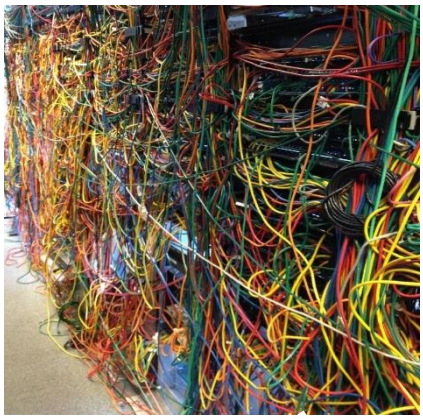
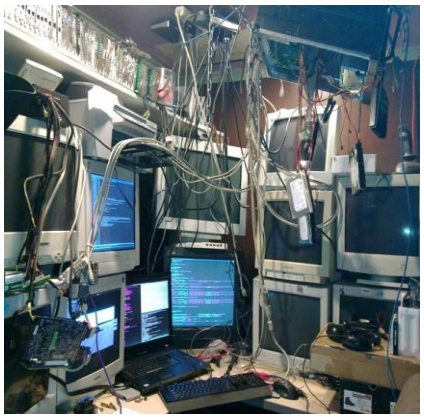
HELLO GALAXY

Parsa Partovi

Maktab / Network Q





Q1 : use the ping command to test connectivity between two devices on a network ping the IP address of a known device on the network and check if receive a response .

Step 1 : first we should check for the IP address if want to check ourselves ping we gotta search www.whatismyipaddress.com or similar sites and copy the IP .

Step 2 : open cmd (windows) or terminal (linux)

Step 3: we use the ping command and after we enter the IP address that we want to check its ping .

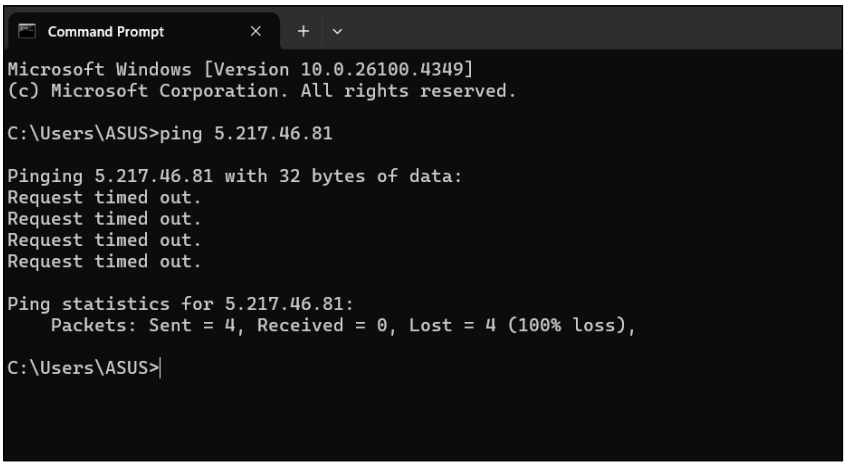
warning : request may fail or come with no response for too long due to IP address or network connection .

Step 4 :for successful connectivity we'll see some replies that include byte , IP , time and …

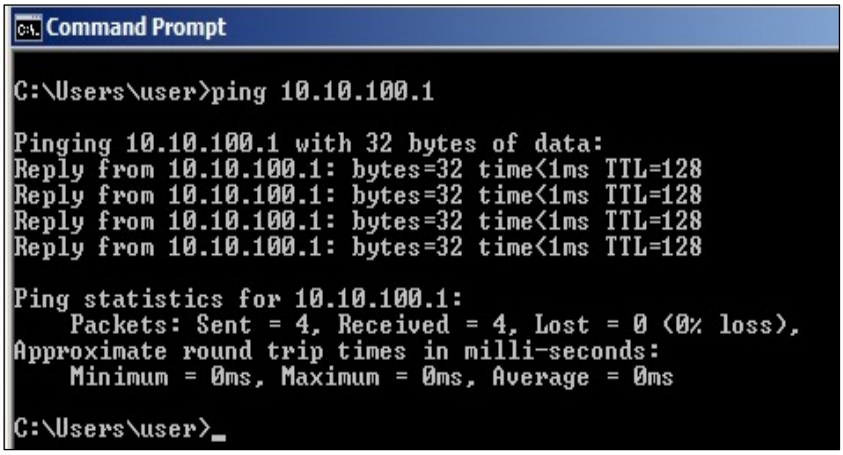
Step 5 :to stop the process on windows cmd it will stop automatically after 4 packets but at linux terminal we should press cntrl + c

On windows

Request timed out example

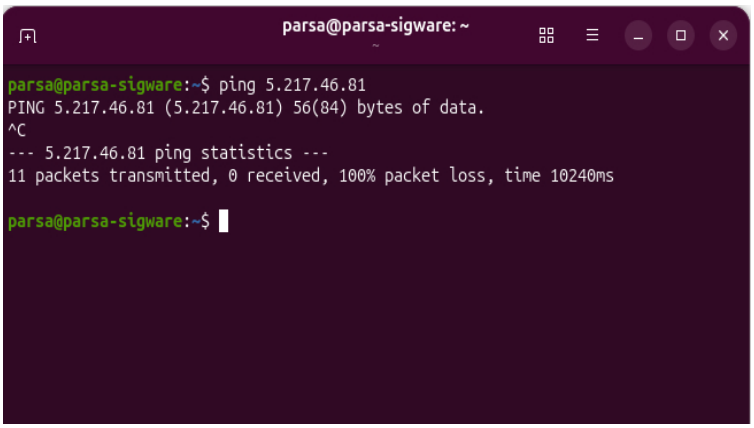


Successful request :

