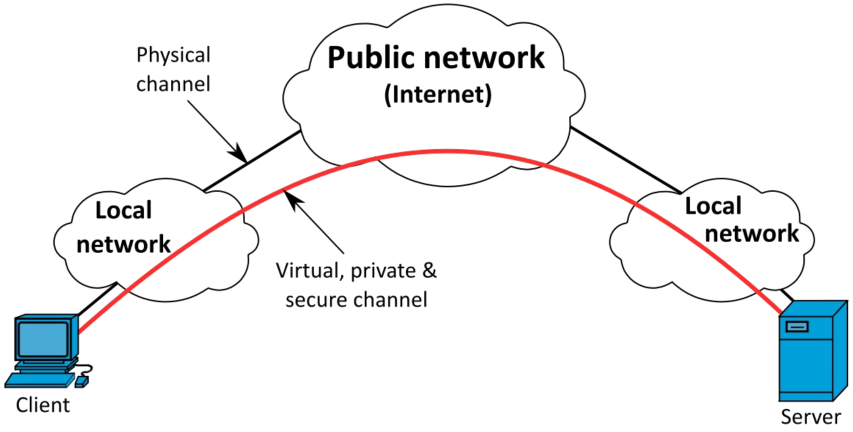
VPN

D.Krishna (17MCME09)
R.Indu(17MCME12)
J.Yagnahaun(17MCME22)

What is a VPN?

- A virtual private network (VPN) gives you online privacy and anonymity by creating a private network from a public internet connection.
- VPNs mask your internet protocol (IP) address so your online actions are virtually untraceable.
- Most important, VPN services establish secure and encrypted connections to provide greater privacy than even a secured Wi-Fi hotspot.

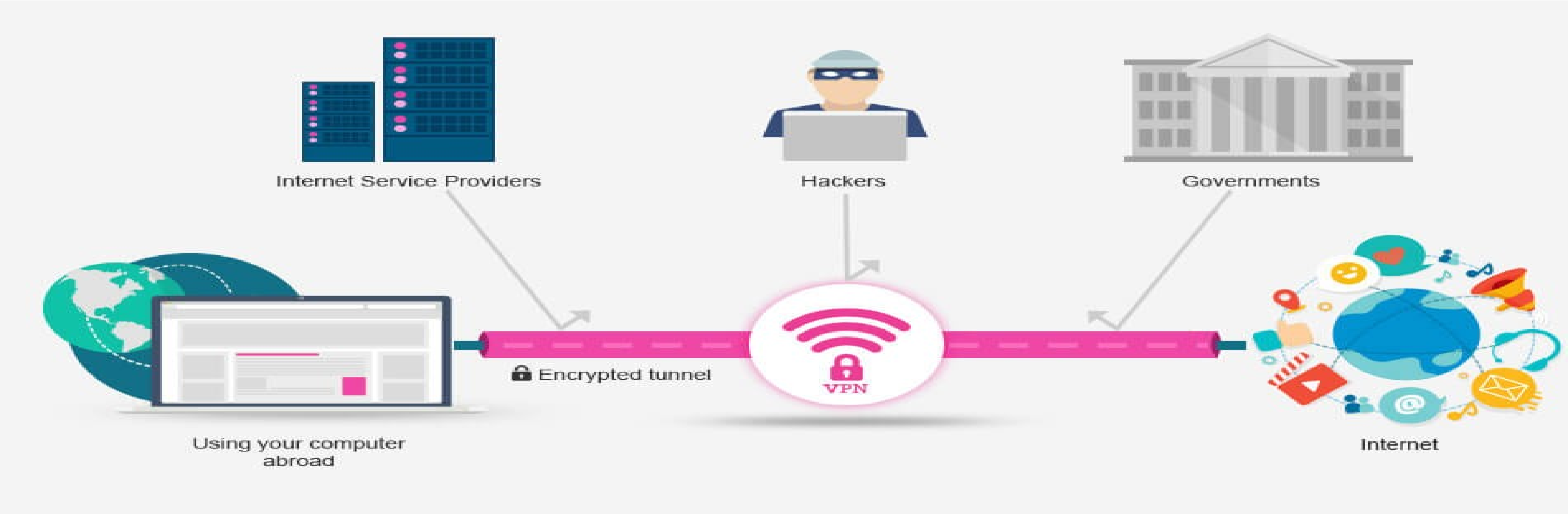


Why do you need a VPN service?

