Goal: Create a secure VPN between two Linux virtual machines named **Sri Lanka** (client) and **US** (server) hosted on AWS EC2. Use an open-source VPN (WireGuard) and set up a Samba server on **US** that is accessible only over the VPN from **Sri Lanka**. Provide commands, configuration files, and a network diagram.

In this assignment I'm used to setup two virtual machines as AWS cloud EC2 instances. Because in my computer doesn't have any power to run 2 Virtual machines same time.

Overview / design

- Use two AWS EC2 Ubuntu instances (22.04). One acts as **US (server)** and the other as **Sri Lanka (client)**.
- Use **WireGuard** as the VPN (open-source, lightweight, easier to configure for lab). VPN internal subnet: 10.10.0.0/24.
    - **US (server)** WireGuard VPN IP →10.10.0.1/24
    - **Sri Lanka (client)** WireGuard VPN IP → 10.10.0.2/24
- Samba server runs on **US** and binds to the WireGuard interface so the share is accessible only through VPN.
- Use **wg-quick** to manage the interface. Enable IP forwarding on server if required.

Security considerations (lab):

- Allow SSH (22) from your admin IP only.
- Allow WireGuard UDP (51820) to server from your client public IP.
- Use ICMP for testing (ping).

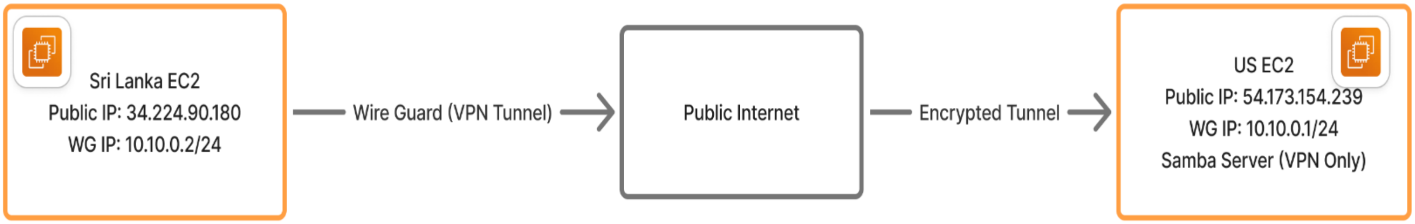
Two EC2 instances setup through AWS console.

1. Login to AWS console.
2. Select a region (In this scenario, I'm used same region to both instances). Launch two EC2 instances (Ubuntu Server 22.04 LTS):
    - Name: **US-vpn-server**
    - Name: **SriLanka-vpn-client**
3. Create or choose a key pair (.pem) for SSH access.
4. Create security groups as both instances;

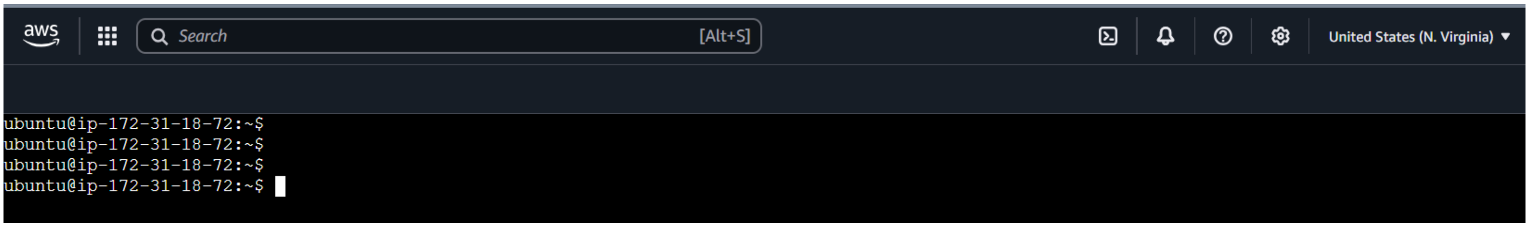
    Inbound rules:

    - SSH (TCP 22) — Source: 0.0.0.0/0
    - WireGuard (UDP 51820) — Source: 0.0.0.0/0
    - ICMP (ping) — Source: <your-ip-address> /32 or allowed as needed.

Network diagram of this VPN,

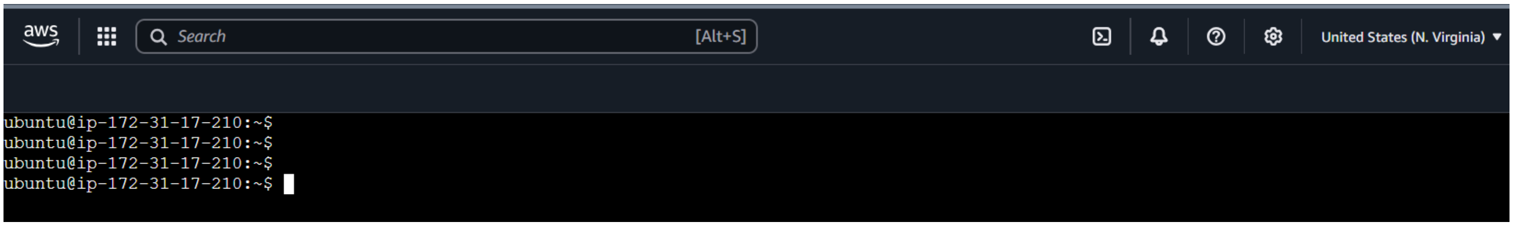


SriLanka-vpn-server (172.31.18.72)



```
ubuntu@ip-172-31-18-72:~$
ubuntu@ip-172-31-18-72:~$
ubuntu@ip-172-31-18-72:~$
ubuntu@ip-172-31-18-72:~$
```

US-vpn-client (172.31.17.210)



```
ubuntu@ip-172-31-17-210:~$
ubuntu@ip-172-31-17-210:~$
ubuntu@ip-172-31-17-210:~$
ubuntu@ip-172-31-17-210:~$
```

Prepare both VMs (Ubuntu commands)

Run these commands on **both** machines (replace ubuntu with your username if different):