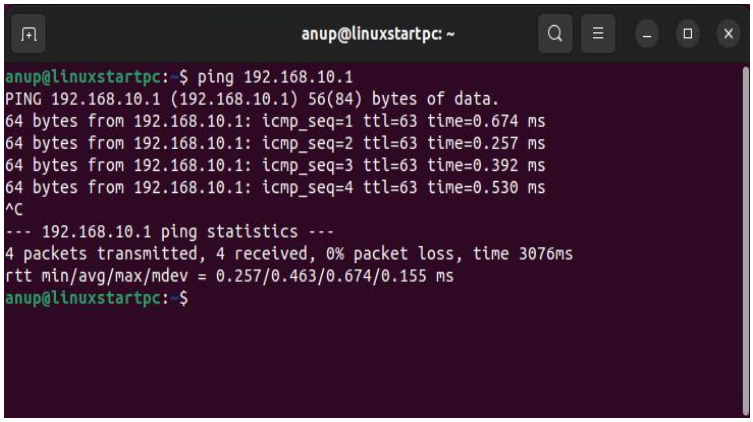


On linux

Unsuccessful /timeout :



Successful:



Q2 : Modify the ping command to specify the the number of pockets to send and observe the response time for each pocket

Step 1 and 2 same as the Q1 ;

Step 3 : on windows; we use -n option to specify the number of packet requests but on linux we use -c

Step 4 : to observe the response time at the end of each pocket ; we look at "round trip time " time=0._ _ _ for each packet , the lower that number be , the faster that packet is!

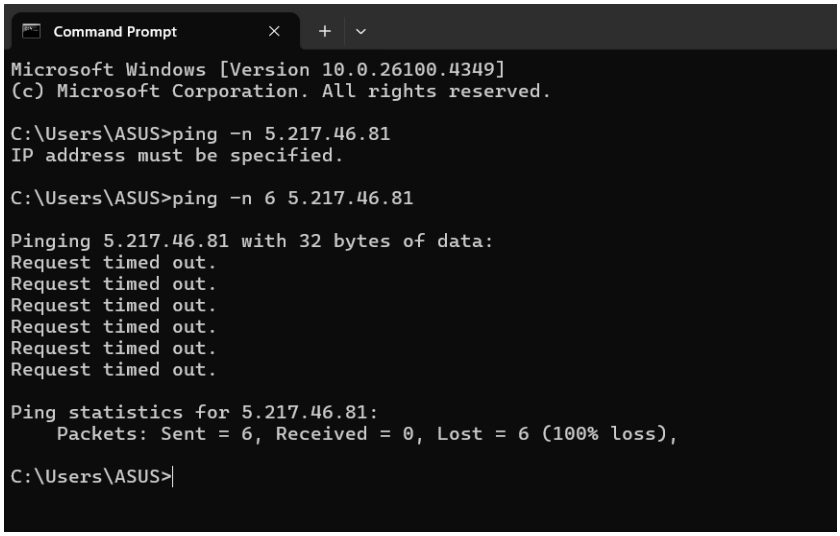
Warning : if it was unsuccessful we'll get timed outs or no response (linux) .

Step 5 : at the end we get a summery that includes "min/avg/max/mdev " minimum, average, maximum, and deviation of response times to measure network stability .

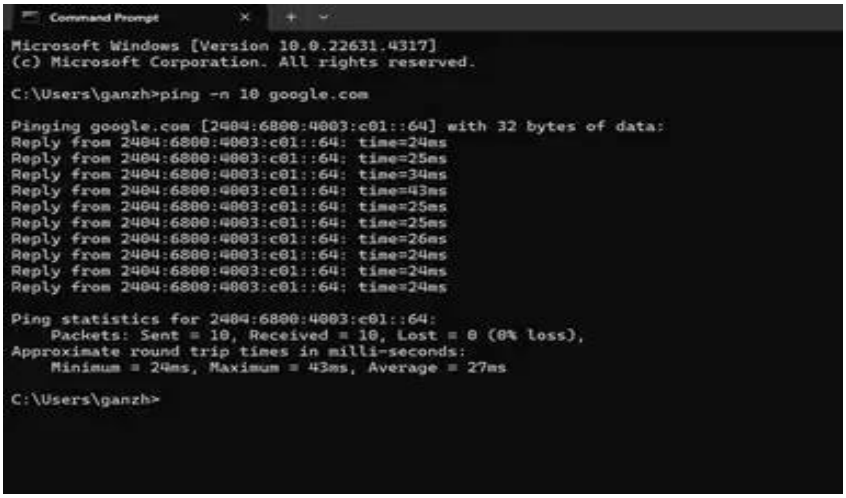
Step 6 : in order to exit while process is failed in linux we got to press cntrl + c

On windows

Unsuccessful:

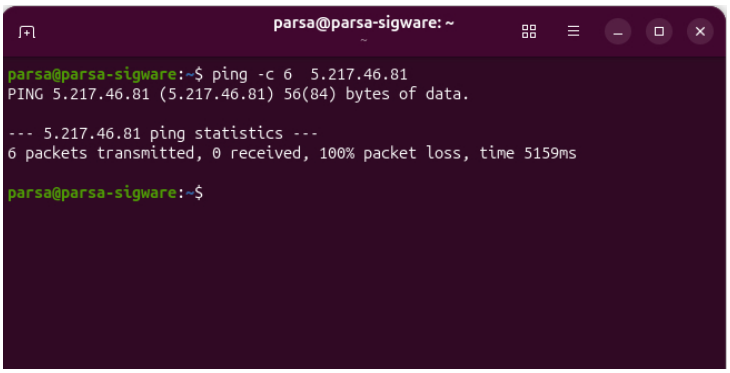


Successful:

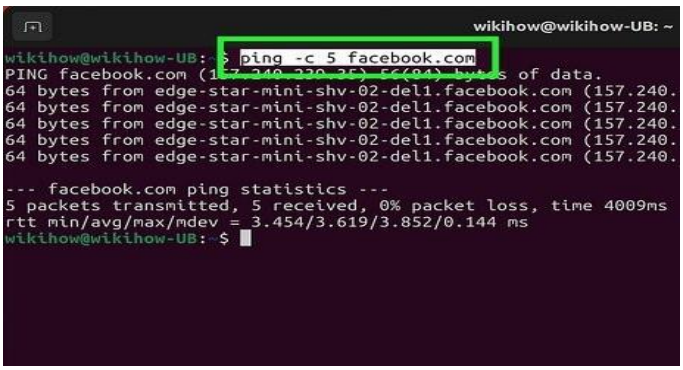


On linux

Unsuccessful :



Successful:



Q3 : Use the ping command to test the connectivity between two devices on different networks and observe the response .

Same steps as before just putting the 8.8.8.8 instead of IP .