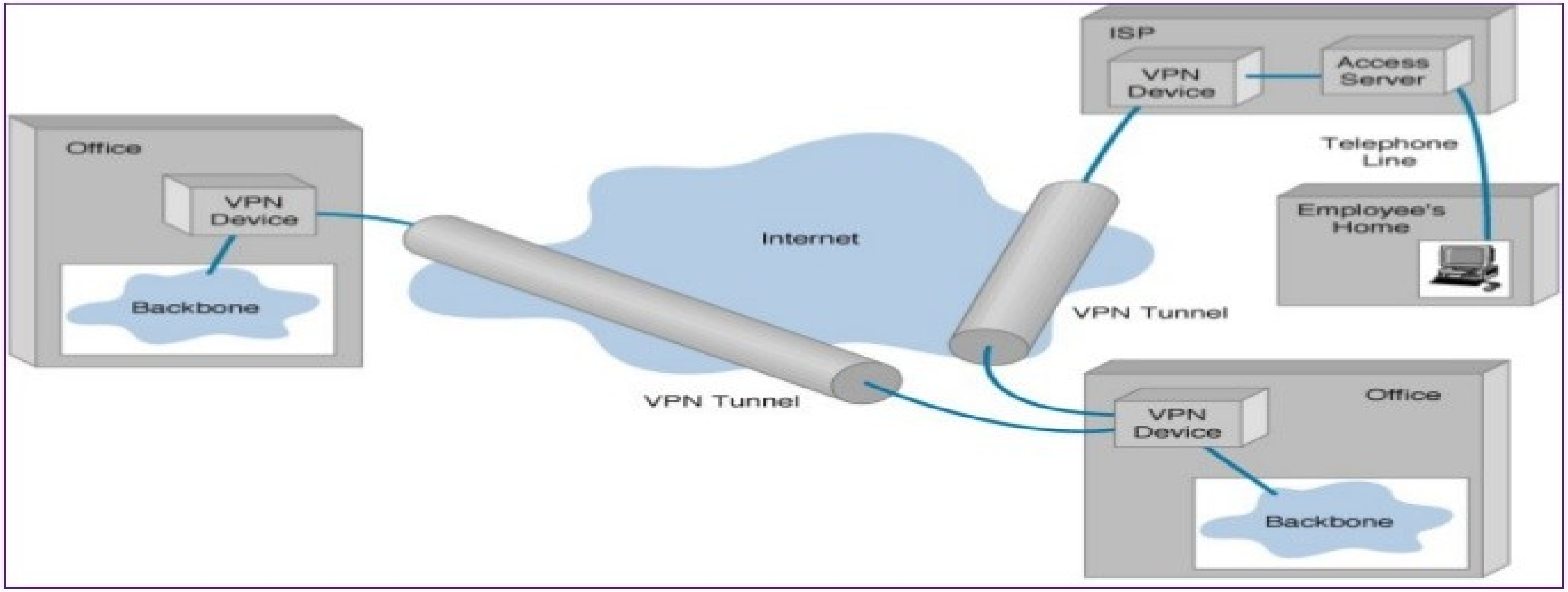
- Surfing the web or transacting on an unsecured Wi-Fi network means you could be exposing your private information and browsing habits.
- That's why a virtual private network, better known as a VPN, should be a must for anyone concerned about their online security and privacy.
- The encryption and anonymity that a VPN provides helps protect your online activities: sending emails, shopping online, or paying bills. VPNs also help keep your web browsing anonymous.

How a VPN protects your IP address and privacy

- VPNs essentially create a data tunnel between your local network and an exit node in another location, which could be thousands of miles away, making it seem as if you’re in another place.
- VPNs use encryption to scramble data when it’s sent over a Wi-Fi network. Encryption makes the data unreadable.