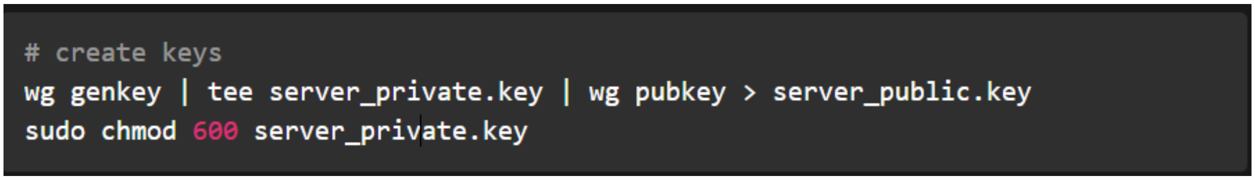
```
# update & basic tools

sudo apt update && sudo apt upgrade -y

sudo apt install -y wireguard qrencode samba smbclient
cifs-utils ufw curl


# enable ufw and allow ssh

sudo ufw allow from <Your_Public_IP> to any port 22 proto
tcp

sudo ufw --force enable
```

Assign "Your_public_IP" to each instance public IP address.

WireGuard setup

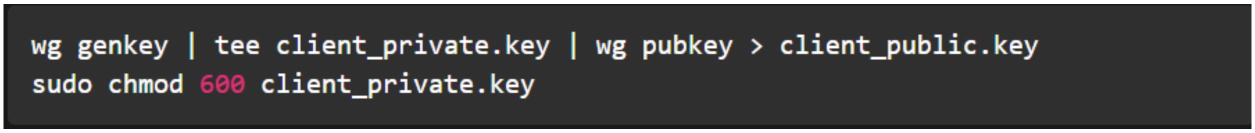
1) Generate keypairs

On US (server):



```
# create keys
wg genkey | tee server_private.key | wg pubkey > server_public.key
sudo chmod 600 server_private.key
```

On SL (Client) :

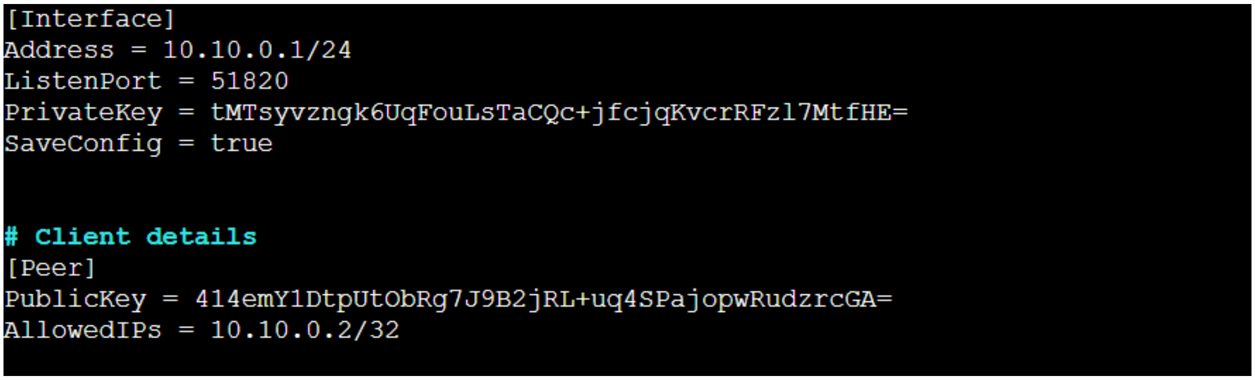


```
wg genkey | tee client_private.key | wg pubkey > client_public.key
sudo chmod 600 client_private.key
```

You can see both public keys as “cat client_public.key” and “cat server_public.key” commands using in terminal.

2) Server config /etc/wireguard/wg0.conf (US server)

Create the file with “ sudo vi /etc/wireguard/wg0.conf” command and modify that file as follows,

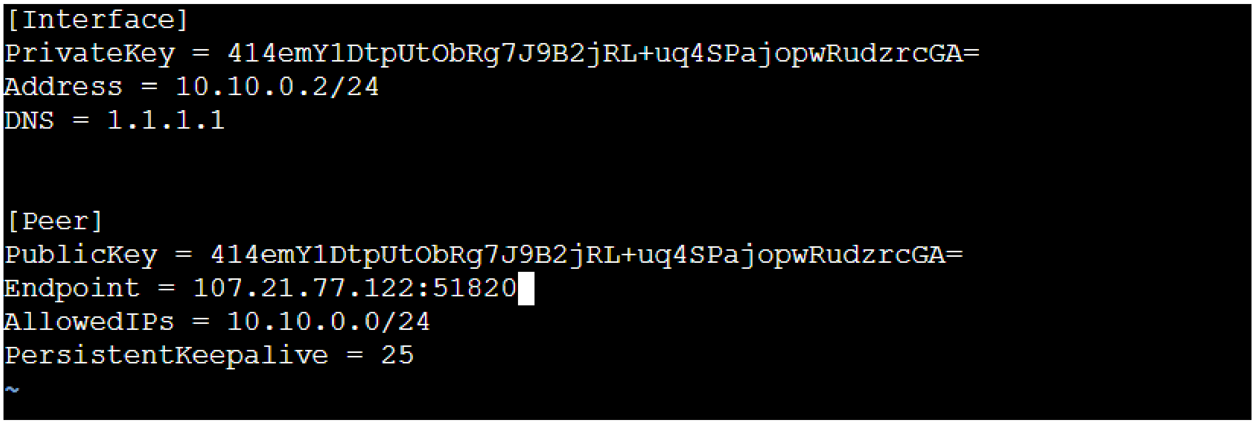


```
[Interface]
Address = 10.10.0.1/24
ListenPort = 51820
PrivateKey = tMTsyvzngk6UqFouLsTaCQc+jfcjqKvcrRFzl7MtfHE=
SaveConfig = true


# Client details
[Peer]
PublicKey = 414emY1DtpUtObRg7J9B2jRL+uq4SPajopwRudzrcGA=
AllowedIPs = 10.10.0.2/32
```

In this **PrivateKey** section added server’s private key and **PublicKey** section added client’s public key.