If you want to test connectivity to a device on another network (e.g., across the internet), you need its public IP address or domain name.

Example:

8.8.8.8 (Google's public DNS server).

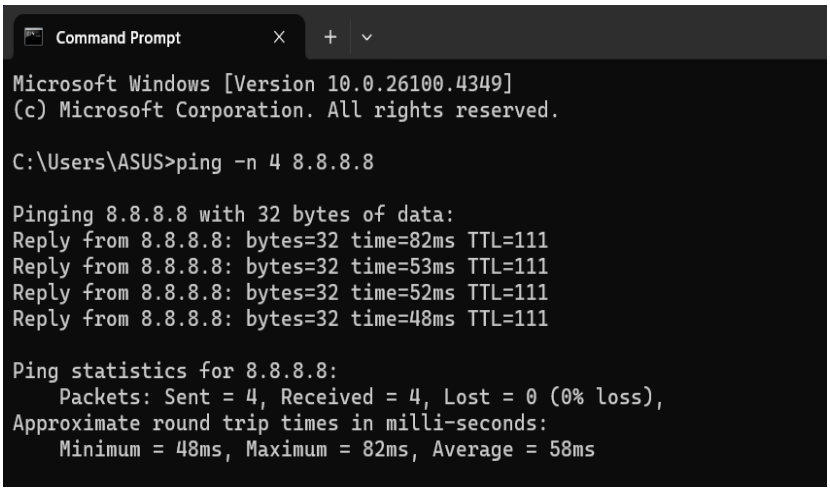
In case of fail well see the sane timouts such as :

Request timed out.

Destination Host Unreachable.

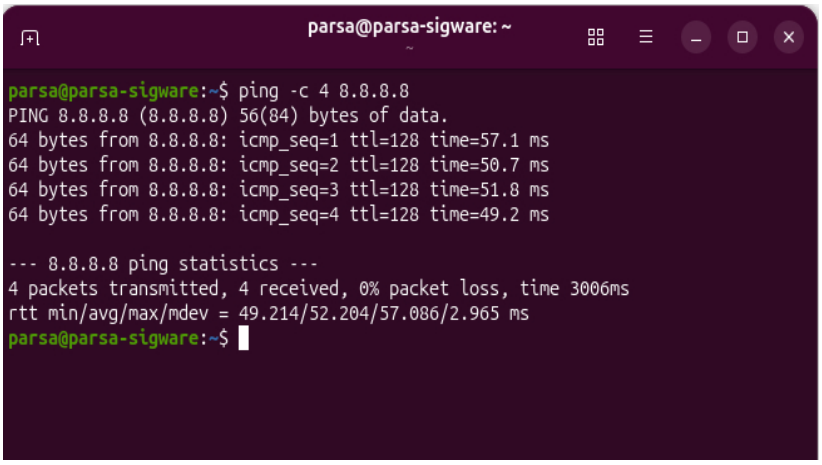
On windows

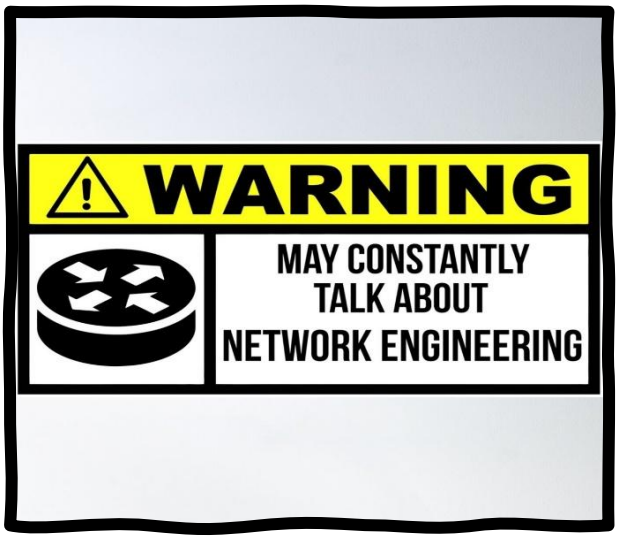
Successful :



On linux

Successful :





Q4 : use traceroute/tracepath command to trace the path a packet takes from your device to a specific Website , analyze the IP addresses of the routes displayed .

using traceroute (Linux) or tracepath (Linux) to trace the path a packet takes to a website, and then analyzing the IP addresses.

Step 1 :
Linux → traceroute <an address>
Windows → tracert <an address>

Step 2 :
after the following commands ; You'll see a list of "hops" , each hop is a router or device that your packet passes through on the way to the destination .

Step 3 :
to analyze the IP addresses ; we should the hop purpose ; a hop shows:

IP address of the router.

Response times (ms) → round-trip latency.

If a hop shows * * *, it means the router didn't respond to ICMP requests (common for security reasons).

Hop 1 (192.168.1.1) → Your local router (private IP).
Hop 2 (10.0.0.1) → Your ISP's gateway (private IP).
Hop 3 (203.0.113.5) → ISP backbone router (public IP)
Hop 4 (198.51.100.22) → Transit provider or regional internet exchange.
Hop 5 (142.250.185.14) → Destination server (Google).

On windows

Successful :

```
Command Prompt
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>tracert www.google.com

Tracing route to www.google.com [216.239.38.120]
over a maximum of 30 hops:

  1   122 ms     *       69 ms  10.228.168.110
  2     *        *        *     Request timed out.
  3    29 ms    39 ms    35 ms  10.134.144.129
  4    25 ms    34 ms    19 ms  10.134.147.1
  5    42 ms    24 ms    31 ms  10.0.138.35
  6    58 ms    44 ms    41 ms  10.0.12.105
  7    30 ms    49 ms    26 ms  10.0.81.213
  8    42 ms    27 ms    28 ms  10.0.81.218
  9    57 ms    66 ms    57 ms  10.202.7.132
 10    30 ms   118 ms    47 ms  10.21.211.10
 11    84 ms    46 ms    57 ms  134.0.220.186
 12    54 ms    70 ms     *     213.202.5.239
 13    47 ms    58 ms    56 ms  216.239.48.87
 14    72 ms    78 ms   322 ms  192.178.87.251
 15   412 ms   199 ms    68 ms  any-in-2678.1e100.net [216.239.38.120]

Trace complete.
```