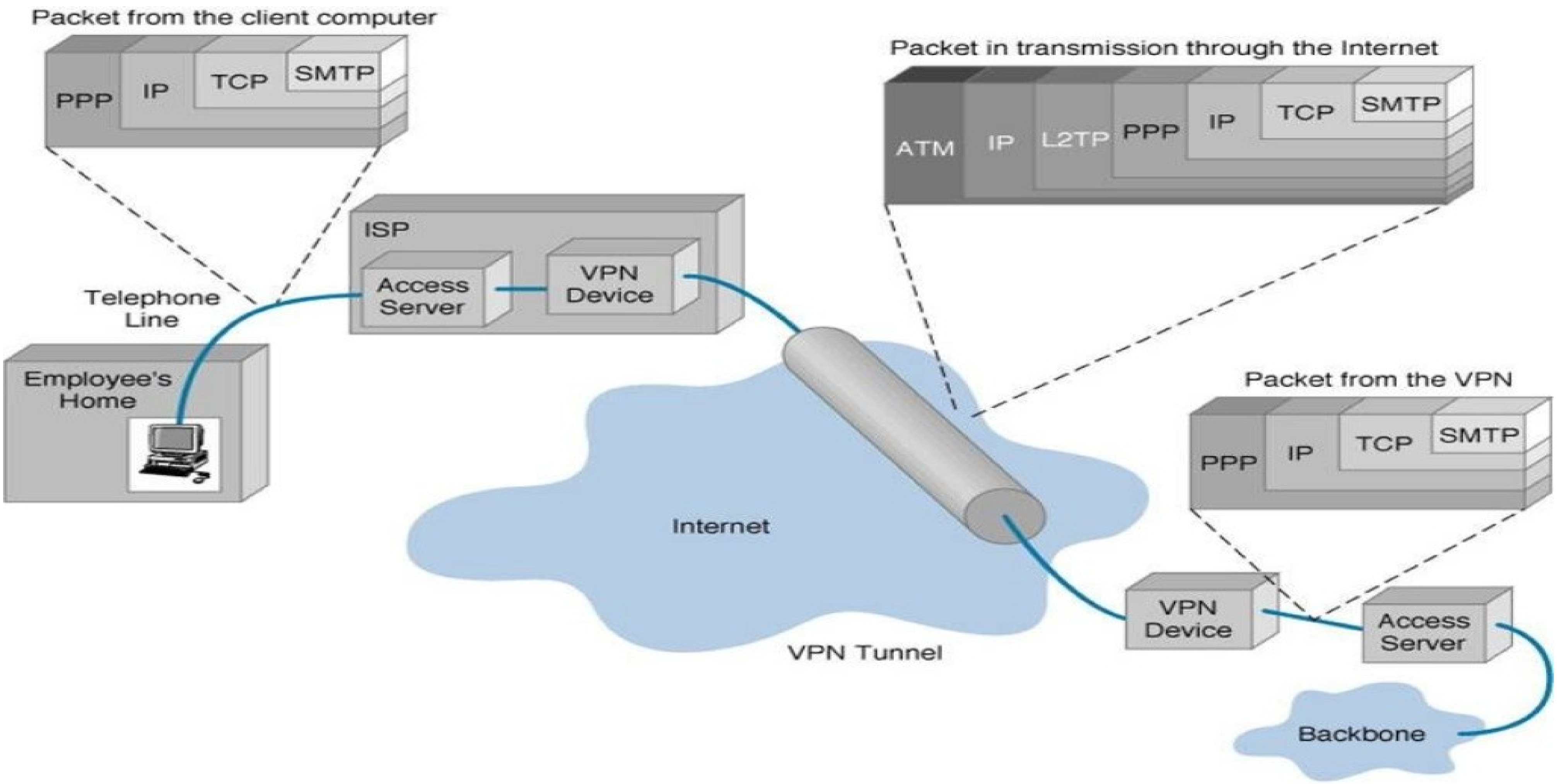


VPN Basic Architecture



Tunneling

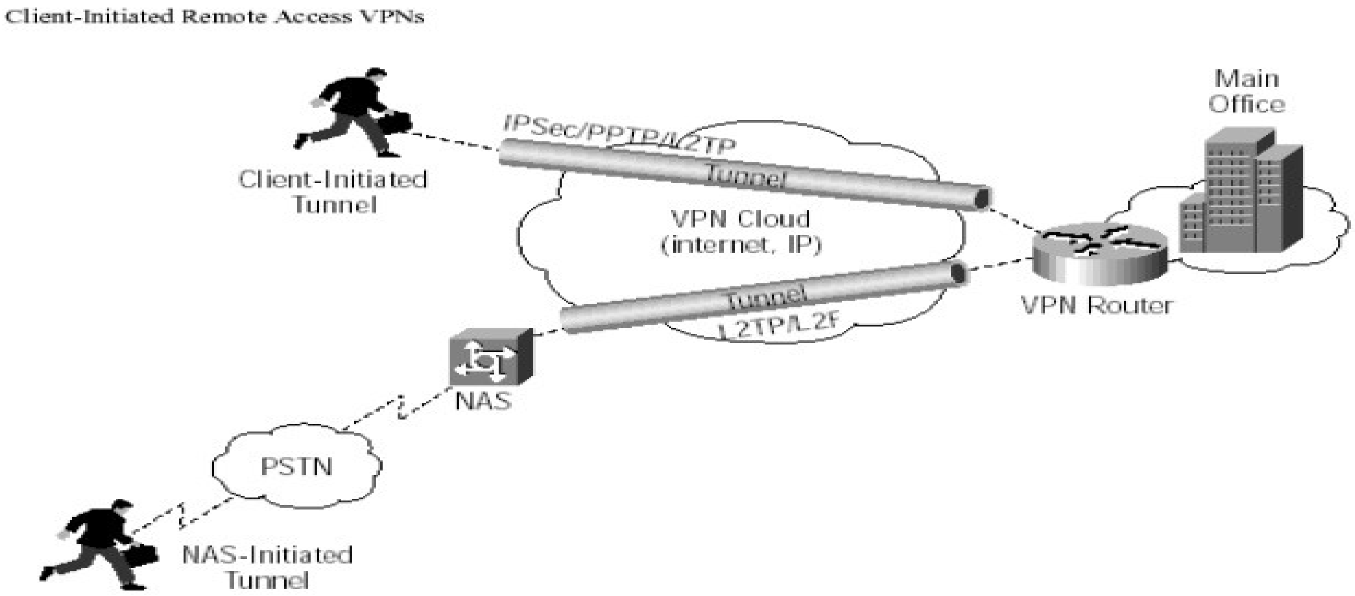
- VPN technology is based on the idea of tunneling.
- VPN tunneling invloves establishing and maintaining a logical network connection.
- Tunneling is the process of placing an entire packet within another packet before it’s transported over the Internet.
- That outer packet protects the contents from public view and ensures that the packet moves within a virtual tunnel.