3) Client config /etc/wireguard/wg0.conf (on SL client)

Create the file with “sudo vi /etc/wireguard/wg0.conf” command and notify that file as follows,

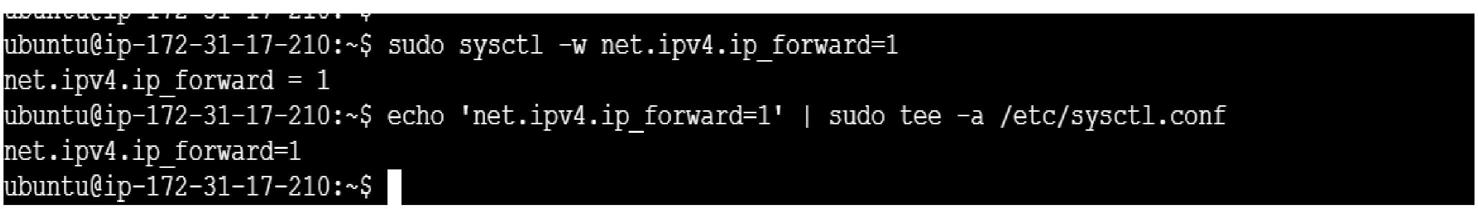


```
[Interface]
PrivateKey = 414emY1DtpUtObRg7J9B2jRL+uq4SPajopwRudzrcGA=
Address = 10.10.0.2/24
DNS = 1.1.1.1


[Peer]
PublicKey = 414emY1DtpUtObRg7J9B2jRL+uq4SPajopwRudzrcGA=
Endpoint = 107.21.77.122:51820
AllowedIPs = 10.10.0.0/24
PersistentKeepalive = 25
~
```

In this **PrivateKey** section added client’s private key and **PublicKey** section added server’s public key as well as **Endpoint** added server’s public Ip address with 51820 port number.

4) Enable forwarding on server,

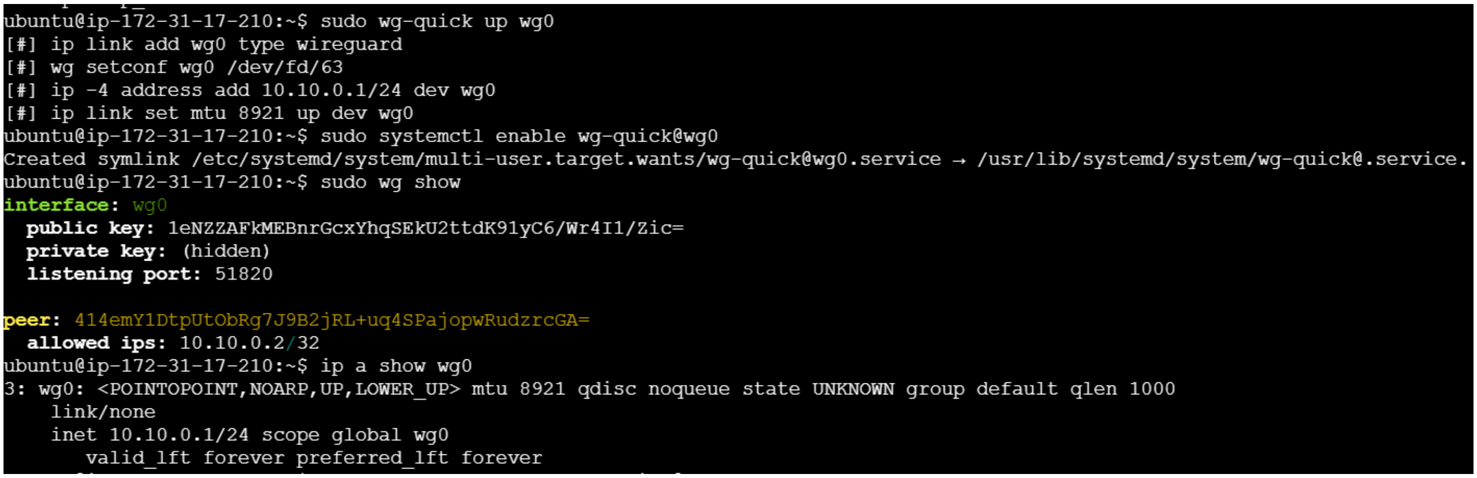
On US (server):



```
ubuntu@ip-172-31-17-210:~$
ubuntu@ip-172-31-17-210:~$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
ubuntu@ip-172-31-17-210:~$ echo 'net.ipv4.ip_forward=1' | sudo tee -a /etc/sysctl.conf
net.ipv4.ip_forward=1
ubuntu@ip-172-31-17-210:~$
```

