

Q1: explain how to get the private IP?

By command in MacOS `ipconfig getifaddr en0` (Wi-Fi)

```
[yousefabdullah@MacBook-Air-Yousef ~ % ipconfig getifaddr en0  
192.168.100.52
```

Q2: What is the difference between `ipconfig` & `ipconfig/all`?

`Ipconfig` in MacOSes ---> `ipconfig getifaddr en0` Show IPv4 address only

Shows basic network configuration

`Ipconfig/all` in MacOS ---> `ifconfig`

Shows full detailed network configuration

```
[yousefabdullah@MacBook-Air-Yousef ~ % ifconfig
```

```
lo0: flags=8049<UP,LOOPBACK,RUNNING> mtu 1280  
    options=120<SOCKETS,TIMEWHEEL>  
    inet 127.0.0.1 netmask 0xfffc  
    inet6 ::1 prefixlen 64  
    inet6 fe80::1%lo0 prefixlen 64  
    nd6 options=201<PERFORM,NODEADSTMT>  
gif0: flags=8010<POINTOPOINT,NOTRAFFIC> mtu 1280  
stf0: flags=0<> mtu 1280  
anpi0: flags=8863<UP,LOOPBACK,RUNNING> mtu 1280  
    options=400<SOCKETS,TIMEWHEEL>  
    ether 5a:ea:00:00:00:00  
    media: none  
    status: inactive  
anpi1: flags=8863<UP,LOOPBACK,RUNNING> mtu 1280  
    options=400<SOCKETS,TIMEWHEEL>  
    ether 5a:ea:00:00:00:00  
    media: none  
    status: inactive  
en3: flags=8863<UP,LOOPBACK,RUNNING> mtu 1280  
    options=400<SOCKETS,TIMEWHEEL>  
    ether 5a:ea:00:00:00:00  
    nd6 options=201<PERFORM,NODEADSTMT>  
    media: none  
    status: inactive  
en4: flags=8863<UP,LOOPBACK,RUNNING> mtu 1280  
    options=400<SOCKETS,TIMEWHEEL>  
    ether 5a:ea:00:00:00:00  
    nd6 options=201<PERFORM,NODEADSTMT>  
    media: none  
    status: inactive  
en1: flags=8963<UP,LOOPBACK,RUNNING> mtu 1280  
    options=460<SOCKETS,TIMEWHEEL,SMARTDRIVER>  
    ether 36:6d:00:00:00:00  
    media: auto  
    status: inactive  
en2: flags=8963<UP,LOOPBACK,RUNNING> mtu 1280  
    options=460<SOCKETS,TIMEWHEEL,SMARTDRIVER>  
    ether 36:6d:00:00:00:00  
    media: auto  
    status: inactive
```

```
bridge0: flags=8863
        options=63<
        ether 36:6d
        Configurati
            id
            max
            roo
            ipf
        member: en1
            ifm
        member: en2
            ifm
        nd6 options
        media: <unk
        status: ina
utun0: flags=8051<U
        inet6 fe80:
        nd6 options
ap1: flags=8863<UP,
        options=646
        ether 0a:1c
        nd6 options
        media: auto
        status: ina
en0: flags=8863<UP,
        options=646
        ether 9e:d4
        inet6 fe80:
        inet 192.16
        nd6 options
        media: auto
        status: act
awdl0: flags=8863<U
        options=646
        ether 02:76
        inet6 fe80:
        nd6 options
        media: auto
        status: act
llw0: flags=8863<UP
        options=400
        ether 02:76
        inet6 fe80:
        nd6 options
        media: auto
```

```
utun1: flags=8051
        inet6 fe8
        nd6 optio
utun2: flags=8051
        inet6 fe8
        nd6 optio
utun3: flags=8051
        inet6 fe8
        nd6 optio
```

Q3: explain how to get the public IP?

use this Command in MacOS and Windows ---> curl ifconfig.me

```
yousefabdullah@MacBook-Air-Yousef ~ % curl ifconfig.me
```

```
41.39.205.223%
```

use this website <https://www.whatismyip.com>

What Is My IP?

My IPv4: [41.39.205.223](https://www.whatismyip.com) 

My IPv6: Not Detected 

Q4: What is the difference between public ip & private ip?

