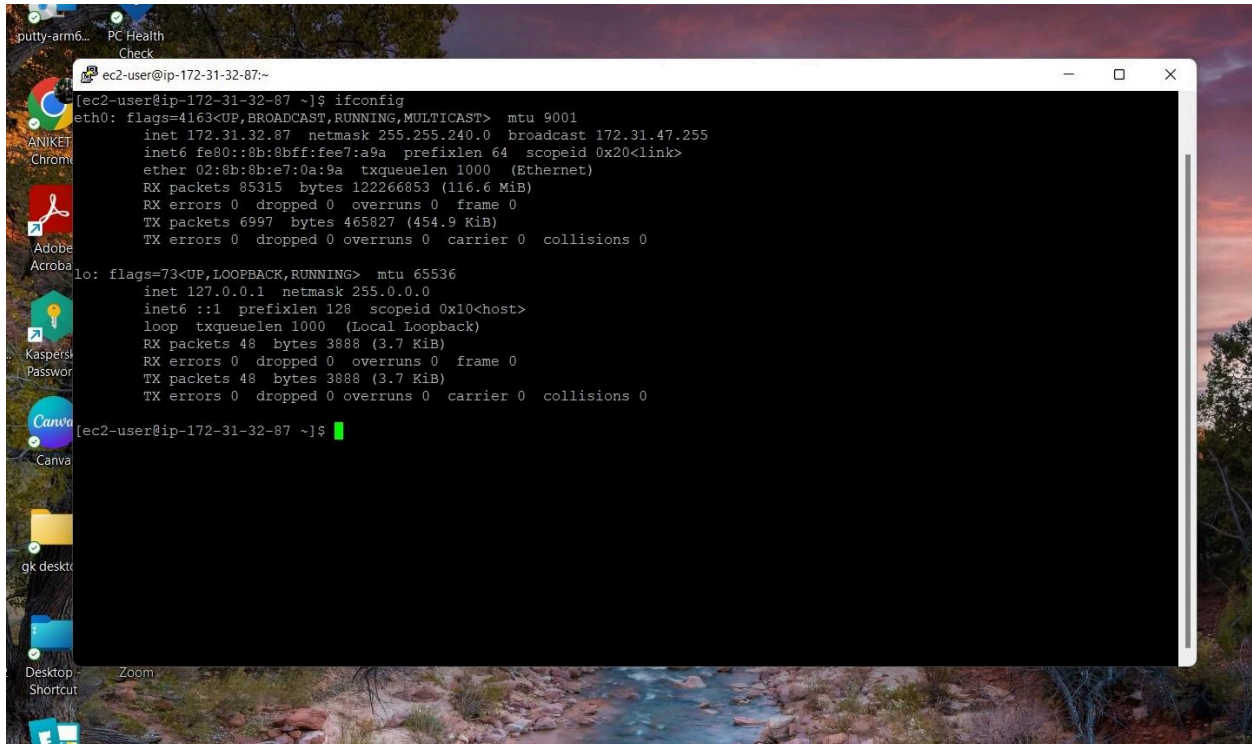


Task 2: Network Commands in Linux

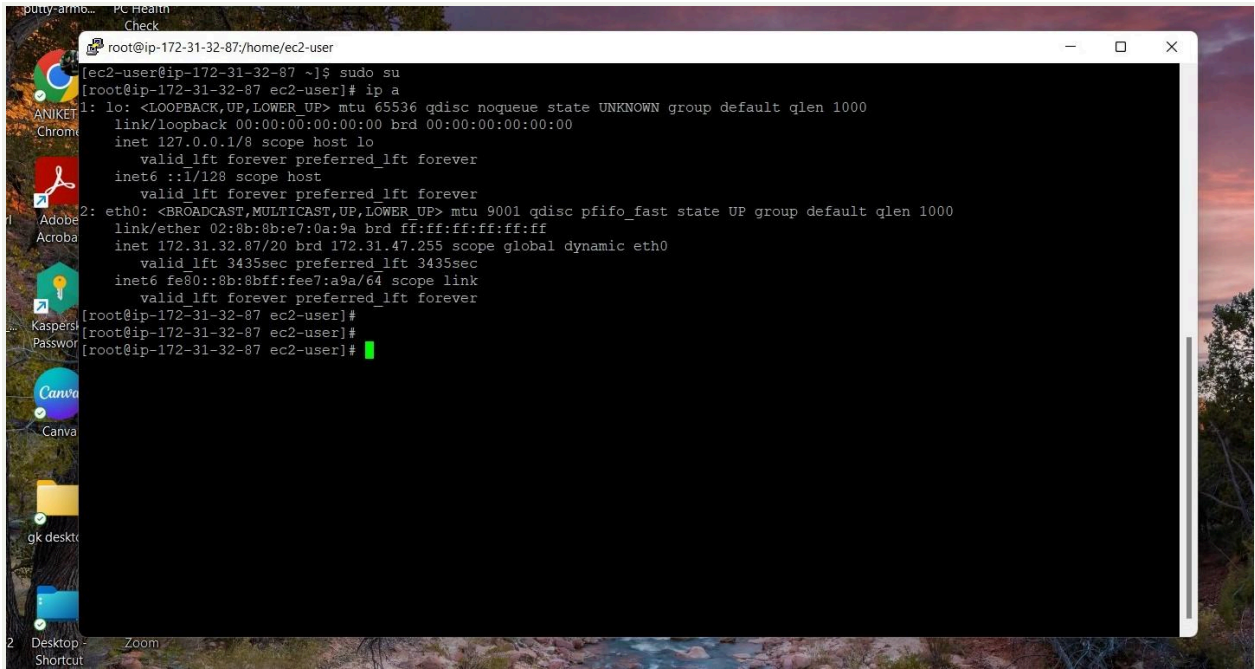


```
ec2-user@ip-172-31-32-87:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
    inet 172.31.32.87  netmask 255.255.240.0  broadcast 172.31.47.255
    inet6 fe80::8b:8bff:fee7:a9a  prefixlen 64  scopeid 0x20<link>
    ether 02:8b:8b:e7:0a:9a  txqueuelen 1000  (Ethernet)
    RX packets 85315  bytes 122266853 (116.6 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 6997  bytes 465827 (454.9 KiB)
    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 48  bytes 3888 (3.7 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 48  bytes 3888 (3.7 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

ec2-user@ip-172-31-32-87 ~$
```

