  8.594 ms
 4 * * *
 5 * * *
 6 142.251.67.234 (142.251.67.234)  1.493 ms 142.251.52.62 (142.251.52.62)  2.381 ms 142.251.52.64 (142.251.52.64)  1.486 ms
 7 192.178.248.38 (192.178.248.38)  2.164 ms 192.178.248.40 (192.178.248.40)  1.899 ms 192.178.248.38 (192.178.248.38)  2.532 ms
 8 142.251.49.187 (142.251.49.187)  2.163 ms 108.170.232.199 (108.170.232.199)  2.372 ms 142.251.49.189 (142.251.49.189)  2.069 ms
 9 142.250.59.233 (142.250.59.233)  3.386 ms 142.250.211.189 (142.250.211.189)  2.482 ms 142.250.59.233 (142.250.59.233)  3.140 ms
10 142.251.52.182 (142.251.52.182)  2.870 ms 142.250.209.44 (142.250.209.44)  7.388 ms 172.253.66.84 (172.253.66.84)  3.125 ms
11 172.253.66.155 (172.253.66.155)  3.392 ms 172.253.66.153 (172.253.66.153)  2.271 ms 172.253.66.157 (172.253.66.157)  3.202 ms
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 bh-in-f101.1e100.net (172.253.122.101)  2.387 ms  2.109 ms *
```

bh-in-f101.1e100.net is a Google server.

Early Hops (1–3)

- 100.100.32.62 ...
- 240.0.184.33 ...
- 100.100.34.110 ...

These are internal **AWS infrastructure** or carrier-grade NAT routers. They're working fine with **low latency**.

Hops 4–6 (Time Out)

- 4 * * *
- 5 * * *
- 6 * * *

This means:

- These intermediate routers **did not respond to the ICMP/UDP packets** used by `traceroute`.
- This is **common and not necessarily a problem** — many networks (especially cloud providers) disable `traceroute` responses for security or performance reasons.

Hops 7–11

These include Google infrastructure:

- 7 142.251.67.234 ...
- 8 192.178.248.38 ...
- 9 142.251.49.187 ...
- 10 142.250.59.233 ...
- 11 142.251.52.182 ...

These show smooth routing through **Google's edge and backbone** network. The latency is consistently low (under ~4 ms), which is excellent.

Hops 12–20 (Time Out Again)

12–20 * * *

Again, these routers chose not to respond.

Since the trace resumes successfully after, it's **not a sign of failure**, just **ICMP filtering**.

Final Hop (21)

21 bh-in-f101.1e100.net (172.253.122.101) 2.387 ms 2.109 ms *

- You **successfully reached Google!**
 - `bh-in-f101.1e100.net` is a Google server.
Round-trip times (RTTs) are low — 2.1 to 2.4 ms, which is **very good**.
-

```
[ec2-user@ip-172-31-41-0 ~]$ nslookup google.com
Server:          172.31.0.2
Address:         172.31.0.2#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.31.113
Name:   google.com
Address: 142.250.31.102
Name:   google.com
Address: 142.250.31.139
Name:   google.com
Address: 142.250.31.101
Name:   google.com
Address: 142.250.31.100
Name:   google.com
Address: 142.250.31.138
Name:   google.com
Address: 2607:f8b0:4004:c17::66
Name:   google.com
Address: 2607:f8b0:4004:c17::8a
Name:   google.com
Address: 2607:f8b0:4004:c17::71
Name:   google.com
Address: 2607:f8b0:4004:c17::65
```



Server:

- Server: 172.31.0.2
- Address: 172.31.0.2#53

This is the **DNS server** your EC2 instance is using.

172.31.0.2 is the **Amazon VPC internal DNS resolver**, automatically provided by AWS.

Non-authoritative answer:

This means the DNS response came **from a cache**, not directly from Google's authoritative name servers — which is totally normal.

You see multiple IP addresses for google.com, both **IPv4** and **IPv6**:

IPv4 Addresses (A Records)

- 142.250.31.113
- 142.250.31.102
- 142.250.31.139
- 142.250.31.101
- 142.250.31.100
- 142.250.31.138

These are different public IPs for Google's servers — this is called **DNS load balancing**. When you access google.com, your browser may connect to any of these IPs for performance or redundancy.

IPv6 Addresses (AAAA Records)

- 2607:f8b0:4004:c17::66
- 2607:f8b0:4004:c17::8a
- 2607:f8b0:4004:c17::71
- 2607:f8b0:4004:c17::65

These are **IPv6 addresses** for the same domain. If your system and network support IPv6, it may prefer these.

google.com resolves to multiple IPs, which is normal for large services using **anycast**, **CDNs**, and **geographically distributed servers**.