On linux

At linux it's a little;e bit different in case you be having the "traceroute" you can give the address , if not , you should download it using sudo

Step 1

```
parsa@parsa-sigware: ~
parsa@parsa-sigware:~$ traceroute www.google.com
Command 'traceroute' not found, but can be installed with:
sudo apt install inetutils-traceroute  # version 2:2.6-1ubuntu3, or
sudo apt install traceroute            # version 1:2.1.6-1
parsa@parsa-sigware:~$ sudo apt install traceroute
[sudo: authenticate] Password:
Installing:
  traceroute

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1
  Download size: 59.2 kB
  Space needed: 180 kB / 44.5 MB available

Get:1 http://ir.archive.ubuntu.com/ubuntu questing/universe amd64 traceroute amd64 1:2.1.6-1 [59.2 kB]
Fetched 59.2 kB in 0s (151 kB/s)
Selecting previously unselected package traceroute.
(Reading database ... 147261 files and directories currently installed.)
Preparing to unpack .../traceroute_1%3a2.1.6-1_amd64.deb ...
Unpacking traceroute (1:2.1.6-1) ...
Setting up traceroute (1:2.1.6-1) ...
update-alternatives: using /usr/bin/traceroute.db to provide /usr/bin/traceroute (traceroute) in auto mode
update-alternatives: using /usr/bin/traceroute6.db to provide /usr/bin/traceroute6 (traceroute6) in auto mode
update-alternatives: using /usr/bin/lft.db to provide /usr/bin/lft (lft) in auto mode
update-alternatives: using /usr/bin/traceproto.db to provide /usr/bin/traceproto (traceproto) in auto mode
update-alternatives: using /usr/sbin/tcptraceroute.db to provide /usr/sbin/tcptraceroute (tcptraceroute) in auto mode
Processing triggers for man-db (2.13.1-1) ...
```

Step 2

```
parsa@parsa-sigware:~$ traceroute www.google.com
traceroute to www.google.com (216.239.38.120), 30 hops max, 60 byte packets
 1  _gateway (192.168.31.2)  0.509 ms  0.455 ms  0.428 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Hint :

Similar result in case of successful trace here we've failed

Q6 : modify the traceroute/tracepath command to specify the maximum numbers of hobs and observe the results

We use Use -m (Linux) or -h (Windows) to set the maximum hops. Lowering the number shows only part of the route, while increasing it ensures you capture long paths across the internet backbone.

-Default behavior→ traceroute usually allows up to 30 hops.

-Modified behavior→ when you lower the maximum hops, you may not reach the destination. This is useful if you only want to see the “nearby” network path (like your router + ISP).

-Analysis→Early hops are usually private IPs (your LAN).

Middle hops are ISP and backbone routers.

Final hop is the destination server.

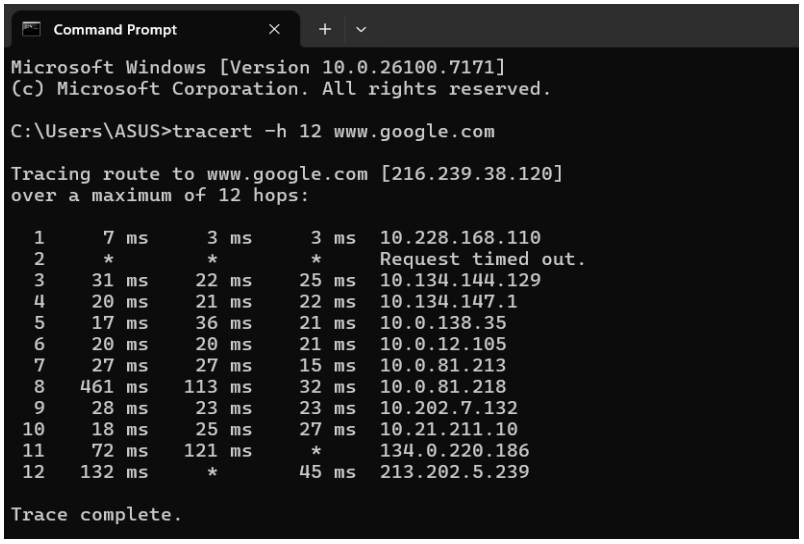
If you cut the max hops short, you’ll only see part of the journey.

"gateway" refers to your default gateway/router — the first hop out of your local network . It’s not a flag you set, but rather the program telling you the packet went on gateway (your router).

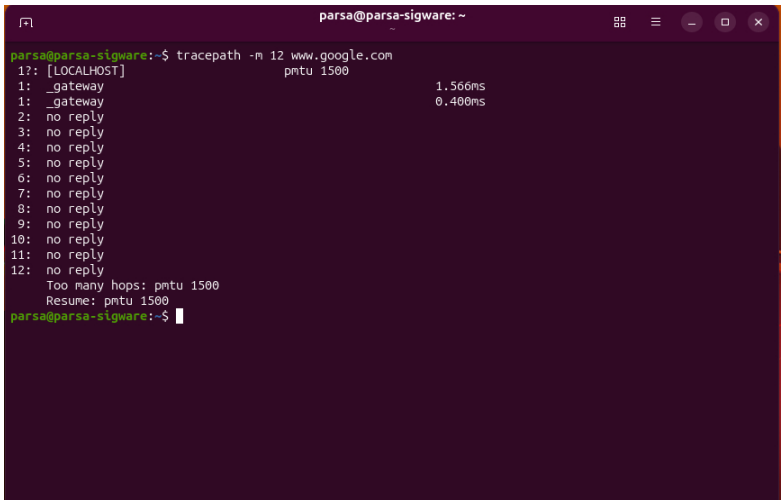
no reply was received from that hop.The router/firewall is configured to ignore ICMP packets (common for security).The packet was dropped due to congestion or filtering. The hop exists, but it doesn’t send back the "Time Exceeded" message.

On windows

successful



On linux



Hint :

It doesn’t necessarily mean the path is broken — traffic may still be forwarded, but the router just doesn’t respond to traceroute probes.

Use traceroute/trace path to trace the path between two devices on a local network and analyze the output

On windows

```
Command Prompt
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>tracert 192.168.1.50