Packet from the client computer
PPP
IP
TCP
SMTP
ISP
Access Server
VPN Device
Telephone Line
Employee's Home
Packet in transmission through the Internet
ATM
IP
L2TP
PPP
IP
TCP
SMTP
Internet
VPN Tunnel
Packet from the VPN
PPP
IP
TCP
SMTP
VPN Device
Access Server
Backbone

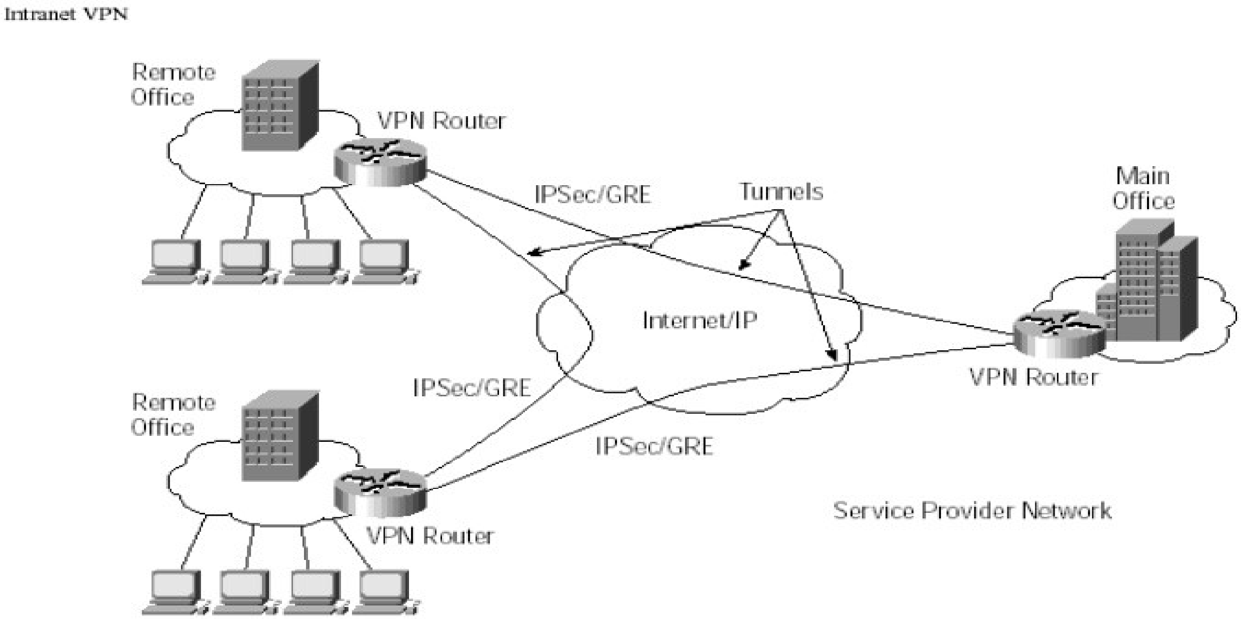
Types of VPN

- Remote access VPN
- Intranet VPN
- Extranet VPN

Remote access VPN

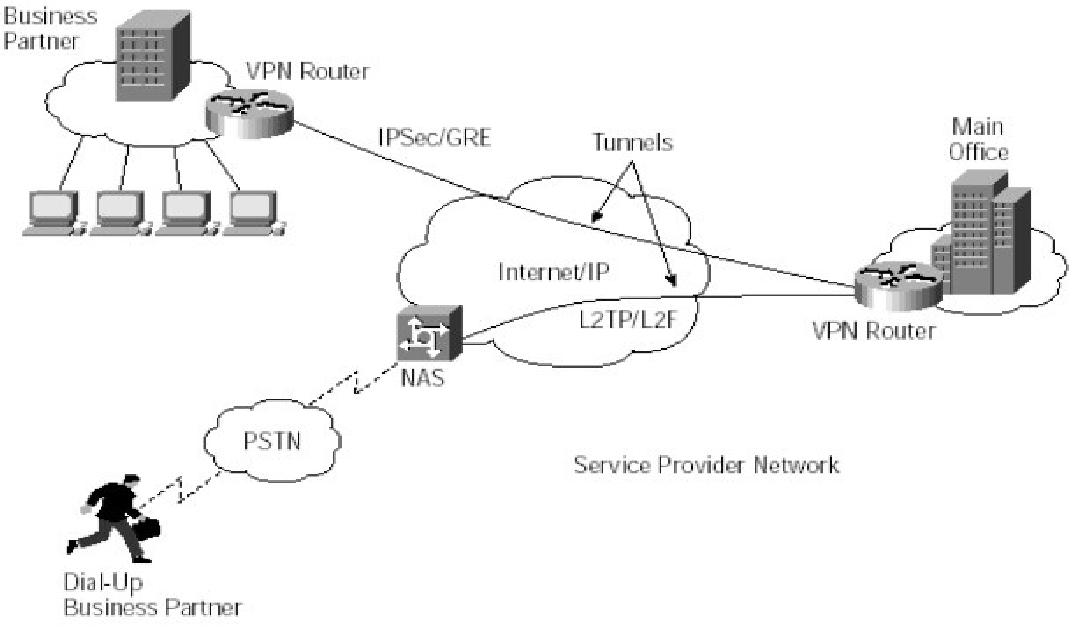


Intranet VPN



Extranet VPN

Extranet VPN



VPN privacy: What does a VPN hide?

A VPN can hide a lot of information that can put your privacy at risk. Here are five of them.

1. Your browsing history
2. Your IP address and location
3. Your location for streaming
4. Your devices
5. Your web activity — to maintain internet freedom

How to choose a VPN

A smart way to stay secure when using public Wi-Fi is to use a VPN solution. But what’s the best way to choose a virtual private network? Here are some questions to ask when you’re choosing a VPN provider.