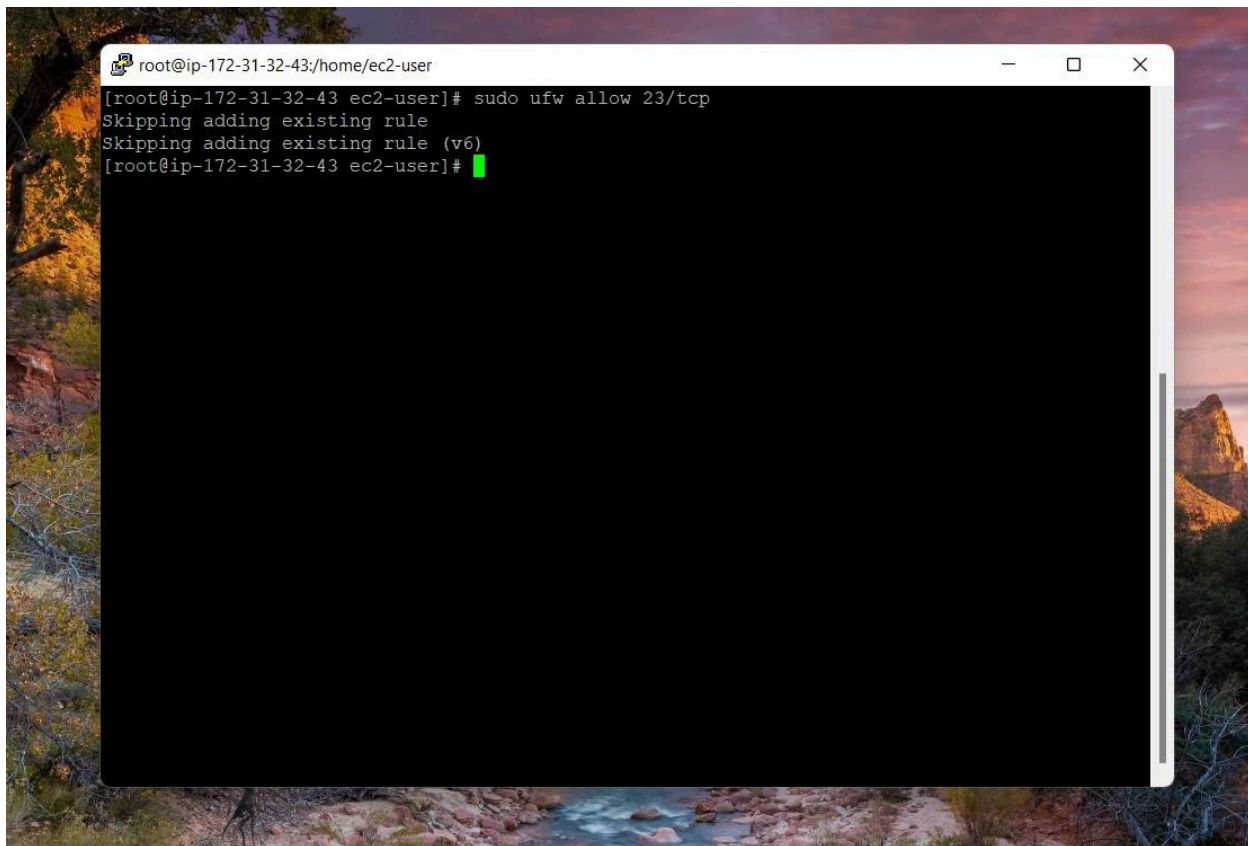
- **Ifconfig** display and manipulates route and network interfaces.



The image shows a terminal window titled "root@ip-172-31-32-87/home/ec2-user" running on a Linux desktop. The desktop background is a scenic landscape with a river and trees. On the left side, there is a dock with several application icons: ANIKET, Chrome, Adobe Acrobat, Kaspersky Password Manager, Canva, and a folder named "gk desktop". The terminal window displays the following commands and output:

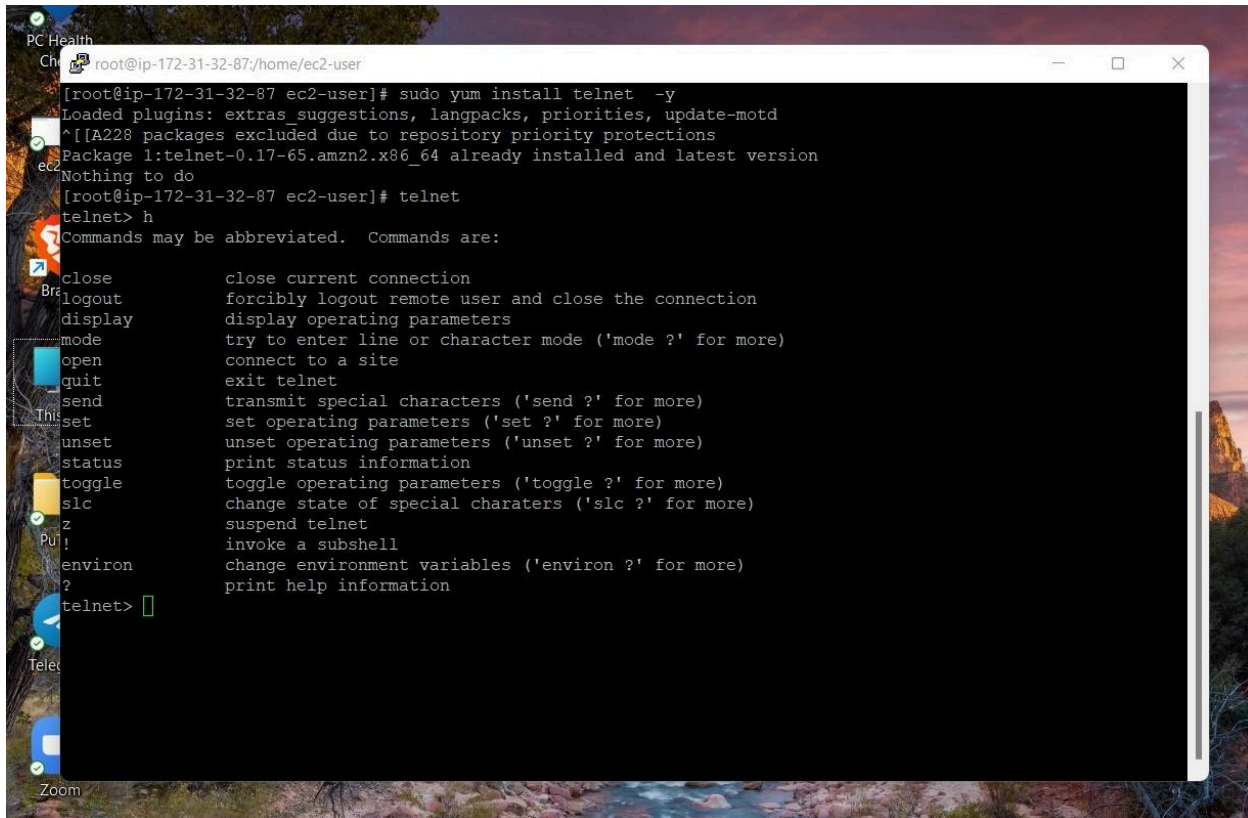
```
(ec2-user@ip-172-31-32-87 ~)$ sudo su
[root@ip-172-31-32-87 ec2-user]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 02:8b:8b:e7:0a:9a brd ff:ff:ff:ff:ff:ff
    inet 172.31.32.87/20 brd 172.31.47.255 scope global dynamic eth0
        valid_lft 3435sec preferred_lft 3435sec
    inet6 fe80::8b:8b:ff:fe7:a9a/64 scope link
        valid_lft forever preferred_lft forever
[root@ip-172-31-32-87 ec2-user]#
[root@ip-172-31-32-87 ec2-user]#
[root@ip-172-31-32-87 ec2-user]#
```

- **Ip a** it is a replacement of the ifconfig command.

A terminal window is displayed over a scenic background of a river and mountains at sunset. The terminal window has a title bar with a small icon and the text 'root@ip-172-31-32-43/home/ec2-user'. The terminal content shows the command 'sudo ufw allow 23/tcp' being executed. The output consists of three lines: 'Skipping adding existing rule', 'Skipping adding existing rule (v6)', and a green cursor on the line '[root@ip-172-31-32-43 ec2-user]#'.

```
root@ip-172-31-32-43/home/ec2-user
[root@ip-172-31-32-43 ec2-user]# sudo ufw allow 23/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
[root@ip-172-31-32-43 ec2-user]#
```

- **ufw** is used for To enable a firewall application profile
- **Install and enable epel repository** on Amazon Linux
- Then **enable** ufw to open the port.
- **ufw allow 23/tcp** open port23 in the ufw firewall.



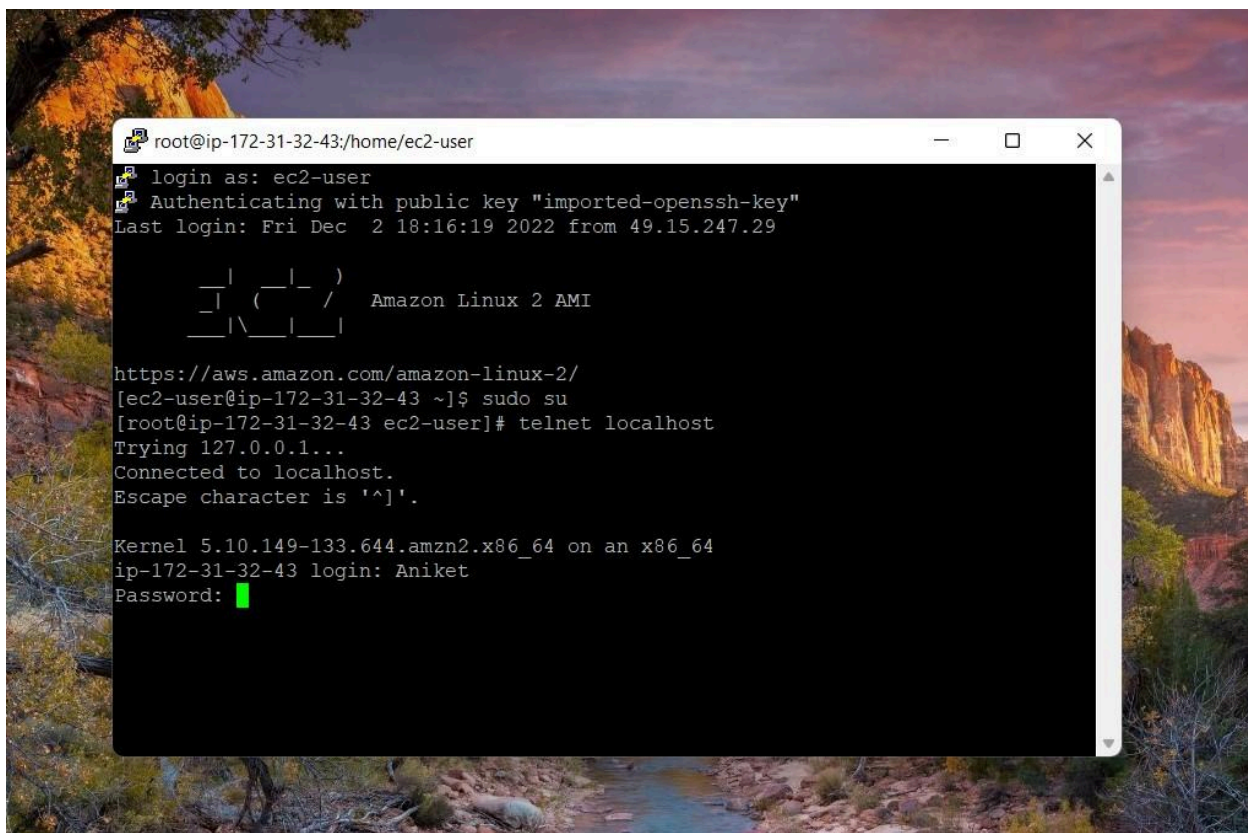
The screenshot shows a terminal window on a Linux desktop. The terminal prompt is `[root@ip-172-31-32-87 ec2-user]#`. The user enters `sudo yum install telnet -y`, which outputs: `Loaded plugins: extras_suggestions, langpacks, priorities, update-motd`, `^[[A228 packages excluded due to repository priority protections`, and `Package 1:telnet-0.17-65.amzn2.x86_64 already installed and latest version`. The user then enters `telnet`, which outputs: `telnet> h` and `Commands may be abbreviated. Commands are:`. A list of telnet commands and their descriptions is displayed, including `close`, `logout`, `display`, `mode`, `open`, `quit`, `send`, `set`, `unset`, `status`, `toggle`, `slc`, `z`, `!env`, and `?`. The terminal window is titled `root@ip-172-31-32-87:/home/ec2-user` and has standard window controls.

```
[root@ip-172-31-32-87 ec2-user]# sudo yum install telnet -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
^[[A228 packages excluded due to repository priority protections
Package 1:telnet-0.17-65.amzn2.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-32-87 ec2-user]# telnet
telnet> h
Commands may be abbreviated.  Commands are:

close                close current connection
logout               forcibly logout remote user and close the connection
display              display operating parameters
mode                 try to enter line or character mode ('mode ?' for more)
open                 connect to a site
quit                 exit telnet
send                 transmit special characters ('send ?' for more)
set                  set operating parameters ('set ?' for more)
unset                unset operating parameters ('unset ?' for more)
status               print status information
toggle               toggle operating parameters ('toggle ?' for more)
slc                  change state of special charaters ('slc ?' for more)
z                    suspend telnet
!                     invoke a subshell
!env                  change environment variables ('environ ?' for more)
?                     print help information

telnet> 
```