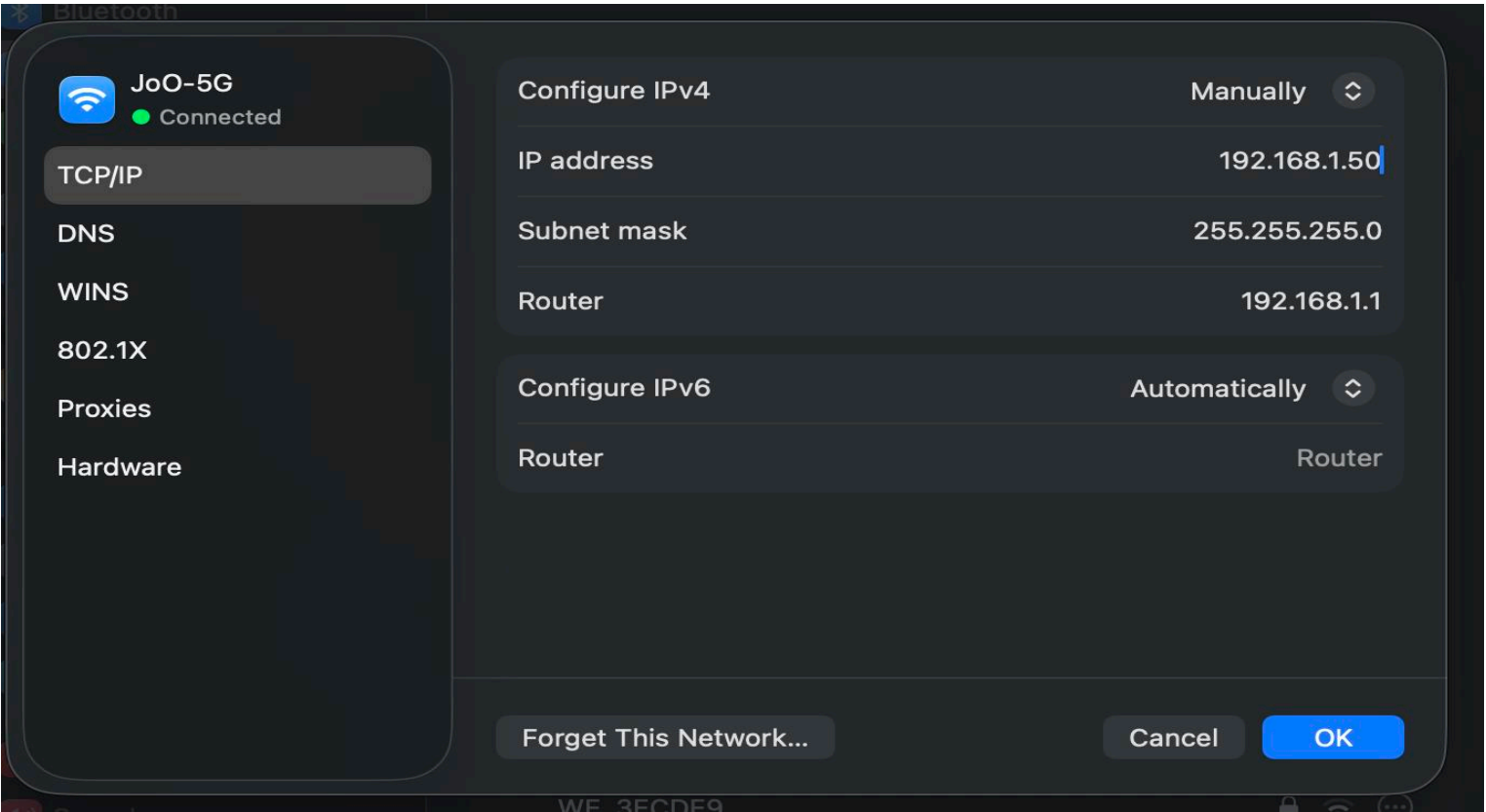
Public ip	Private ip
Used within a local or private network	Used for communication over the internet
Not routable on the public internet	Routable on the public internet
Scope is limited to the local network	Scope is global
Assigned by router or DHCP server	Assigned by ISP
Unique within a local network	Globally unique
Requires NAT for internet access	Does not require NAT
Hidden from external networks	Visible on the internet
Uses reserved private IP ranges	Uses globally assigned IP ranges
Range: 10.0.0.0 to 10.255.255.255 172.16.0.0 to 172.31.2525.255 192.168.0.0 to 192.168.255.255	Range: 1.0.0.0 to 9.255.255.255 11.0.0.0 to 126.255.255.255 128.0.0.0 to 172.15.255.255 172.32.0.0 to 197.255.255.255 192.0.0.0 to 192.167.255.255 192.169.0.0 to 223.255.255.255
Example: 192.168.1.10	Example: 203.0.113.10