1. Do they respect your privacy?

2. Do they run the most current protocol?

3. Do they set data limits?

4. Where are the servers located?

5. Will you be able to set up VPN access on multiple devices?

6. How much will it cost?

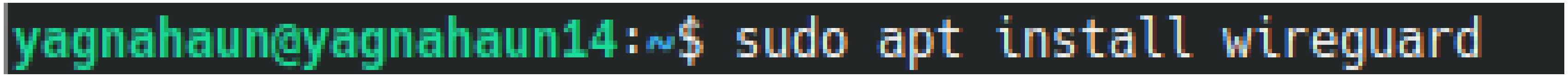
Paid vs Free

We can see that free VPNs:

- don’t offer the most current or secure protocols
- don’t offer the highest bandwidth and connection speeds to free users
- do have a higher disconnection rate
- don’t have as many servers in as many countries globally
- don’t offer support

Wireguard VPN

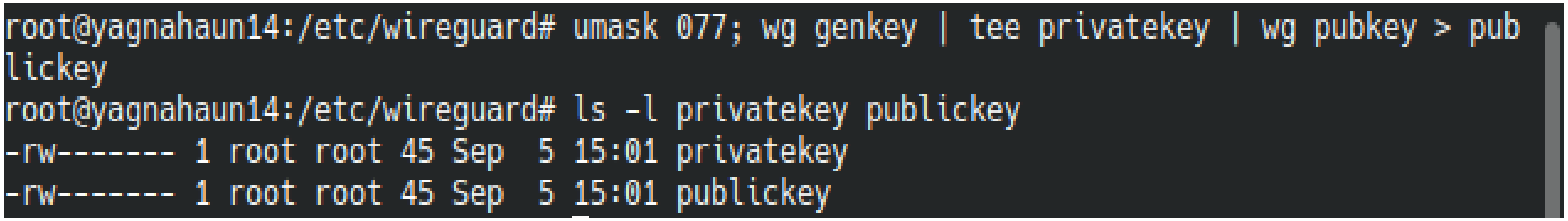
- WireGuard is a security-focused virtual private network (VPN) known for its simplicity and ease of use. It uses proven cryptography protocols and algorithms to protect data.
- Installing and configuring the wireguard vpn is simple as it can be done using simple commands
- For installing wireguard just type the below command



```
yagnahaun@yagnahaun14:~$ sudo apt install wireguard
```

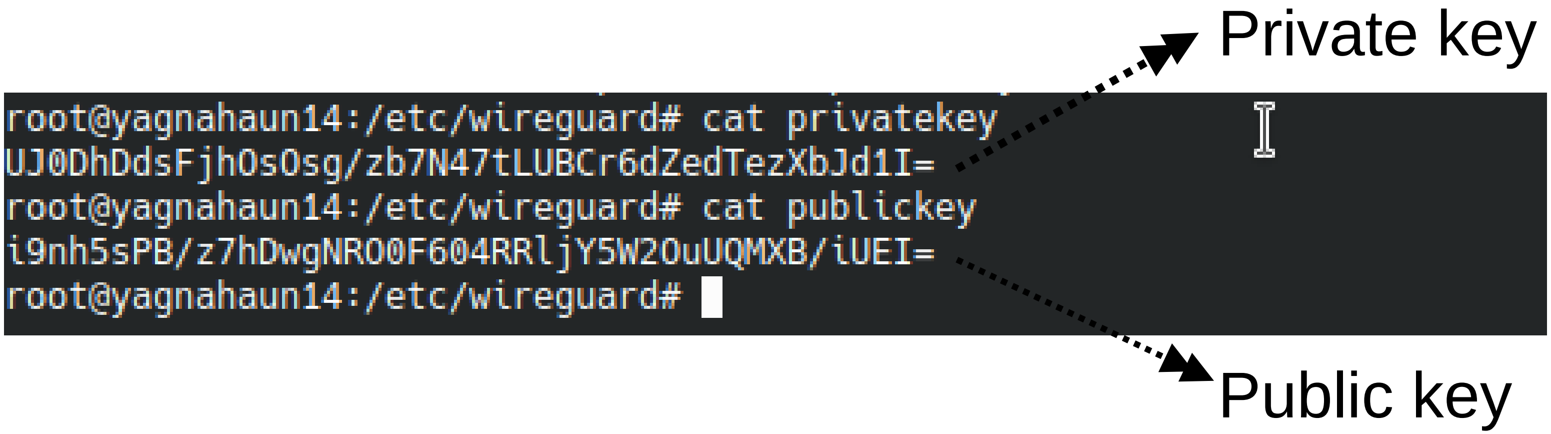
Configuring Wireguard Server

- WireGuard requires base64-encoded public and private keys. These can be generated using the wg(8) utility and the below commands gives us the required public and private keys for setting up the wireguard server:



```
root@yagnahaun14:/etc/wireguard# umask 077; wg genkey | tee privatekey | wg pubkey > pub
lickey
root@yagnahaun14:/etc/wireguard# ls -l privatekey publickey
-rw------- 1 root root 45 Sep  5 15:01 privatekey
-rw------- 1 root root 45 Sep  5 15:01 publickey
```

- From the above you can see that the public and private keys are generated.