Tracing route to 192.168.1.50 over a maximum of 30 hops

  1     *        *        *     Request timed out.
  2     *        *        *     Request timed out.
  3    55 ms    62 ms    57 ms  10.155.144.65
  4    73 ms    50 ms    55 ms  10.155.147.1
  5    34 ms    51 ms    81 ms  10.0.80.201
  6   114 ms   111 ms    26 ms  10.0.81.99
  7    22 ms    54 ms    36 ms  10.0.81.241
  8    57 ms    28 ms    57 ms  10.0.81.246
  9    47 ms    36 ms    41 ms  10.21.21.11
 10     *        *        *     Request timed out.
 11     *        *        *     Request timed out.
 12     *        *        *     Request timed out.
 13     *        *        *     Request timed out.
 14     *        *        *     Request timed out.
 15     *        *        *     Request timed out.
 16     *        *        *     Request timed out.
 17     *        *        *     Request timed out.
 18     *        *        *     Request timed out.
 19     *        *        *     Request timed out.
 20     *        *        *     Request timed out.
 21     *        *        *     Request timed out.
 22     *        *        *     Request timed out.
 23     *        *        *     Request timed out.
```

On Linux

```
parsa@parsa-sigware: ~ — tracepath 192.168.1.50

parsa@parsa-sigware:~$ tracepath 192.168.1.50
 1?: [LOCALHOST]                      pmtu 1500
 1:  _gateway                                              0.540ms
 1:  _gateway                                              0.304ms
 2:  no reply
 3:  no reply
 4:  no reply
 5:  no reply
 6:  no reply
 7:  no reply
 8:  no reply
 9:  no reply
10:  no reply
11:  no reply
12:  no reply
13:  no reply
14:  no reply
15:  no reply
16:  no reply
17:  no reply
18:  no reply
```

Use the ipconfig/ifconfig command to display to display your devices network configuration

Identify IP address , subnet mask , default gateway .

IPv4 Address → your device's unique identifier on the LAN

Subnet Mask → defines the size of the local network; here it allows 254 hosts

Default Gateway →usually your router, the path out of the LAN

On windows

ipconfig

```
Command Prompt
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 9:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::9ce6:a2ba:4b5:d766%10
   IPv4 Address. . . . . . . . . . . : 192.168.87.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f1d3:a0ab:18ad:d3b6%5
   IPv4 Address. . . . . . . . . . . : 192.168.31.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Wi-Fi:
```

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2021:2b4c:5da6:d251%19
   IPv4 Address. . . . . . . . . . . : 10.228.168.195
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.228.168.110
```

On linux

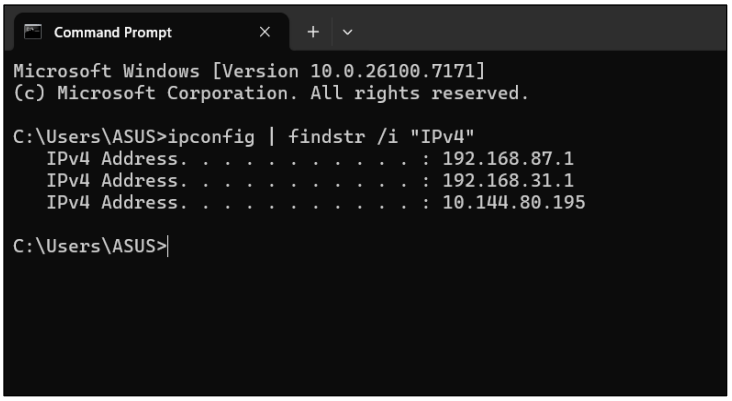
ip addr show

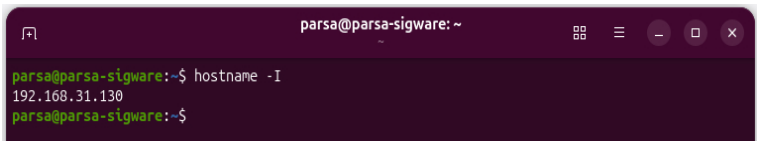
```
parsa@parsa-sigware: ~

parsa@parsa-sigware:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
up default qlen 1000
    link/ether 00:0c:29:21:58:bd brd ff:ff:ff:ff:ff:ff
    inet 192.168.31.130/24 brd 192.168.31.255 scope global dynamic noprefixroute
 ens33
       valid_lft 1720sec preferred_lft 1720sec
    inet6 fe80::20c:29ff:fe21:58bd/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