Nat is used to Translate the private IP address to public IP addresses

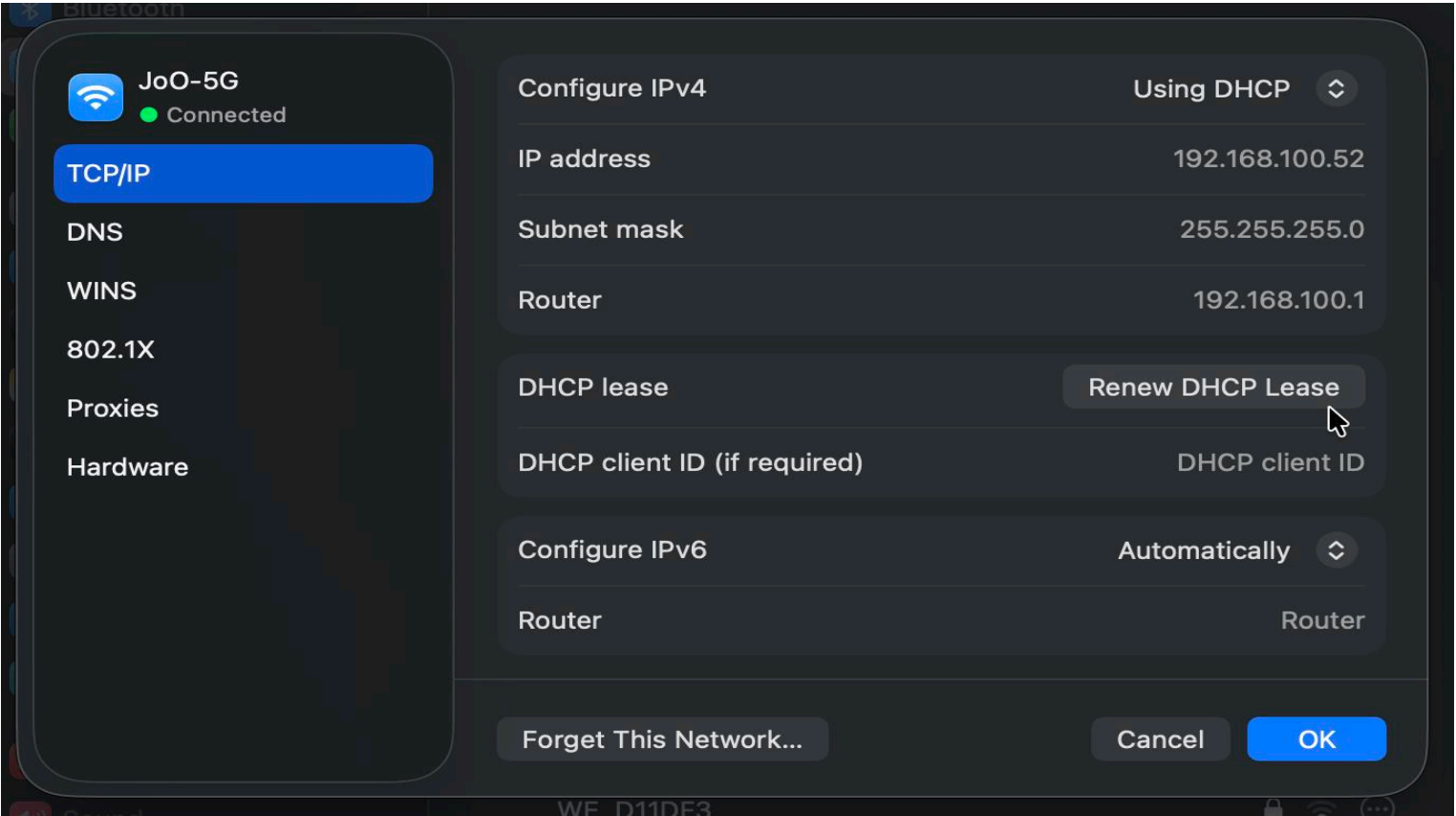
Q5: How does the device get its IP?

- Manually
- Automatic (By DHCP)
- APIPA (Random /Rang : 169.254.X.X)

Q6: Make your device get its private IP statically



Q7: Reset your device to get its private IP automatically



Q8: What do you know about APIPA Address ?