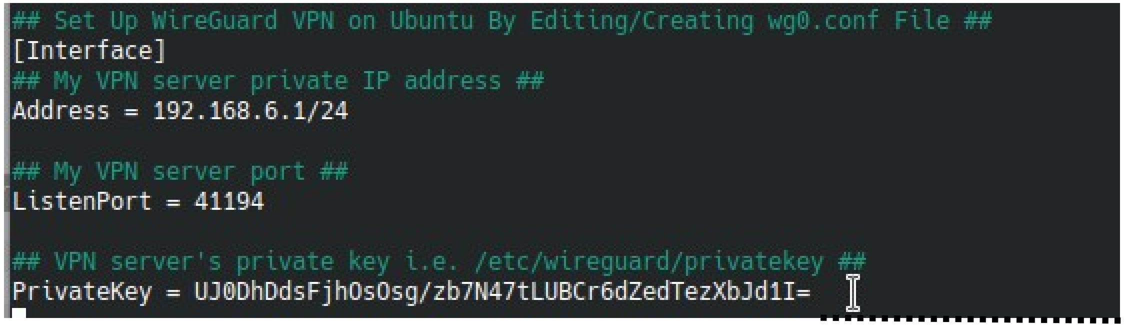
```
root@yagnahaun14:/etc/wireguard# cat privatekey
UJ0DhDdsFjhOsOsg/zb7N47tLUBCr6dZedTezXbJd1I=
root@yagnahaun14:/etc/wireguard# cat publickey
i9nh5sPB/z7hDwgNRO0F604RRljY5W2OuUQMXB/iUEI=
root@yagnahaun14:/etc/wireguard# 
```

Private key

Public key

- Now we will setup the configuration for the wireguard server

- The contents of the config file are:



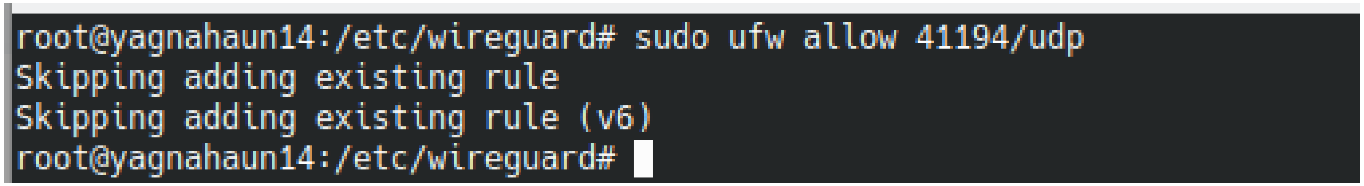
```
## Set Up WireGuard VPN on Ubuntu By Editing/Creating wg0.conf File ##
[Interface]
## My VPN server private IP address ##
Address = 192.168.6.1/24

## My VPN server port ##
ListenPort = 41194

## VPN server's private key i.e. /etc/wireguard/privatekey ##
PrivateKey = UJ0DhDdsFjhOsOsg/zb7N47tLUBCr6dZedTezXbJd1I=
```

Private key that was generated earlier

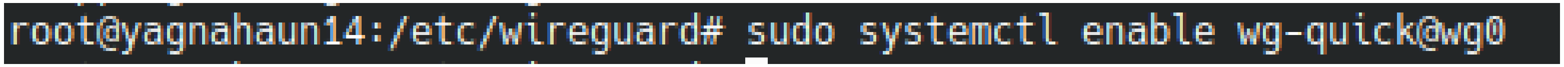
- Now we are going to open UDP 41194 port using the following ufw command



```
root@yagnahaun14:/etc/wireguard# sudo ufw allow 41194/udp
Skipping adding existing rule
Skipping adding existing rule (v6)
root@yagnahaun14:/etc/wireguard#
```

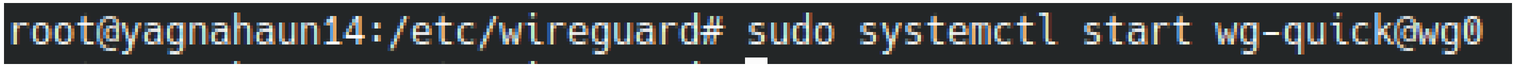
Enable and start WireGuard service

- Turn the WireGuard service at boot time using the systemctl command, run:



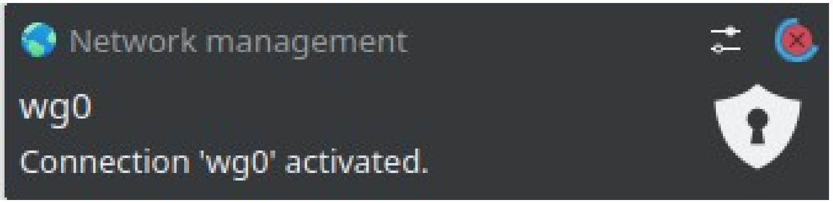
```
root@yagnahaun14:/etc/wireguard# sudo systemctl enable wg-quick@wg0
```

- Start the service, execute:



```
root@yagnahaun14:/etc/wireguard# sudo systemctl start wg-quick@wg0
```