- **Telnet** is commonly used by terminal emulation programs that allow you to log into a remote host.
- **Install** telnet in the linux
- Start and enable the telnet
- **telnet** commands exclusive for telnet

A terminal window is displayed over a scenic background of a river and cliffs. The terminal shows the process of logging in as 'ec2-user' using a public key, then using 'sudo su' to become root. Finally, 'telnet localhost' is used to connect to the local host, showing a successful connection to 127.0.0.1. The user 'Aniket' is prompted for a password.


```
root@ip-172-31-32-43:/home/ec2-user
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Fri Dec  2 18:16:19 2022 from 49.15.247.29

  _ | _ | _ |
 _ | ( _ | _ | /   Amazon Linux 2 AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-32-43 ~]$ sudo su
[root@ip-172-31-32-43 ec2-user]# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Kernel 5.10.149-133.644.amzn2.x86_64 on an x86_64
ip-172-31-32-43 login: Aniket
Password: 
```

- We are connecting our system with the **localhost**
- Fill the login details and password.

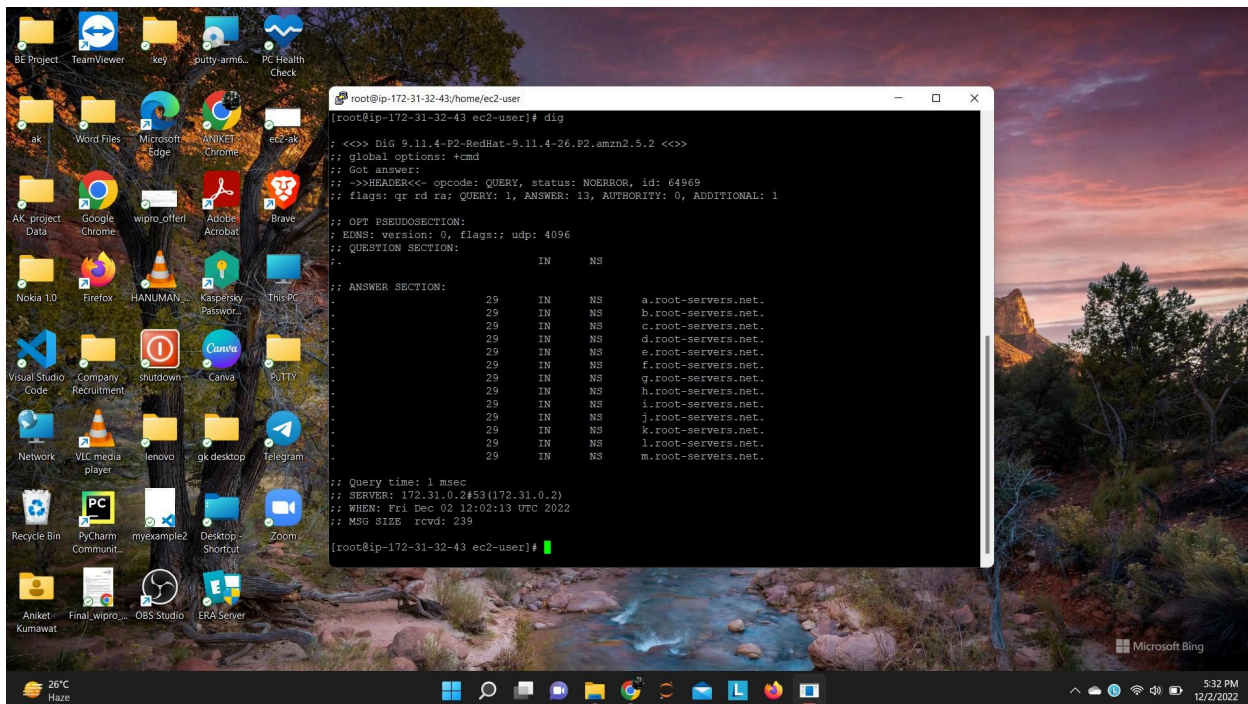
 root@ip-172-31-32-43:/home/ec2-user

```
[ec2-user@ip-172-31-32-43 ~]$ sudo su
[root@ip-172-31-32-43 ec2-user]# ping 172.31.32.43
ping: 172.31.32.43: Name or service not known
[root@ip-172-31-32-43 ec2-user]# ping 172.31.32.43
PING 172.31.32.43 (172.31.32.43) 56(84) bytes of data.
64 bytes from 172.31.32.43: icmp_seq=1 ttl=255 time=0.030 ms
64 bytes from 172.31.32.43: icmp_seq=2 ttl=255 time=0.037 ms
64 bytes from 172.31.32.43: icmp_seq=3 ttl=255 time=0.036 ms
64 bytes from 172.31.32.43: icmp_seq=4 ttl=255 time=0.038 ms
64 bytes from 172.31.32.43: icmp_seq=5 ttl=255 time=0.065 ms
64 bytes from 172.31.32.43: icmp_seq=6 ttl=255 time=0.039 ms
64 bytes from 172.31.32.43: icmp_seq=7 ttl=255 time=0.048 ms
64 bytes from 172.31.32.43: icmp_seq=8 ttl=255 time=0.038 ms
█
```