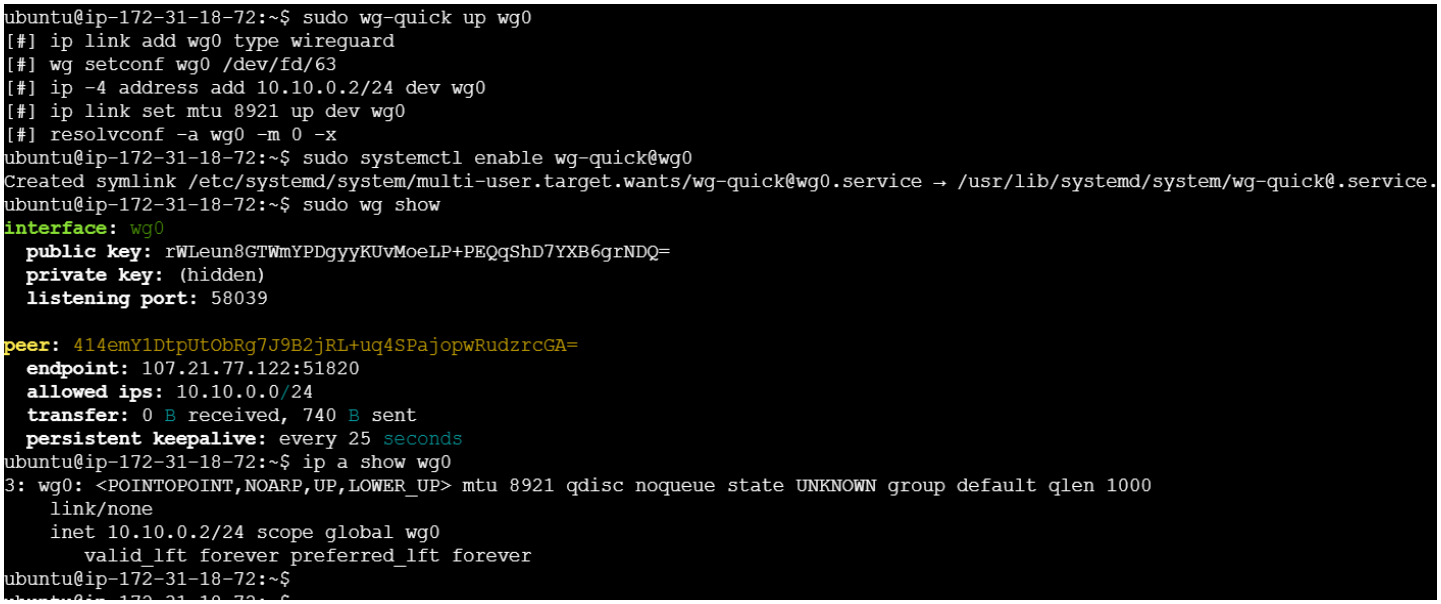
5) Start WireGuard on both machines,

Us server,



```
ubuntu@ip-172-31-17-210:~$ sudo wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.10.0.1/24 dev wg0
[#] ip link set mtu 8921 up dev wg0
ubuntu@ip-172-31-17-210:~$ sudo systemctl enable wg-quick@wg0
Created symlink /etc/systemd/system/multi-user.target.wants/wg-quick@wg0.service → /usr/lib/systemd/system/wg-quick@.service.
ubuntu@ip-172-31-17-210:~$ sudo wg show
interface: wg0
  public key: 1eNZZAFkMEBnrGcxYhqSEkU2ttdK91yC6/Wr4I1/Zic=
  private key: (hidden)
  listening port: 51820

peer: 414emY1DtpUtObRg7J9B2jRL+uq4SPajopwRudzrcGA=
  allowed ips: 10.10.0.2/32
ubuntu@ip-172-31-17-210:~$ ip a show wg0
3: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 8921 qdisc noqueue state UNKNOWN group default qlen 1000
    link/none
    inet 10.10.0.1/24 scope global wg0
       valid_lft forever preferred_lft forever
```

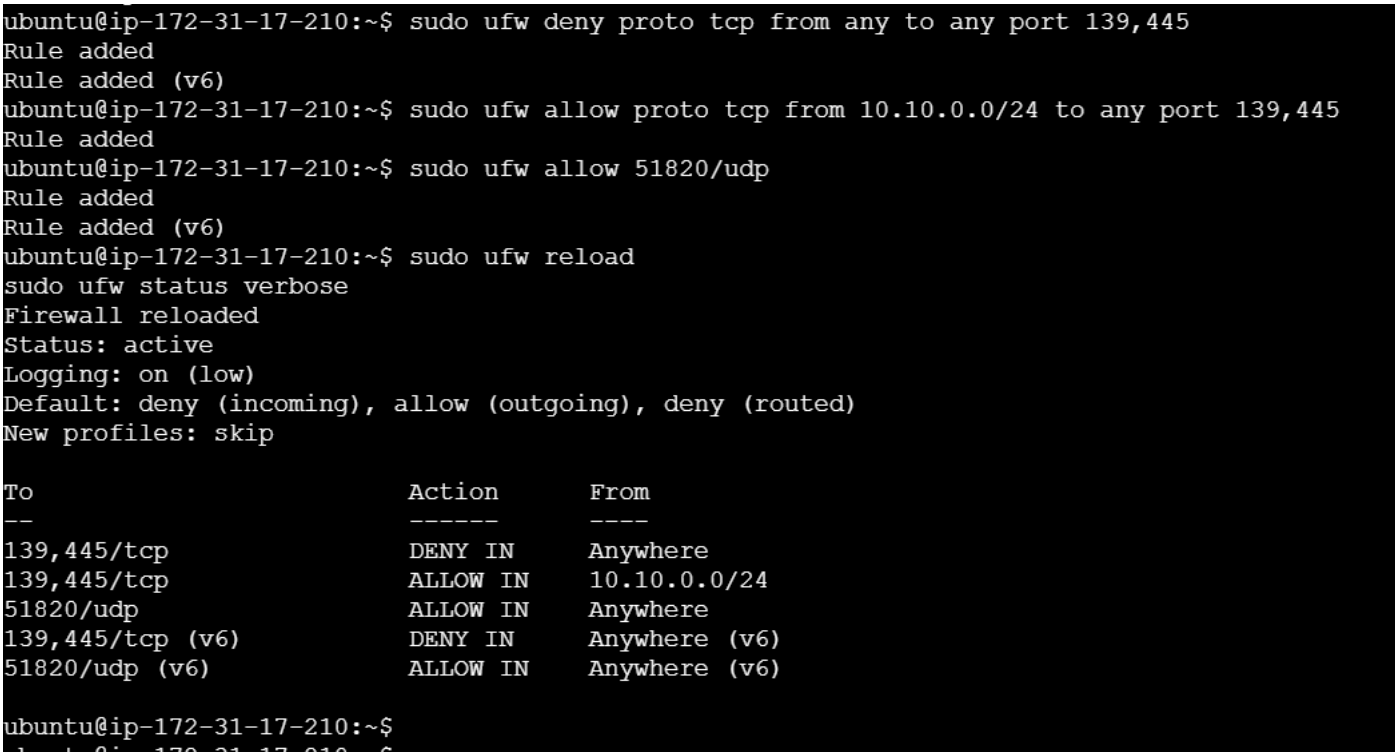
SL Client,



```
ubuntu@ip-172-31-18-72:~$ sudo wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.10.0.2/24 dev wg0
[#] ip link set mtu 8921 up dev wg0
[#] resolvconf -a wg0 -m 0 -x
ubuntu@ip-172-31-18-72:~$ sudo systemctl enable wg-quick@wg0
Created symlink /etc/systemd/system/multi-user.target.wants/wg-quick@wg0.service → /usr/lib/systemd/system/wg-quick@.service.
ubuntu@ip-172-31-18-72:~$ sudo wg show
interface: wg0
  public key: rWLeun8GTWmYPDgyyKUvMoeLP+PEQqShD7YXB6grNDQ=
  private key: (hidden)
  listening port: 58039

peer: 414emY1DtpUtObRg7J9B2jRL+uq4SPajopwRudzrcGA=
  endpoint: 107.21.77.122:51820
  allowed ips: 10.10.0.0/24
  transfer: 0 B received, 740 B sent
  persistent keepalive: every 25 seconds
ubuntu@ip-172-31-18-72:~$ ip a show wg0
3: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 8921 qdisc noqueue state UNKNOWN group default qlen 1000
    link/none
    inet 10.10.0.2/24 scope global wg0
       valid_lft forever preferred_lft forever
ubuntu@ip-172-31-18-72:~$
```