```
parsa@parsa-sigware:~$ ip route | grep default
default via 192.168.31.2 dev ens33 proto dhcp src 192.168.31.130 metric 100
parsa@parsa-sigware:~$
```

Modify the command to display only the IP address of your device

On windows | On Linux

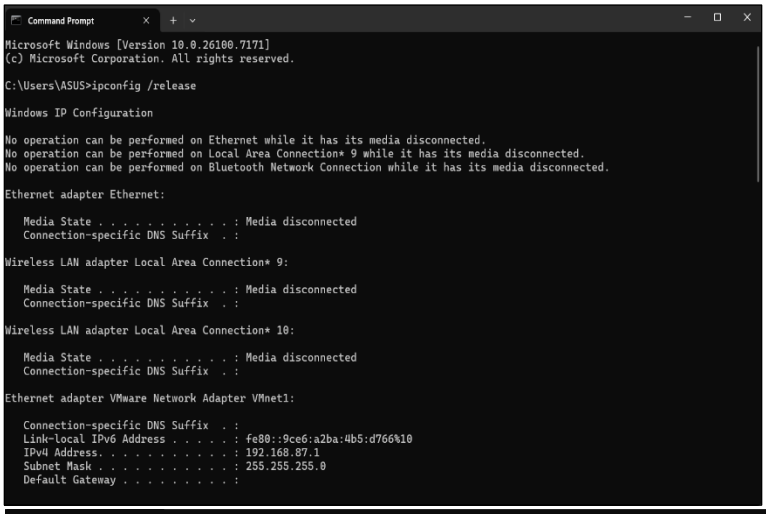


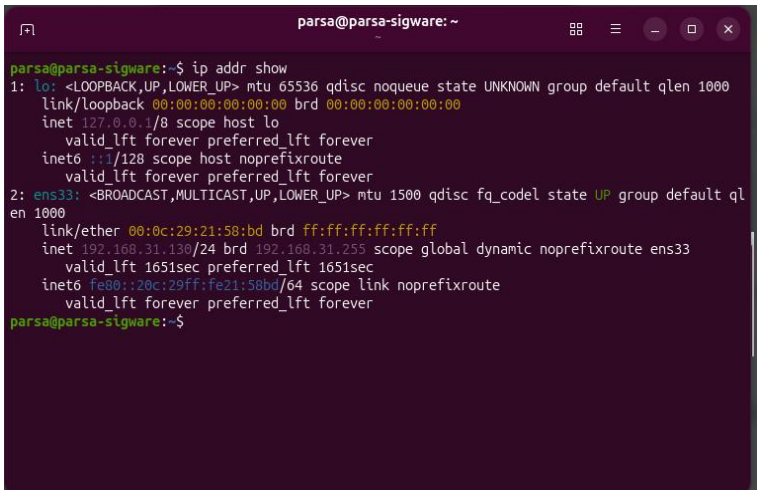


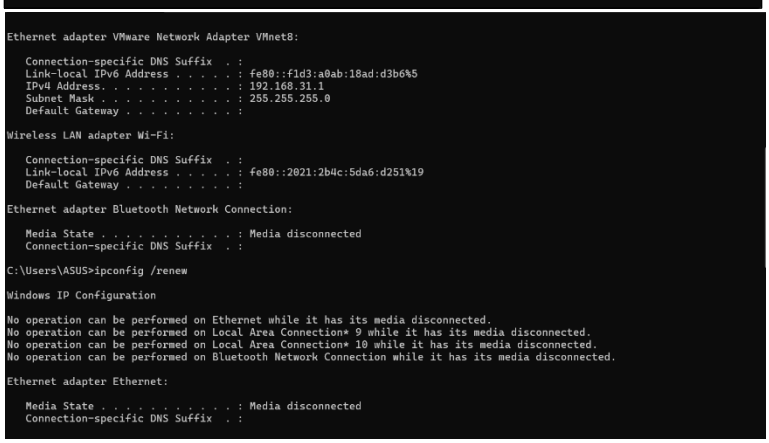
The hostname -I command (Linux/Unix) prints the IP addresses assigned to your machine.It shows all network interfaces' addresses, separated by spaces, without extra details like subnet mask or gateway.

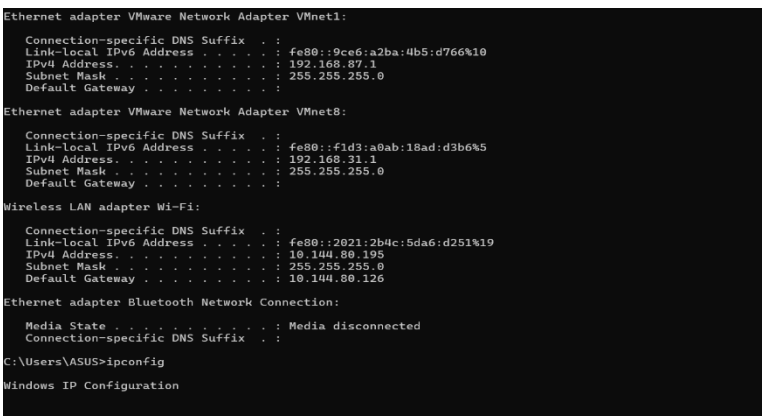
Use the ipconfig/ifconfig to release and renew the IP address observe the changes In IP and configuration .

On windows | On Linux



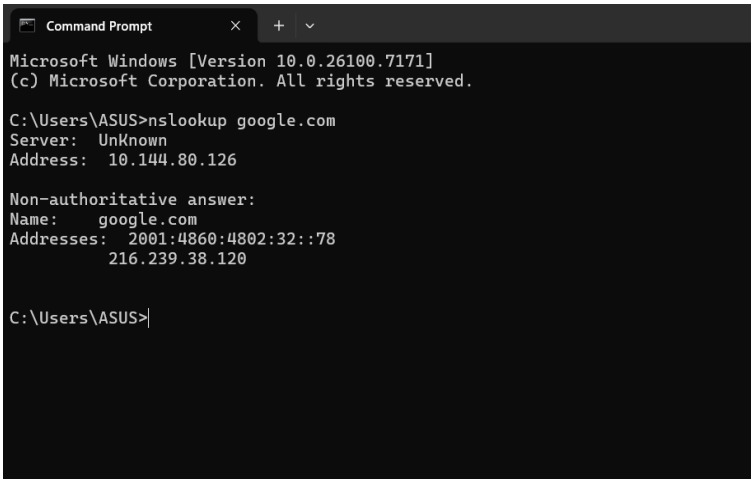






Use nslookup to resolve the IP address of a specific domain (e.g.google.com)

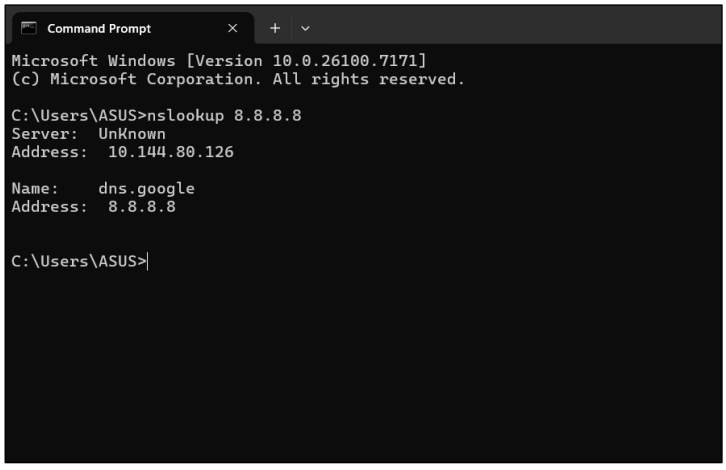
On windows



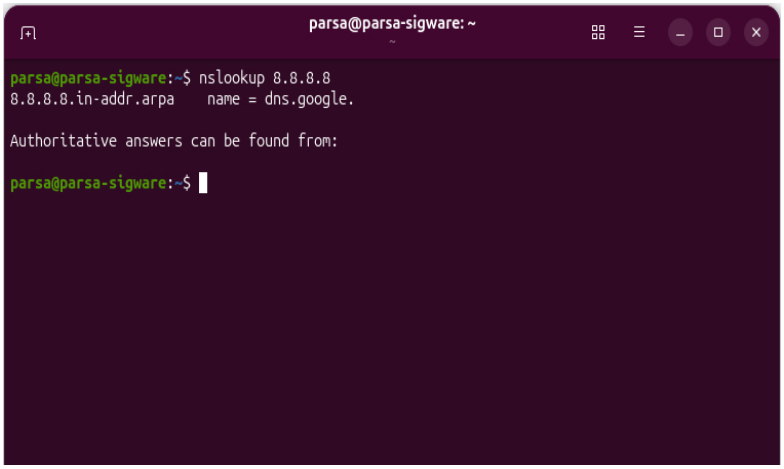
On Linux

Modify nslookup to perform a reverse DNS lookup using an IP address instead of a domain name

On windows

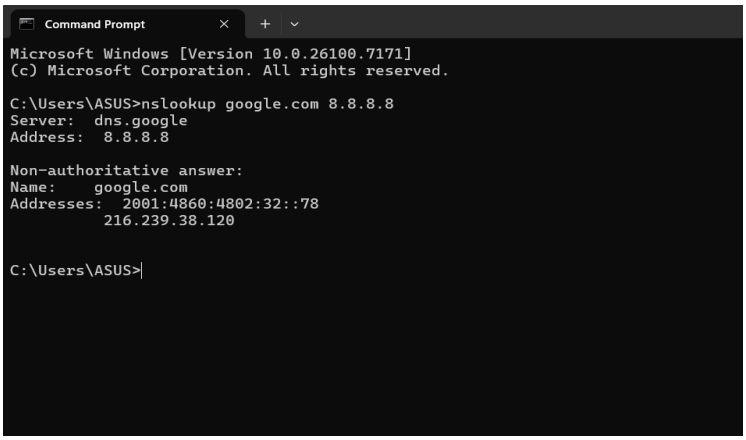


On Linux

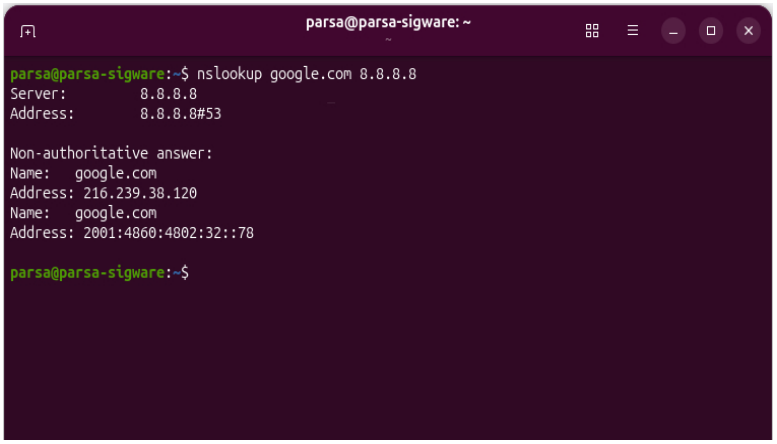


Query a specific DNS server using to get IP address of a domain name

On windows



On Linux