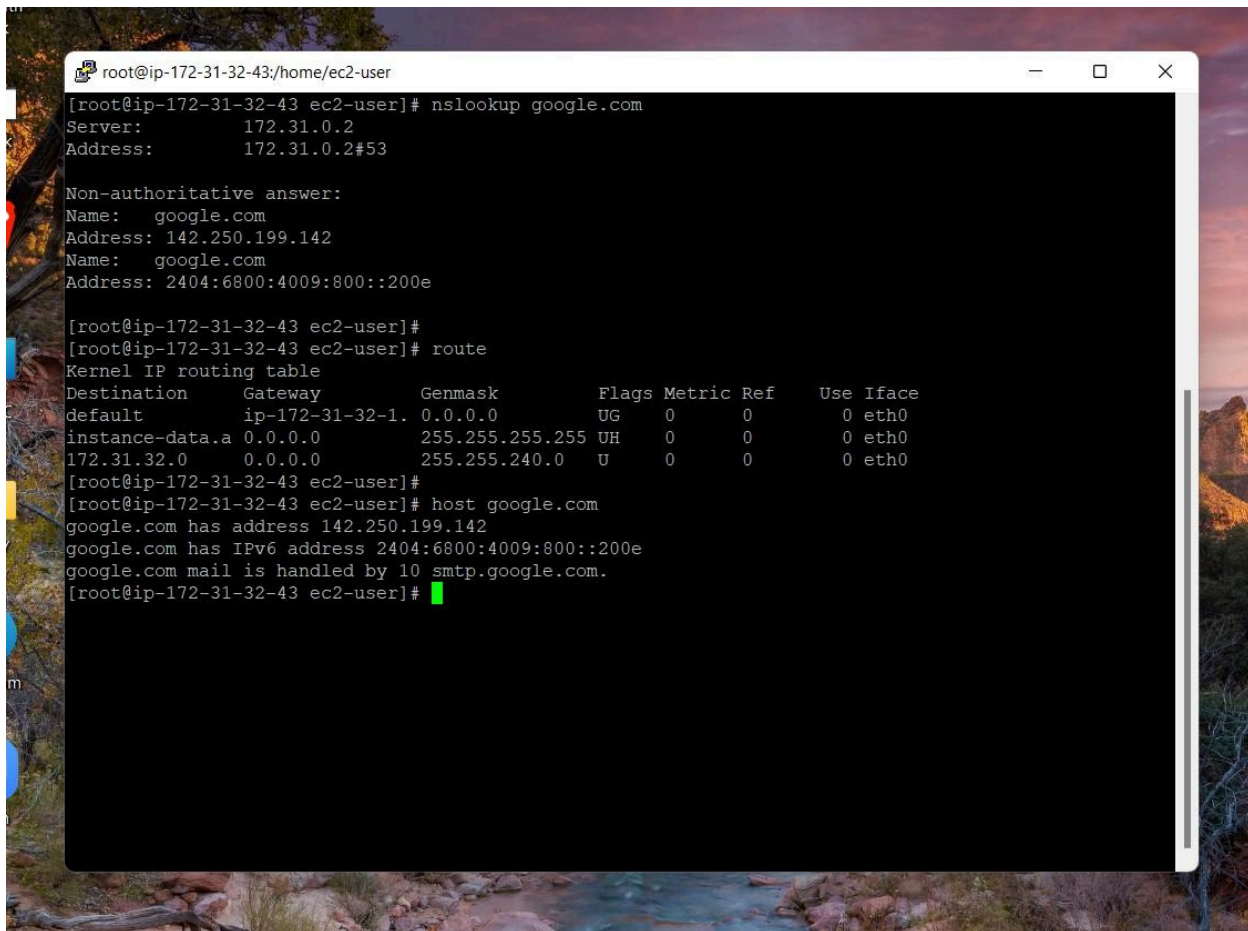
- **Ping** is used to check the network connectivity between host and server/host.


```
root@ip-172-31-32-43/home/ec2-user
[root@ip-172-31-32-43 ec2-user]# netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      2587/rpcbind
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      3061/sshd
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN      3016/master
tcp6       0      0 :::111                 :::*                   LISTEN      2587/rpcbind
tcp6       0      0 :::22                 :::*                   LISTEN      3061/sshd
tcp6       0      0 :::23                 :::*                   LISTEN      1/systemd
udp        0      0 0.0.0.0:68             0.0.0.0:*               *          2824/dhclient
udp        0      0 0.0.0.0:111            0.0.0.0:*               *          2587/rpcbind
udp        0      0 0.0.0.0:629            0.0.0.0:*               *          2587/rpcbind
udp        0      0 127.0.0.1:323          0.0.0.0:*               *          2601/chronyd
udp6       0      0 fe80::dd:8aff:fe55::546 :::*                   *          2873/dhclient
udp6       0      0 :::111                 :::*                   *          2587/rpcbind
udp6       0      0 :::629                 :::*                   *          2587/rpcbind
udp6       0      0 :::1:323               :::*                   *          2601/chronyd
[root@ip-172-31-32-43 ec2-user]#
```

- The network statistics (**netstat**) command is a networking tool used for troubleshooting and configuration.
- **netstat -tunlp** displays connection information. And shows the port number.