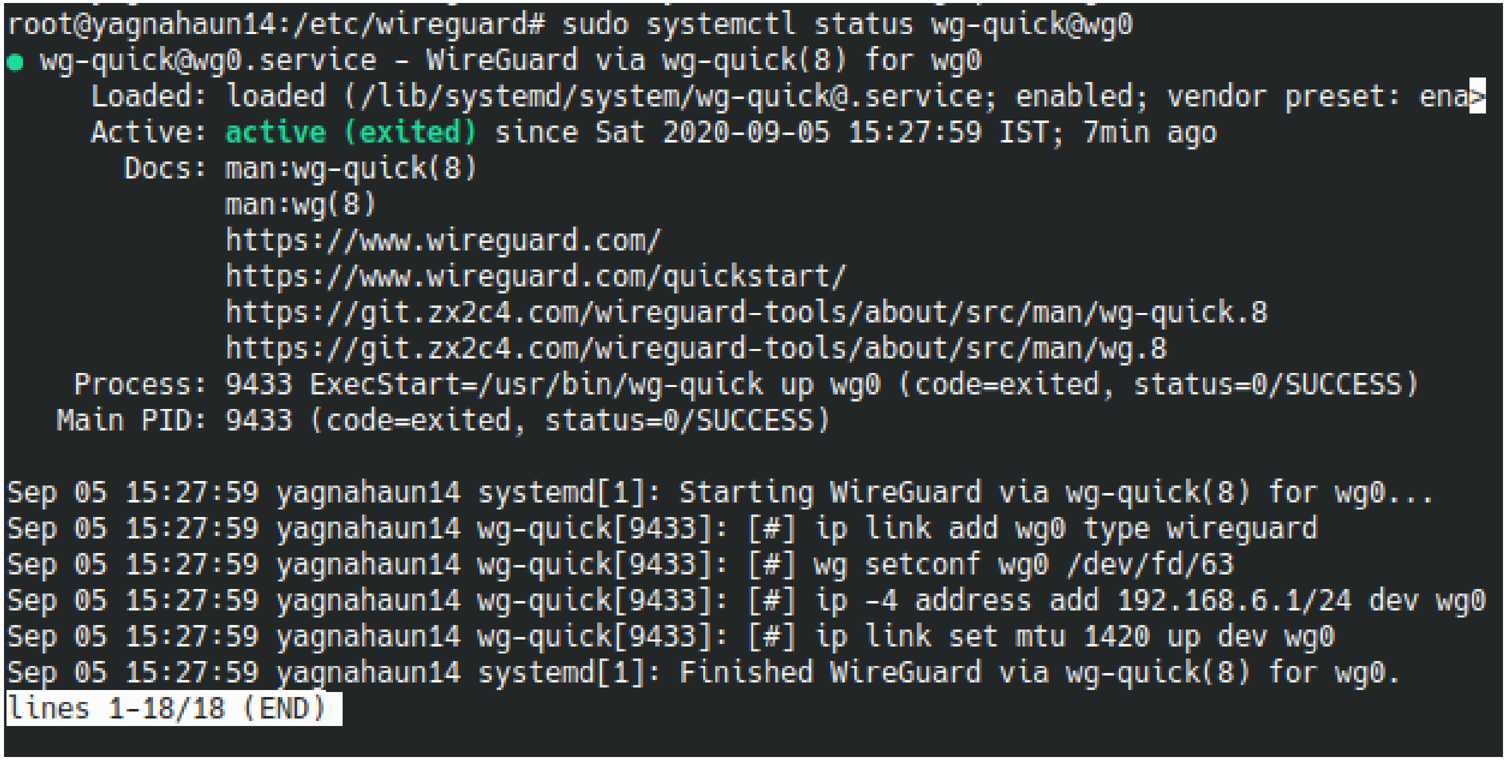
When you run the above command you can see the below prompt:



wg0

Connection 'wg0' activated.