APIPA stands for **Automatic Private IP Addressing**. It is a networking feature that allows a device to **automatically assign itself a private IP address when it cannot get one from a DHCP server**.

How Does APIPA Work?

- When a device tries to get an IP using DHCP but **fails to contact a DHCP server**, it picks an IP address **automatically** from a special range.
- The APIPA range is: **169.254.0.1 to 169.254.255.254** with subnet mask **255.255.0.0**.
- The device **self-assigns** this IP so it can still communicate with other devices on the same local network that also have APIPA addresses.
- APIPA **does not provide internet access** or a default gateway because it's only for local communication.

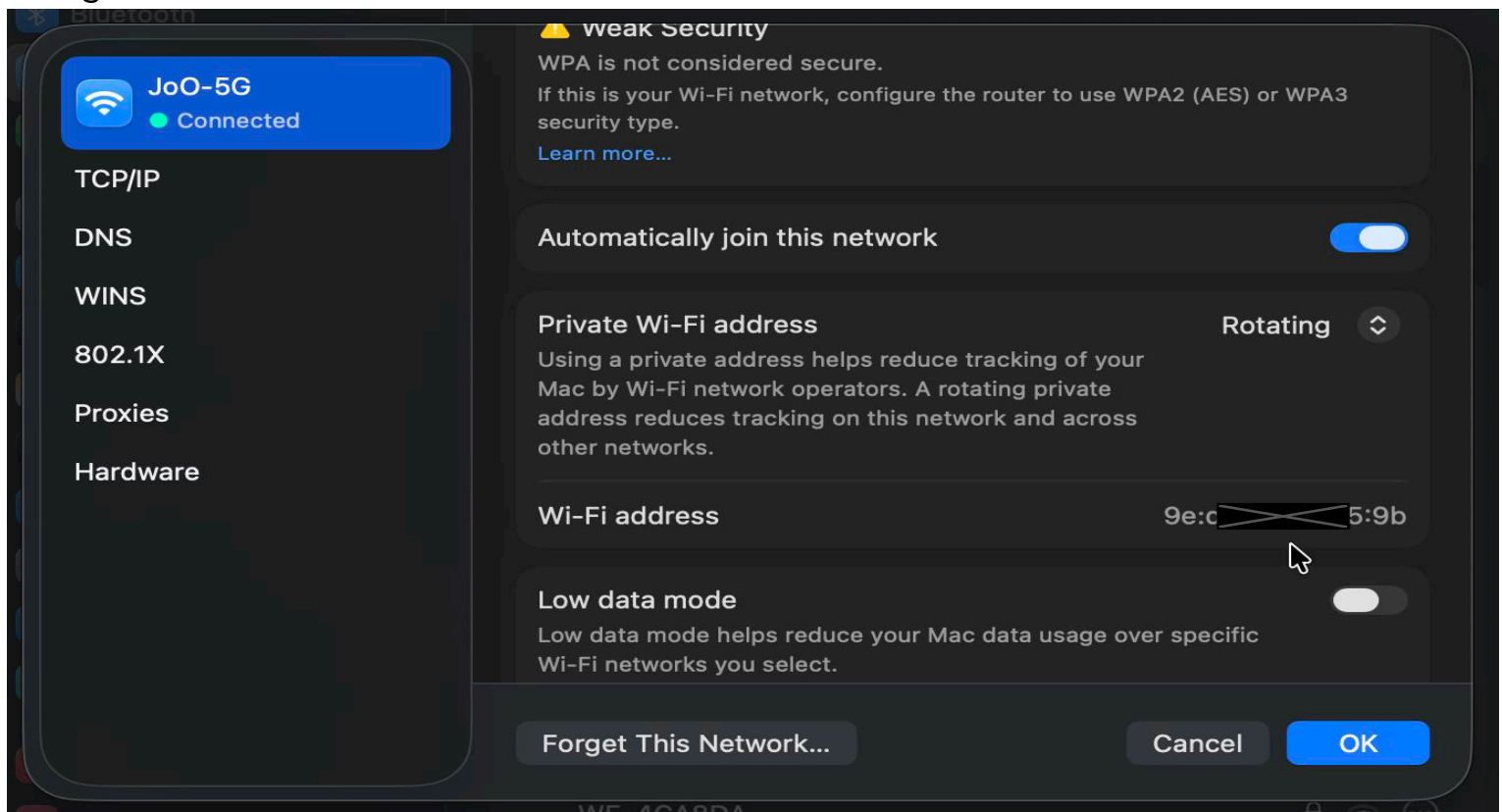
Q9: Give me 2 ways to find out your device's MAC address.