- The **dig (domain information groper)** command is a flexible tool for interrogating DNS name servers.
- **dig** query DNS related information
- **dig** command replaces older tools such as nslookup and the host.

A screenshot of a terminal window titled 'root@ip-172-31-32-43:/home/ec2-user'. The terminal shows the following commands and output:

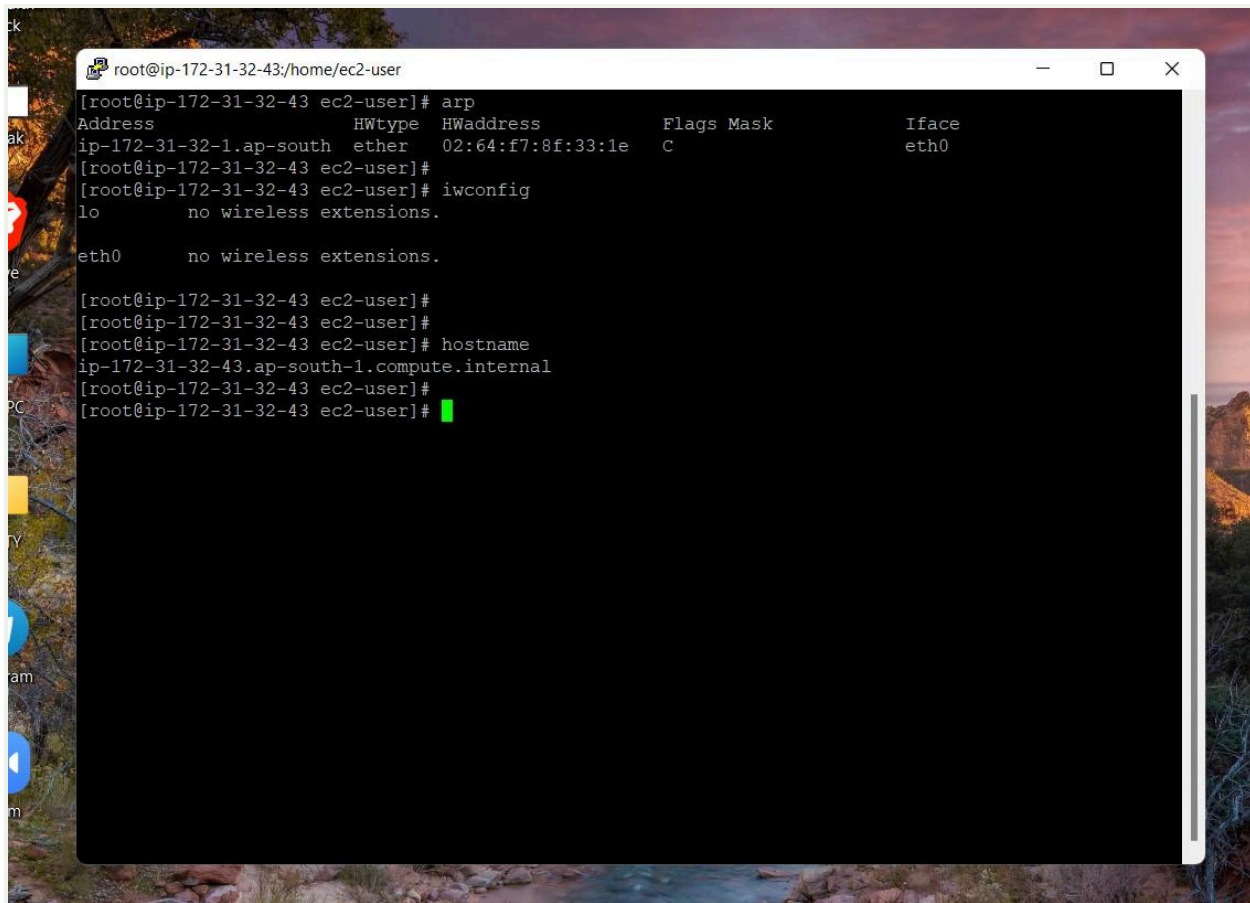
```
[root@ip-172-31-32-43 ec2-user]# nslookup google.com
Server:          172.31.0.2
Address:         172.31.0.2#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.199.142
Name:   google.com
Address: 2404:6800:4009:800::200e

[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         ip-172-31-32-1 0.0.0.0         UG    0      0      0 eth0
instance-data.a 0.0.0.0         255.255.255.255 UH    0      0      0 eth0
172.31.32.0     0.0.0.0         255.255.240.0   U     0      0      0 eth0

[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]# host google.com
google.com has address 142.250.199.142
google.com has IPv6 address 2404:6800:4009:800::200e
google.com mail is handled by 10 smtp.google.com.
[root@ip-172-31-32-43 ec2-user]#
```

- **Nslookup** (stands for “Name Server Lookup”) is a useful command for getting information from the DNS server
- **nslookup google.com** (shows the ip of website)
- **route** shows and manipulates IP routing table.
- **route** is used for showing or update the IP/kernel routing table.
- **Linux host** command displays domain name for given IP address or vice-versa.
- **host** performs DNS lookups.