Identify and explain the purpose of common HTTPS methods :

GET / POST / PUT /DELETE

GET → To retrieve data from a server. When you visit a webpage or request information

Safe! Doesn't change server data. / Multiple identical requests have the same effect. / gets parameters sent in the URL (query string).

POST → To submit data to the server to create a new resource.

To submitting a form, uploading a file, or creating a new user / Repeating the same request may create multiple resources. / Sent in the request body, not visible in the URL.

POST → Update or replace an existing resource.

/ For updating a user profile or replacing a document /Repeating the same request results in the same state. / Sent in the request body.

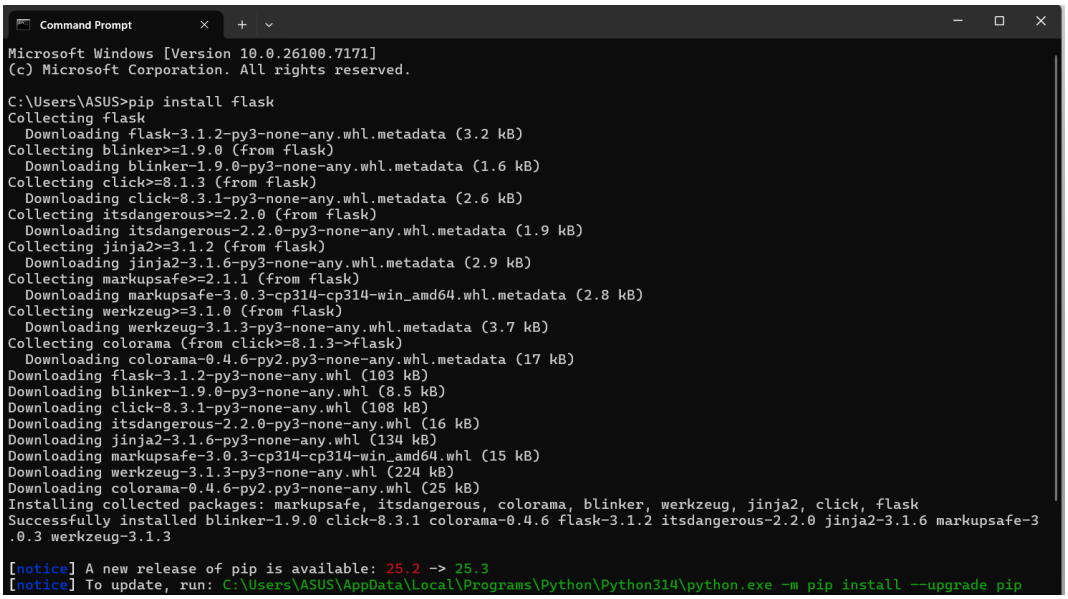
DELETE → Remove a resource from the server.

For deleting a user or a post / Deleting the same resource multiple times has the same effect — it's gone.

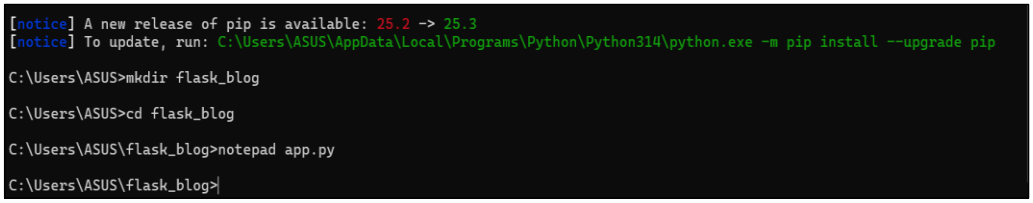
Design a new simple a web application that use these HTTPS methods for CRUD operations

On a resource (e.g. , a blog post)

Step 1 (on python): Download flask



Step 2 : write your apps code



Here writing by using mkdir folder and selecting it cd folder

Then by using notepad app.py we have a page for writing our code

Step 3 : RUNNING IT ! and then using curl