Firewall rules (UFW) to restrict Samba to VPN only

On US (server):

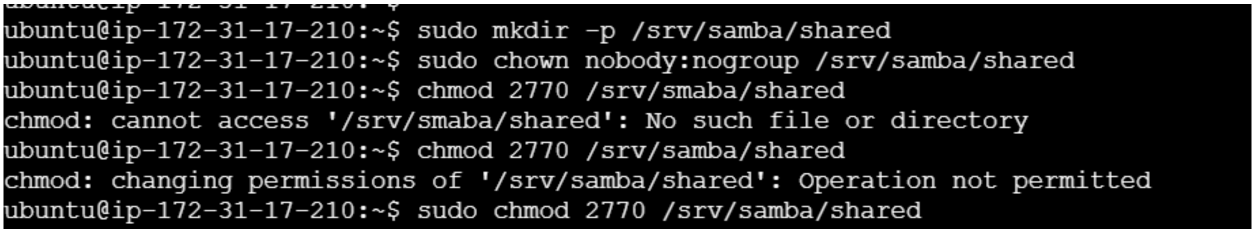


```
ubuntu@ip-172-31-17-210:~$ sudo ufw deny proto tcp from any to any port 139,445
Rule added
Rule added (v6)
ubuntu@ip-172-31-17-210:~$ sudo ufw allow proto tcp from 10.10.0.0/24 to any port 139,445
Rule added
ubuntu@ip-172-31-17-210:~$ sudo ufw allow 51820/udp
Rule added
Rule added (v6)
ubuntu@ip-172-31-17-210:~$ sudo ufw reload
sudo ufw status verbose
Firewall reloaded
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
139,445/tcp                DENY IN     Anywhere
139,445/tcp                ALLOW IN    10.10.0.0/24
51820/udp                  ALLOW IN    Anywhere
139,445/tcp (v6)           DENY IN     Anywhere (v6)
51820/udp (v6)             ALLOW IN    Anywhere (v6)

ubuntu@ip-172-31-17-210:~$
ubuntu@ip-172-31-17-210:~$
```

This ensures Samba traffic is only allowed from the WireGuard virtual network.

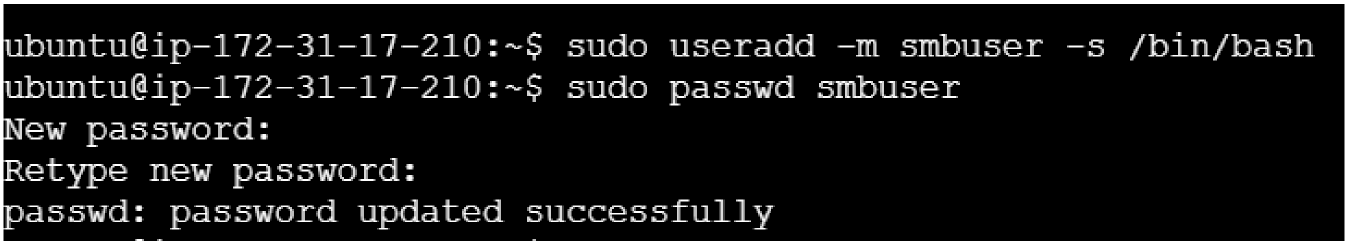
6) Samba setup (on US)

I. Create a directory (srv/smba/shared) and set permissions:

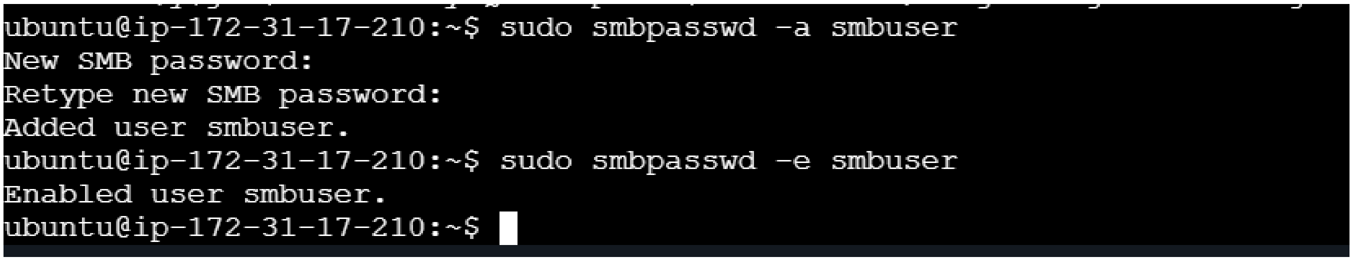


```
ubuntu@ip-172-31-17-210:~$
ubuntu@ip-172-31-17-210:~$ sudo mkdir -p /srv/samba/shared
ubuntu@ip-172-31-17-210:~$ sudo chown nobody:nogroup /srv/samba/shared
ubuntu@ip-172-31-17-210:~$ chmod 2770 /srv/smaba/shared
chmod: cannot access '/srv/smaba/shared': No such file or directory
ubuntu@ip-172-31-17-210:~$ chmod 2770 /srv/samba/shared
chmod: changing permissions of '/srv/samba/shared': Operation not permitted
ubuntu@ip-172-31-17-210:~$ sudo chmod 2770 /srv/samba/shared
```