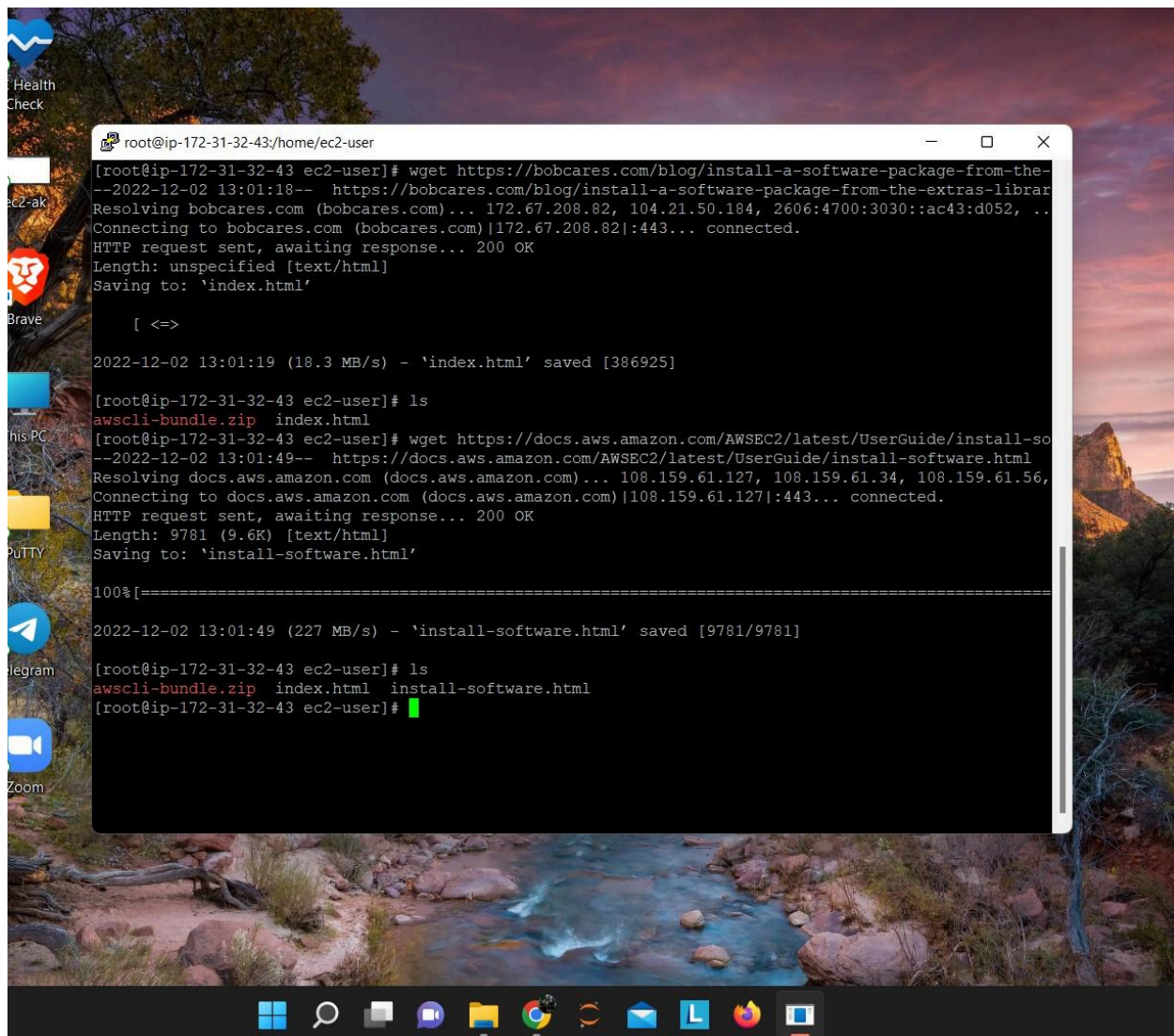


```
root@ip-172-31-32-43/home/ec2-user
[root@ip-172-31-32-43 ec2-user]# arp
Address            HWtype  HWaddress           Flags Mask            Iface
ip-172-31-32-1.ap-south  ether    02:64:f7:8f:33:1e    C                    eth0
[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]# iwconfig
lo                no wireless extensions.

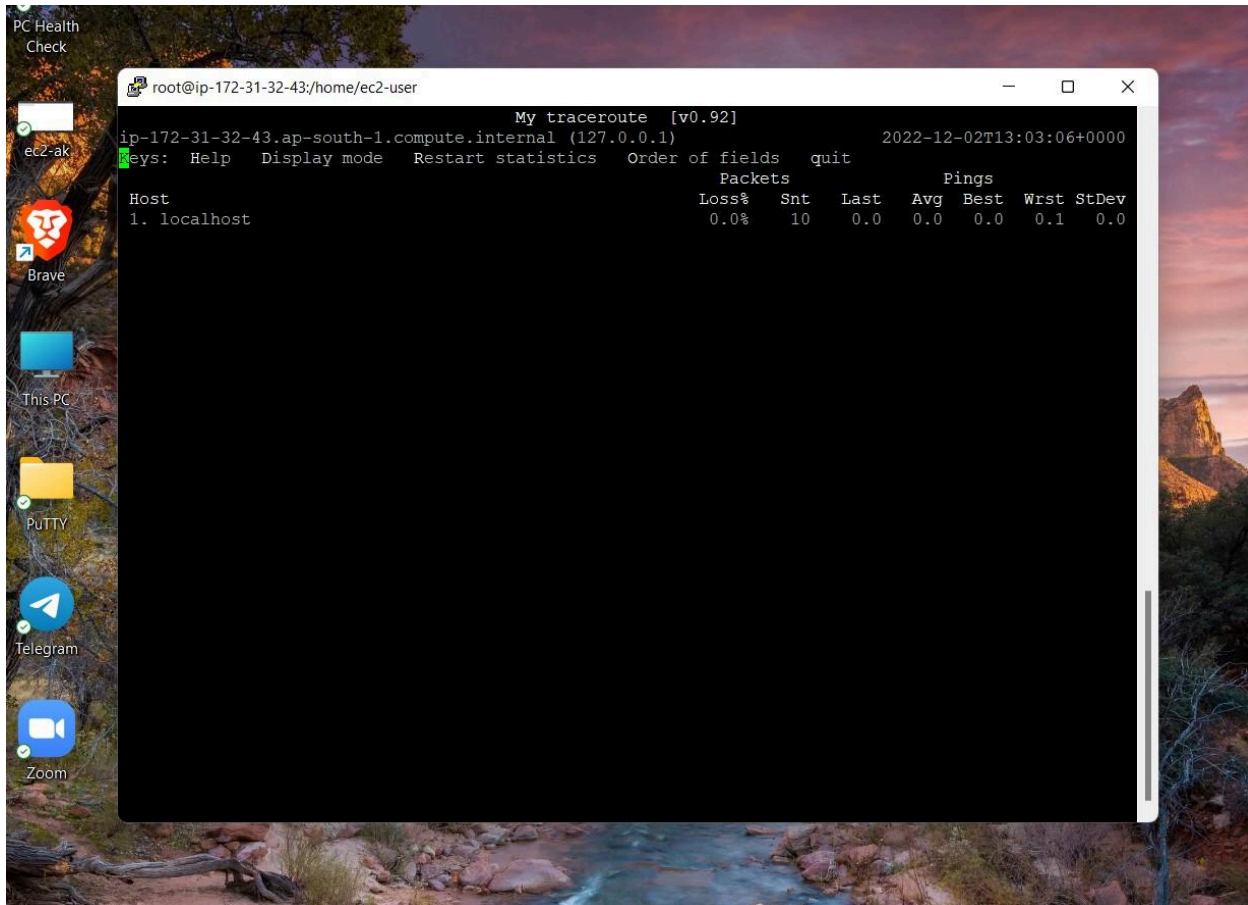
eth0              no wireless extensions.

[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]# hostname
ip-172-31-32-43.ap-south-1.compute.internal
[root@ip-172-31-32-43 ec2-user]#
[root@ip-172-31-32-43 ec2-user]#
```