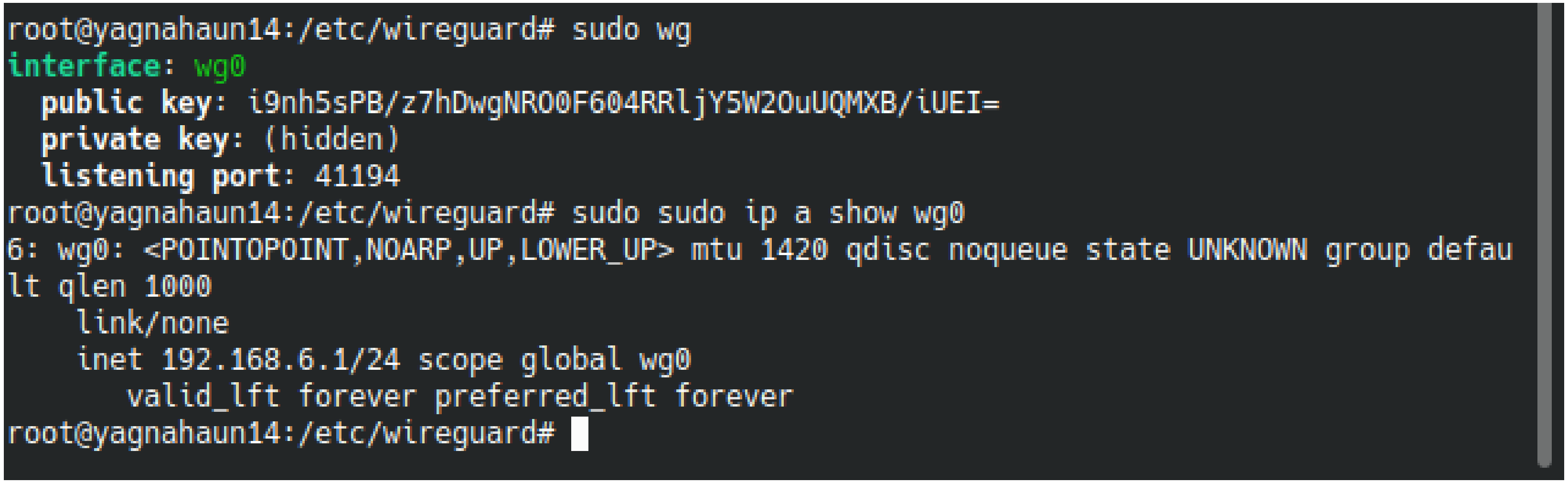
- The below command gives us whether the server is active or down:



```
root@yagnahaun14:/etc/wireguard# sudo systemctl status wg-quick@wg0
● wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0
     Loaded: loaded (/lib/systemd/system/wg-quick@.service; enabled; vendor preset: ena>
     Active: active (exited) since Sat 2020-09-05 15:27:59 IST; 7min ago
       Docs: man:wg-quick(8)
             man:wg(8)
             https://www.wireguard.com/
             https://www.wireguard.com/quickstart/
             https://git.zx2c4.com/wireguard-tools/about/src/man/wg-quick.8
             https://git.zx2c4.com/wireguard-tools/about/src/man/wg.8
    Process: 9433 ExecStart=/usr/bin/wg-quick up wg0 (code=exited, status=0/SUCCESS)
   Main PID: 9433 (code=exited, status=0/SUCCESS)

Sep 05 15:27:59 yagnahaun14 systemd[1]: Starting WireGuard via wg-quick(8) for wg0...
Sep 05 15:27:59 yagnahaun14 wg-quick[9433]: [#] ip link add wg0 type wireguard
Sep 05 15:27:59 yagnahaun14 wg-quick[9433]: [#] wg setconf wg0 /dev/fd/63
Sep 05 15:27:59 yagnahaun14 wg-quick[9433]: [#] ip -4 address add 192.168.6.1/24 dev wg0
Sep 05 15:27:59 yagnahaun14 wg-quick[9433]: [#] ip link set mtu 1420 up dev wg0
Sep 05 15:27:59 yagnahaun14 systemd[1]: Finished WireGuard via wg-quick(8) for wg0.
lines 1-18/18 (END)
```

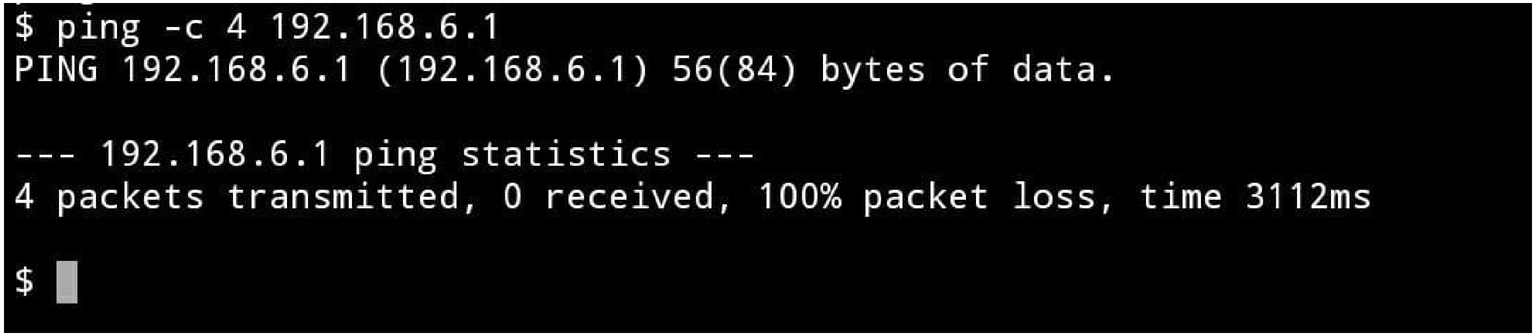
- Verify that interface named wg0 is up and running on Ubuntu server using the below commands:



```
root@yagnahaun14:/etc/wireguard# sudo wg
interface: wg0
  public key: i9nh5sPB/z7hDwgNRO0F604RRljY5W2OuUQMXB/iUEI=
  private key: (hidden)
  listening port: 41194
root@yagnahaun14:/etc/wireguard# sudo sudo ip a show wg0
6: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/none
    inet 192.168.6.1/24 scope global wg0
       valid_lft forever preferred_lft forever
root@yagnahaun14:/etc/wireguard#
```

- We are done with setting up the wireguard vpn server.

- Now we will try to ping the server and check using the ping command:



```
$ ping -c 4 192.168.6.1
PING 192.168.6.1 (192.168.6.1) 56(84) bytes of data.

--- 192.168.6.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3112ms

$ 
```

- As we can see the server is up and packet transmission was done but there is a packet loss which we are looking into