II. Create a Samba user (map to existing Linux user):3.

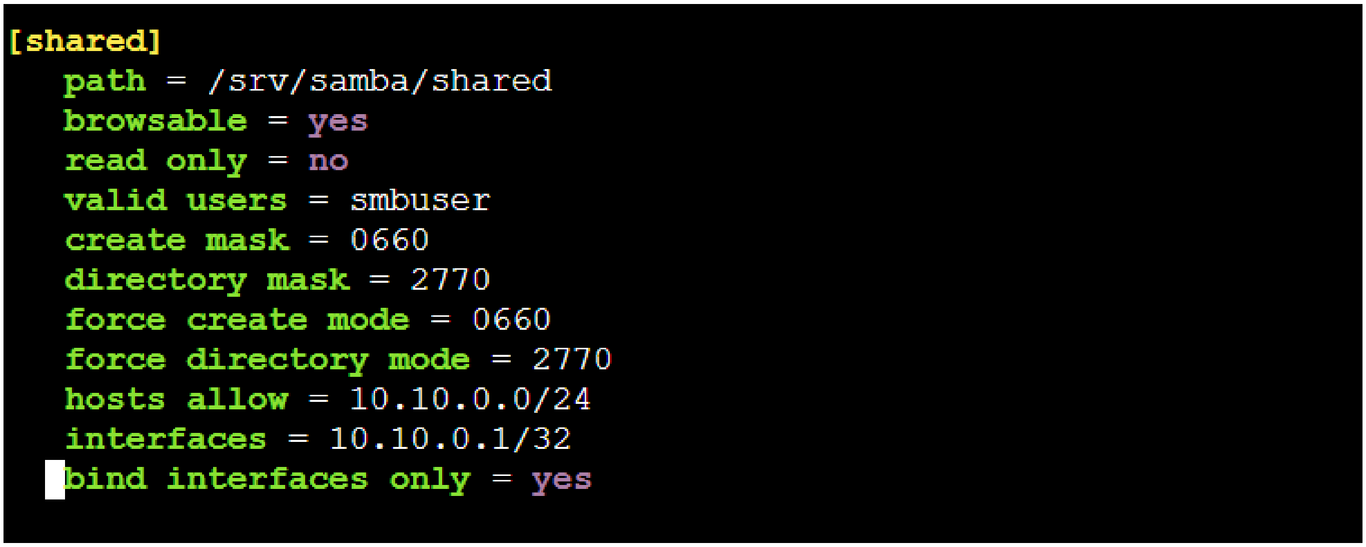


```
ubuntu@ip-172-31-17-210:~$ sudo useradd -m smbuser -s /bin/bash
ubuntu@ip-172-31-17-210:~$ sudo passwd smbuser
New password:
Retype new password:
passwd: password updated successfully
```



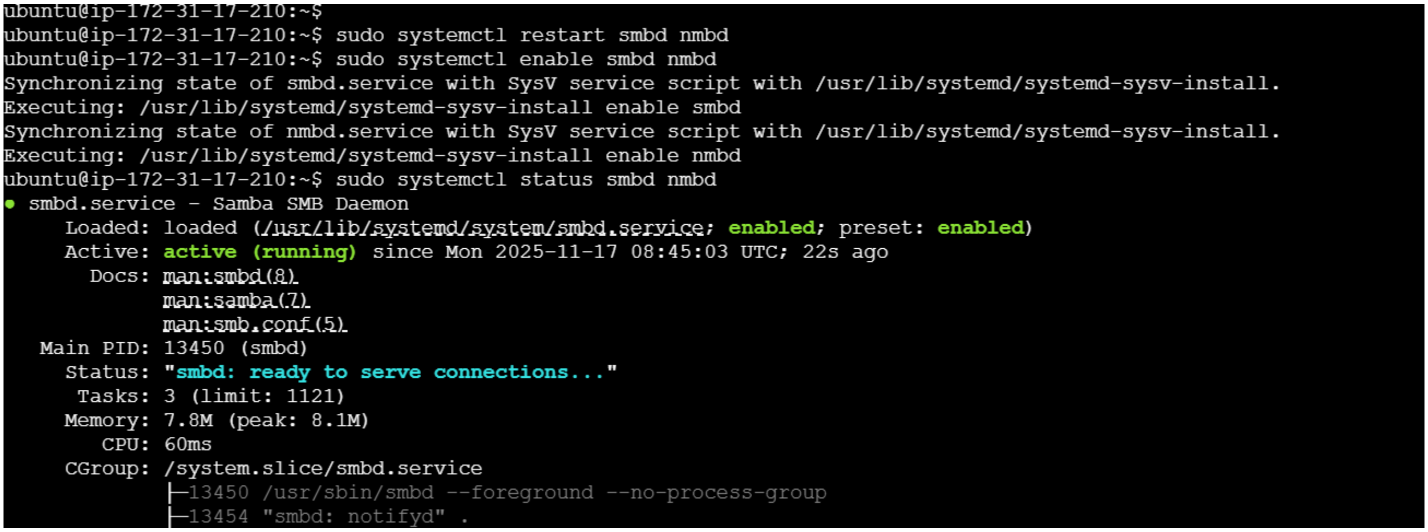
```
ubuntu@ip-172-31-17-210:~$ sudo smbpasswd -a smbuser
New SMB password:
Retype new SMB password:
Added user smbuser.
ubuntu@ip-172-31-17-210:~$ sudo smbpasswd -e smbuser
Enabled user smbuser.
ubuntu@ip-172-31-17-210:~$
```