- **arp** view or add contents of the kernel's ARP table.
- **iwconfig** used to display and change the parameters of the network interface which are specific to the wireless operation
- **hostname** to identify a network name.



- **wget** is the non-interactive network downloader which is used to download files from the server even when the user has not logged on to the system and it can work in the background without hindering the current process.
- **wget** (download the package from the internet)



- **mtr** is a networking tool that combines ping and traceroute to diagnose a network.


```
root@ip-172-31-32-43:/home/ec2-user
[root@ip-172-31-32-43 ec2-user]# whois google.com
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2086851750
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2022-12-02T13:03:47Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

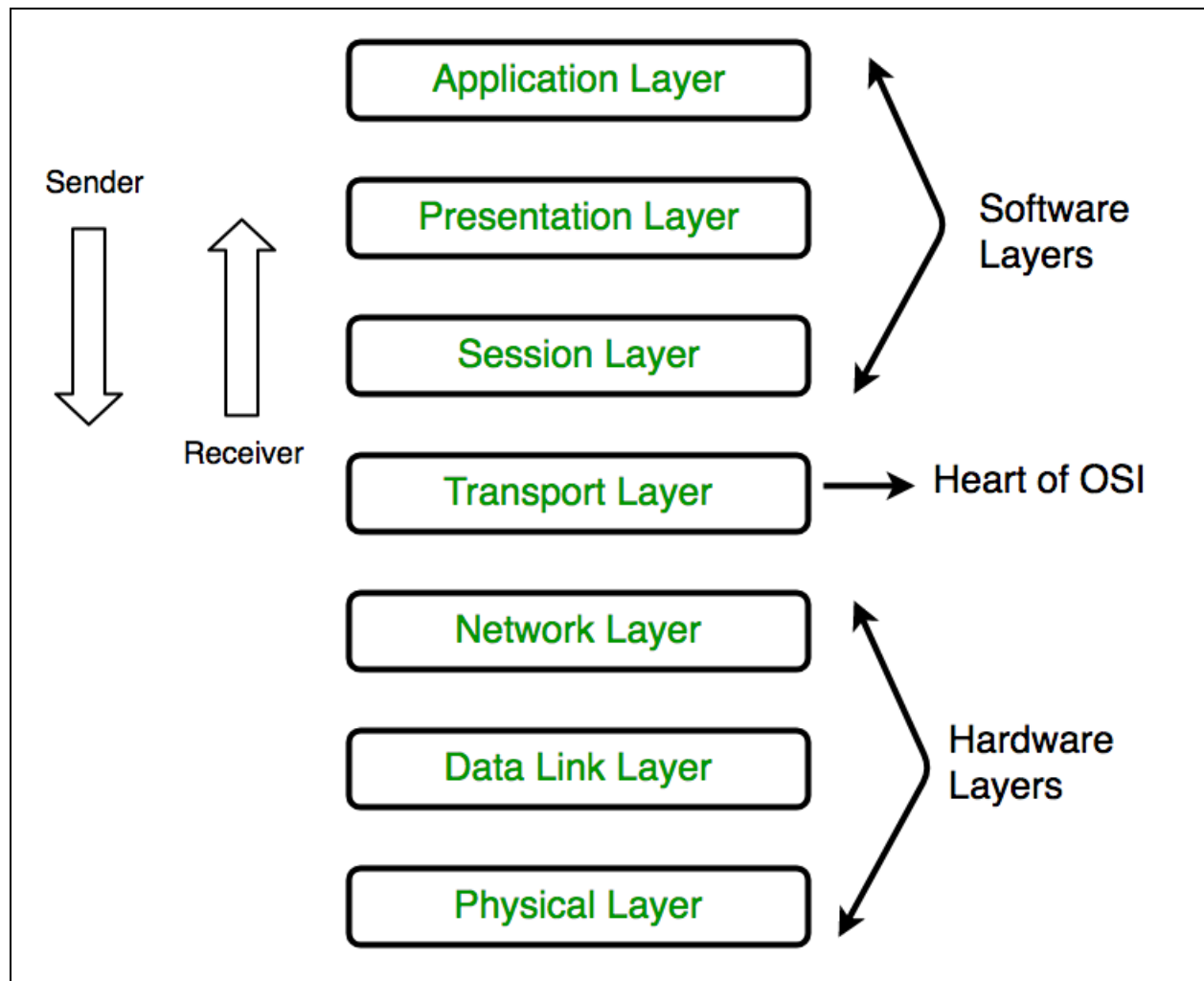
NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.
```

- **whois** is a query and response protocol that is widely used for querying databases that store the registered users or assignees of an Internet resource
- **whois** will tell you about the website's whois.

Ports :

NAME	PORT
http	80
https	443
RDP	3389
ssh	22
FTP	20 & 21
Telnet	23

OSI Model Diagram - (Layer 4, Layer 7):-



- Layer 4 of the **OSI model**, also known as the **transport** layer, manages network traffic between hosts and end systems to ensure complete data transfers.
- **Application** layer is the highest level of open systems, providing services directly for the application process. It allows a user to access, retrieve, and manage files in a remote computer.