Using CLR: macOSes ---> (ifconfig en0 | grep ether) & Windows ---> (getmac)

```
yousefabdullah@MacBook-Air-Yousef ~ % ifconfig en0 | grep ether
```

```
ether 9e:d5:9b
```

Using GUI:



Q10: What is the difference between getmac & getmac/v ?

Feature	getmac	getmac /v
Verbosity	Basic/Minimal	Detailed (Verbose)
Information	Physical Address & Transport Name	Adds Adapter Name, Status, and Connection Type
Readability	Low (hard to identify adapters)	High (easy to identify adapters)
Virtual Adapters	Displays all active/connected	Displays all (including inactive)

Q16. Can you tell me which command that could check connectivity between 2 devices?

ping <ip address>

And then:

a) check connectivity of your loopback IP address “127.0.0.1”.

```
yousefabdullah@MacBook-Air-Yousef ~ % ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.118 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.170 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.176 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.149 ms
^C
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.118/0.153/0.176/0.023 ms
```

b) How many packets are sent to check availability?

Windows: sends 4 packets

macOS: keeps sending packets until you stop it (Ctrl + C)

I can set the number of packets: macOS: ping -c 4 127.0.0.1 sends 4 packets

```
yousefabdullah@MacBook-Air-Yousef ~ % ping -c 4 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.070 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.113 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.170 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.168 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.070/0.130/0.170/0.042 ms
```


c) How many packages did your device receive? What does this mean?

- Packets received = number of replies from the target IP
- If all packets are received **connectivity is working**
- If some are lost **network issue** or device not reachable

```
--- 127.0.0.1 ping statistics ---  
4 packets transmitted, 4 packets received, 0.0% packet loss
```

In this all 4 packets is working

d) Check connectivity between your device and this IP "10.10.0.10". then explain the result

```
[yousefabdullah@MacBook-Air-Yousef ~ % ping -c 4 10.10.0.10  
PING 10.10.0.10 (10.10.0.10): 56 data bytes  
Request timeout for icmp_seq 0  
Request timeout for icmp_seq 1  
Request timeout for icmp_seq 2  
  
--- 10.10.0.10 ping statistics ---  
4 packets transmitted, 0 packets received, 100.0% packet loss
```

Device is not reachable: 0 packets received 100% packet loss

Q17. Verify the connectivity of the loopback IP address "127.0.0.1" by sending "8" packets which the size of each packet is "50000".

MacOS --> ping -c 8 -s 50000 127.0.0.1

Windows --> ping 127.0.0.1 -n 8 -l 50000

```
[yousefabdullah@MacBook-Air-Yousef ~ % ping -c 8 -s 50000 127.0.0.1  
PING 127.0.0.1 (127.0.0.1): 50000 data bytes  
ping: sendto: Message too long  
ping: sendto: Message too long  
Request timeout for icmp_seq 0  
ping: sendto: Message too long  
Request timeout for icmp_seq 1  
ping: sendto: Message too long  
Request timeout for icmp_seq 2  
ping: sendto: Message too long  
Request timeout for icmp_seq 3  
ping: sendto: Message too long  
Request timeout for icmp_seq 4  
ping: sendto: Message too long  
Request timeout for icmp_seq 5  
ping: sendto: Message too long  
Request timeout for icmp_seq 6  
  
--- 127.0.0.1 ping statistics ---  
8 packets transmitted, 0 packets received, 100.0% packet loss
```

Q18. Explain what is the meaning of this command: “ping 127.0.0.1 -t “

The command ping 127.0.0.1 -t sends ICMP packets continuously to the loopback address of the local device, checking the internal network stack and network stability until stopped manually --> press ctr+c

In MacOS --> ping 127.0.0.1

```
yousefabdullah@MacBook-Air-Yousef ~ % ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.115 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.191 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.176 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.372 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.222 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.131 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.181 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.176 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.201 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.198 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.191 ms
^C
--- 127.0.0.1 ping statistics ---
11 packets transmitted, 11 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.115/0.196/0.372/0.063 ms
```

Q19. What is the meaning of “ DOS Attack” ?

A **Denial of Service (DoS) attack** is an unauthorized attack that attempts to make a system or service unavailable by consuming system and network resources.

It uses up resources like:

- Memory
- CPU
- Network bandwidth

Example: **Ping of Death** – sending too many or too large ping packets that crash the system.

Result: Legitimate users **cannot access the service**.