III. Added below [shared] section at the end of **/etc/samba/smb.conf file**,



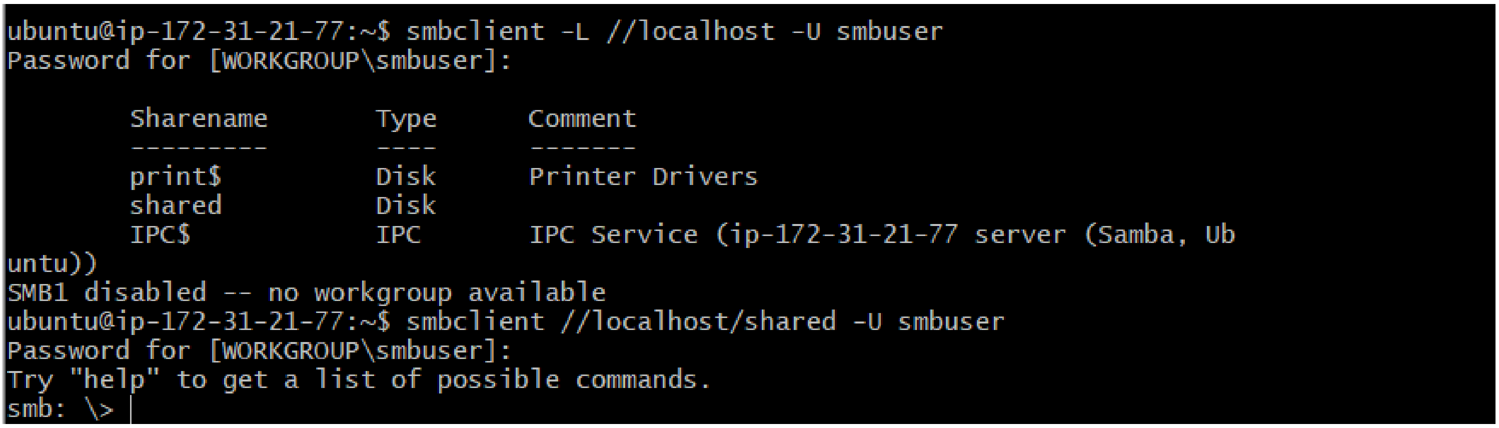
```
[shared]
   path = /srv/samba/shared
   browsable = yes
   read only = no
   valid users = smbuser
   create mask = 0660
   directory mask = 2770
   force create mode = 0660
   force directory mode = 2770
   hosts allow = 10.10.0.0/24
   interfaces = 10.10.0.1/32
   bind interfaces only = yes
```

IV. Restart and enable samba service



```
ubuntu@ip-172-31-17-210:~$
ubuntu@ip-172-31-17-210:~$ sudo systemctl restart smbd nmbd
ubuntu@ip-172-31-17-210:~$ sudo systemctl enable smbd nmbd
Synchronizing state of smbd.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable smbd
Synchronizing state of nmbd.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nmbd
ubuntu@ip-172-31-17-210:~$ sudo systemctl status smbd nmbd
● smbd.service - Samba SMB Daemon
     Loaded: loaded (/usr/lib/systemd/system/smbd.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-11-17 08:45:03 UTC; 22s ago
       Docs: man:smbd(8)
             man:samba(7)
             man:smb.conf(5)
   Main PID: 13450 (smbd)
     Status: "smbd: ready to serve connections..."
      Tasks: 3 (limit: 1121)
     Memory: 7.8M (peak: 8.1M)
        CPU: 60ms
     CGroup: /system.slice/smbd.service
             ├─13450 /usr/sbin/smbd --foreground --no-process-group
             ├─13454 "smbd: notifyd" .
```

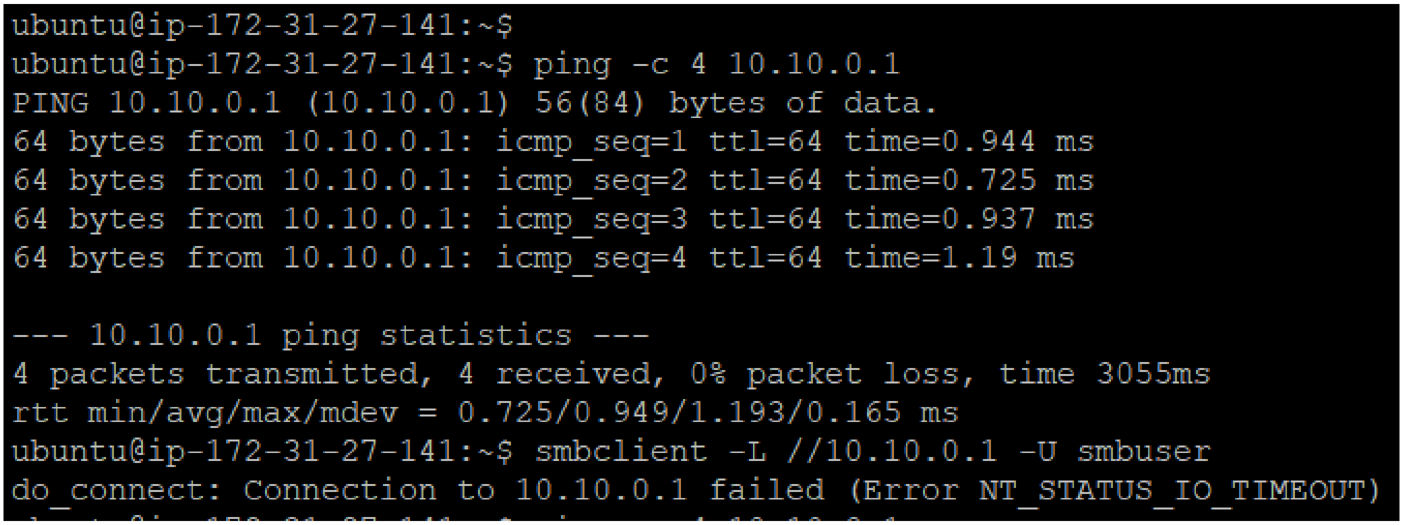
V. Test locally on server



```
ubuntu@ip-172-31-21-77:~$ smbclient -L //localhost -U smbuser
Password for [WORKGROUP\smbuser]:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        shared          Disk
        IPC$            IPC       IPC Service (ip-172-31-21-77 server (Samba, Ub
untu))
SMB1 disabled -- no workgroup available
ubuntu@ip-172-31-21-77:~$ smbclient //localhost/shared -U smbuser
Password for [WORKGROUP\smbuser]:
Try "help" to get a list of possible commands.
smb: \>
```