```
curl -X POST http://localhost:5000/posts ^
  -H "Content-Type: application/json" ^
  -d "{\"title\":\"Hello\",\"content\":\"Python version\"}"
```

Use postman or curl to send HTTPS request to a RESTful API and observe the response

Open Postman.

Set method (GET, POST, etc.).

Enter API URL (e.g.,).

Add headers (like).

Add body (for POST/PUT requests).

Click "Send".

View the response (status, headers, body).

curl -X GET
https://api.example.com/data

2) curl -X POST
**https://api.example.com/data **

**-H "Content-Type: application/json" **

-d '{"key":"value"}'

Explain the role of TCP/IP protocol suite.

The TCP/IP protocol suite is the foundation of modern internet communication, enabling reliable data exchange between devices across networks.

TCP/IP (Transmission Control Protocol/Internet Protocol) is a set of layered communication protocols used to connect devices on the internet and other networks. It defines how data is packaged, addressed, transmitted, routed, and received

Standardized Communication: It provides a universal framework for data exchange, allowing different systems and devices to communicate seamlessly.

Layered Architecture:TCP/IP is

TCP/IP is organized into four layers, each with specific responsibilities: | Layer | Role | |----------|---------- ----------| | Application | Interfaces with software (e.g., HTTP, FTP, SMTP) | | Transport | Ensures reliable data delivery (TCP) or fast delivery (UDP) | | Internet | Handles addressing and routing (IP) | | Network Access | Manages physical transmission over hardware (Ethernet, Wi-Fi) |

Reliable Data Transfer :TCP breaks data into packets, ensures they arrive correctly and in order, and reassembles them at the destination.

Routing and Addressing: IP assigns unique addresses to devices and routes packets across networks to reach the correct destination.

Scalability and Interoperability: TCP/IP supports large-scale networks and works across diverse hardware and operating systems.

Give an IP address and subnet mask , calculate :

Network address / broadcast address / range of valid host Ips

IP Address: 10.0.5.17

Subnet Mask: 255.255.255.240 (A)

Subnet mask A means 16 IP addresses per subnet (2^32-28)

Each block starts at a multiple of 16:

10.0.5.17,10.0.5.17 , 10.0.5.13, etc.

2. Find the Subnet Block

10.0.5.17 falls in the block starting at 10.0.5.16

3. Network Address

First IP in the block: 10.0.5.16 **4. Broadcast Address**

Last IP in the block: 10.0.5.31

5. Valid Host IP Range

Exclude network and broadcast addresses:

10.0.5.17 to 10.0.5.30

Analyze a network topology and identify :

Routers / switches /

IP addresses used for communication

Routers

Connect different networks (e.g., home to internet) / Assign IP addresses and route traffic between networks

Switches

Connect devices within the same network (LAN) / Forward data based on MAC addresses

IP Addresses

Unique identifiers for devices on a network / Used for communication between

Explain the purpose of DNS and show how it translates domain names to IP address

It simplifies access: DNS lets users type easy-to-remember names instead of numeric IP addresses , Enables scalability: It supports billions of domain lookups daily, making the internet usable and efficient , Acts like a phonebook: It maps domain names to IP addresses so browsers can find the correct servers.

User Input: You enter a domain name (google) in your browser.

Local Cache Check: Your device checks its DNS cache for a recent IP match.

Recursive Resolver: If not cached, the request goes to a DNS resolver (usually provided by your ISP).

Root Server Query: The resolver asks a root DNS server where to find domains.

TLD Server Query: The resolver then queries the top-level domain server for .

Authoritative Server: Finally, the resolver contacts the authoritative DNS server for , which returns the correct IP address.

Response: The resolver sends the IP back to your browser, which uses it to connect to the website.

Use nslook up to resolve the IP address of a domain name and analyze the returned DNS records

Server: DNS server used for the query (e.g., is Google DNS)

Name: The domain name queried , Address: The resolved IP address (IPv4 or IPv6)

Non-authoritative answer: Response from a DNS cache, not the original source You can also query specific record types:

```
C:\Users\ASUS>nslookup -type=MX example.com   # Mail exchange records
Usage:
   nslookup [-opt ...]             # interactive mode using default server
   nslookup [-opt ...] - server    # interactive mode using 'server'
   nslookup [-opt ...] host        # just look up 'host' using default server
   nslookup [-opt ...] host server # just look up 'host' using 'server'

C:\Users\ASUS>nslookup -type=NS example.com   # Name server records
Usage:
   nslookup [-opt ...]             # interactive mode using default server
   nslookup [-opt ...] - server    # interactive mode using 'server'
   nslookup [-opt ...] host        # just look up 'host' using default server
   nslookup [-opt ...] host server # just look up 'host' using 'server'

C:\Users\ASUS>nslookup -type=TXT example.com  # Text records
```

Configure a local DNS server to resolve custom domain names on a local network and test name resolution

Configure Local DNS Server STEPS

1. **Install DNS server software (e.g., on Linux).**
2. **Define a new DNS zone in the DNS config file.**
3. **Create a zone file with custom domain and IP mappings.**
4. **Set correct file permissions for the zone file.**
5. **Restart the DNS service to apply changes.**
6. **Configure client devices to use your DNS server IP**