Access Samba from Sri_Lanka (client)

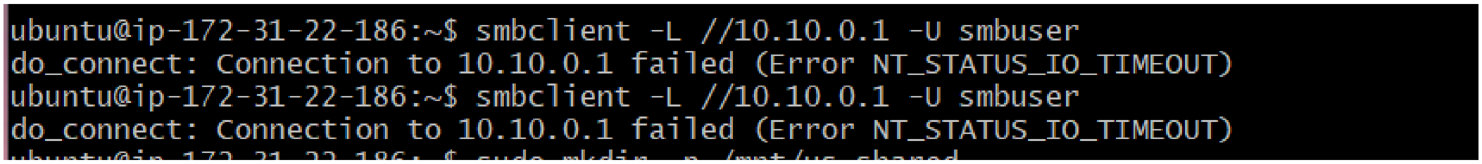
1. Ensure WireGuard is up on client and you can ping the server's VPN IP:



```
ubuntu@ip-172-31-27-141:~$
ubuntu@ip-172-31-27-141:~$ ping -c 4 10.10.0.1
PING 10.10.0.1 (10.10.0.1) 56(84) bytes of data.
64 bytes from 10.10.0.1: icmp_seq=1 ttl=64 time=0.944 ms
64 bytes from 10.10.0.1: icmp_seq=2 ttl=64 time=0.725 ms
64 bytes from 10.10.0.1: icmp_seq=3 ttl=64 time=0.937 ms
64 bytes from 10.10.0.1: icmp_seq=4 ttl=64 time=1.19 ms

--- 10.10.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3055ms
rtt min/avg/max/mdev = 0.725/0.949/1.193/0.165 ms
ubuntu@ip-172-31-27-141:~$ smbclient -L //10.10.0.1 -U smbuser
do_connect: Connection to 10.10.0.1 failed (Error NT_STATUS_IO_TIMEOUT)
```

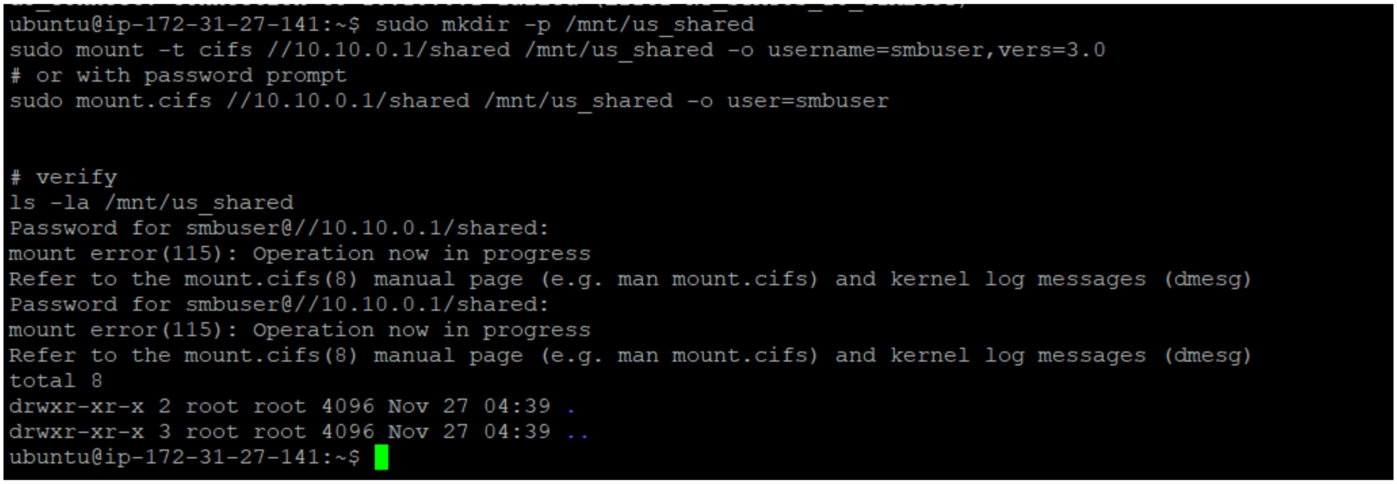
Here you can see ping command output and it is successful. That’s means wiregurad is up on client and can ping to server’s IP address.

2. List shares (from client):



```
ubuntu@ip-172-31-22-186:~$ smbclient -L //10.10.0.1 -U smbuser
do_connect: Connection to 10.10.0.1 failed (Error NT_STATUS_IO_TIMEOUT)
ubuntu@ip-172-31-22-186:~$ smbclient -L //10.10.0.1 -U smbuser
do_connect: Connection to 10.10.0.1 failed (Error NT_STATUS_IO_TIMEOUT)
```

3. Mount the share (create a mount point first)



```
ubuntu@ip-172-31-27-141:~$ sudo mkdir -p /mnt/us_shared
sudo mount -t cifs //10.10.0.1/shared /mnt/us_shared -o username=smbuser,vers=3.0
# or with password prompt
sudo mount.cifs //10.10.0.1/shared /mnt/us_shared -o user=smbuser


# verify
ls -la /mnt/us_shared
Password for smbuser@//10.10.0.1/shared:
mount error(115): Operation now in progress
Refer to the mount.cifs(8) manual page (e.g. man mount.cifs) and kernel log messages (dmesg)
Password for smbuser@//10.10.0.1/shared:
mount error(115): Operation now in progress
Refer to the mount.cifs(8) manual page (e.g. man mount.cifs) and kernel log messages (dmesg)
total 8
drwxr-xr-x 2 root root 4096 Nov 27 04:39 .
drwxr-xr-x 3 root root 4096 Nov 27 04:39 ..
ubuntu@ip-172-31-27-141:~$
```

Finally, Sri Lanka client can be accessing US server correctly, and also the samba setup is working properly. We can use below testing methods to verify that connectivity is success or not.

- ✓ Check wireguard status (run both machines "**sudo wg show**")
- ✓ Ping test (On SL client, "**ping -c 4 10.10.0.1**")
- ✓ Samba Connectivity (List samba users, "**smbclient -L //10.10.0.1 -U smbuser**")