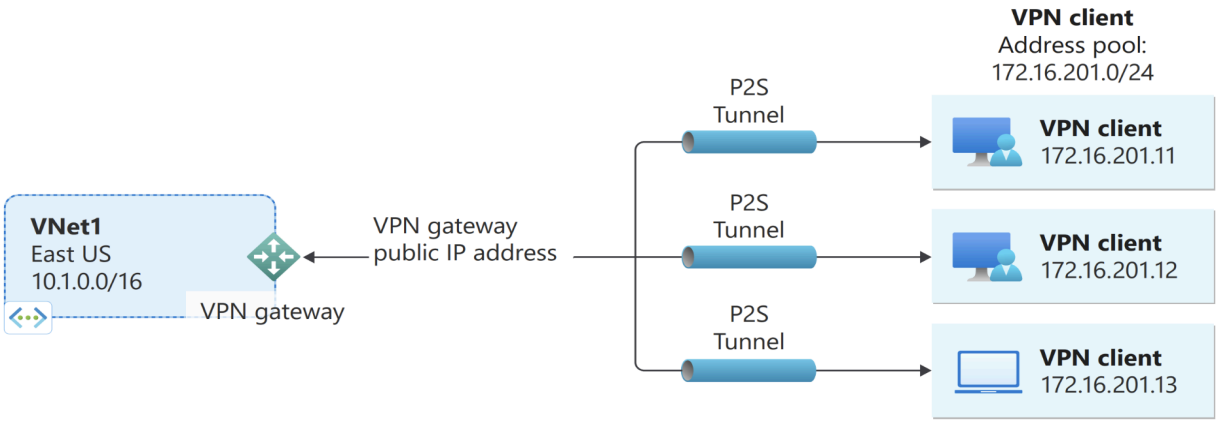
WireGuard VPN Implementation Report

Author:Fouenang Miguel Bruce
Addres:miguelfouenanf@gmail.com
Date: October 20, 2025

1. Introduction



A Virtual Private Network (VPN) provides a secure communication channel between two or more systems through *encryption* and *tunneling*. WireGuard is a modern, lightweight, and high-performance VPN protocol that uses advanced cryptography and is easy to configure compared to traditional VPNs such as OpenVPN and IPSec.

WireGuard's simplicity and speed make it ideal for secure point-to-point connections, as demonstrated in this project.

2. Objectives

The objective of this project is to implement a secure **point-to-point VPN tunnel** between two virtual machines: an *Alpine Linux server* and a *Linux Mint client*. The tunnel allows encrypted data communication within a private subnet (e.g., 10.0.0.0/24).

This setup ensures privacy and security for data transmission in a controlled virtual environment.

3. Methodology

The setup was implemented using two virtual machines running on a hypervisor (such as VirtualBox or Virt-Manager). The Alpine VM was configured as the VPN server, and the Linux Mint VM as the client. The WireGuard interface (wg0) was configured on both sides to create an encrypted link across the virtual network.
Key steps included package installation, key pair generation, configuration file setup, and interface activation.

4. Implementation

Server Configuration (Alpine Linux)

1. Install WireGuard:
`sudo apk add wireguard-tools`

2. Generate key pairs:
`wg genkey | tee privatekey | wg pubkey > publickey`

3. Configure /etc/wireguard/wg0.conf:
```
[Interface]
Address = 10.0.0.1/24
PrivateKey = <SERVER_PRIVATE_KEY>
ListenPort = 51820

[Peer]
PublicKey = <CLIENT_PUBLIC_KEY>
AllowedIPs = 10.0.0.2/32
```

4. Bring up the interface:
`sudo wg-quick up wg0`

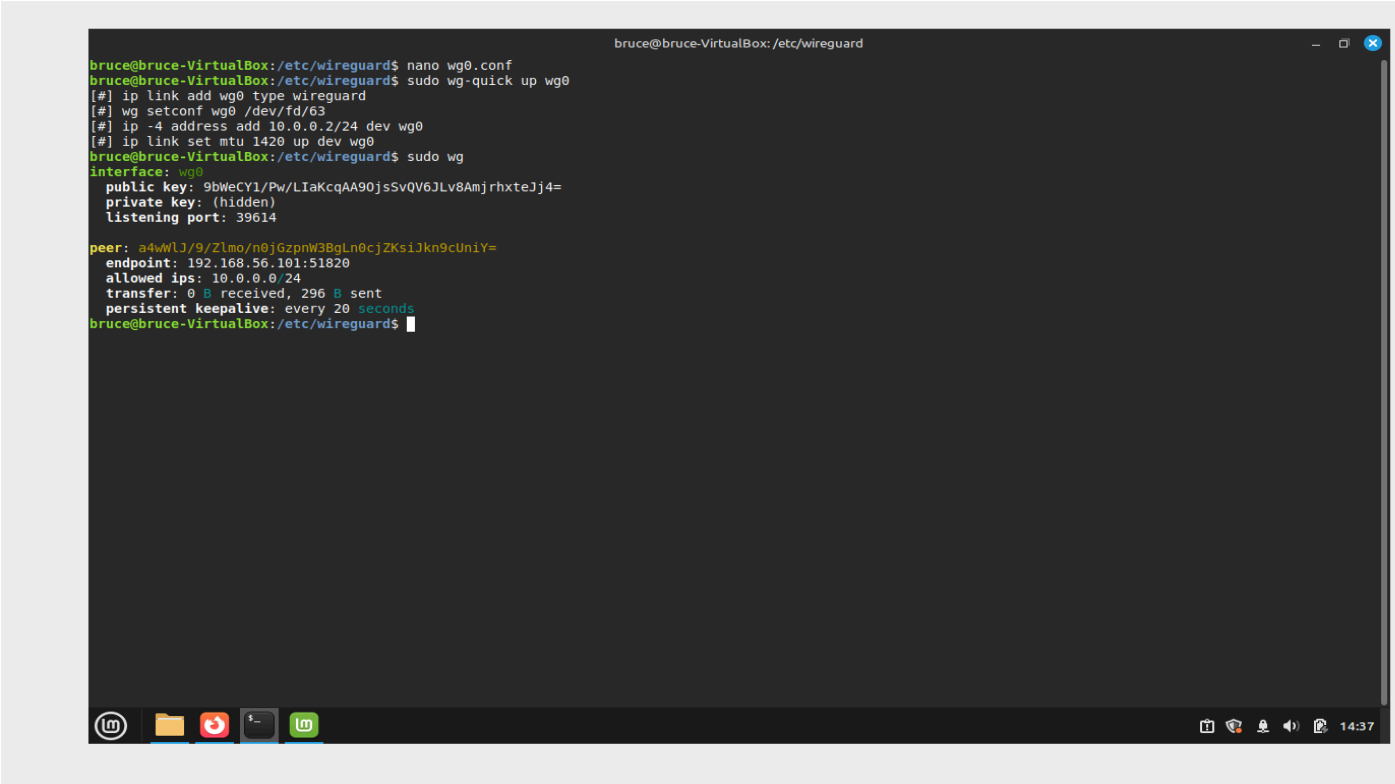
Client Configuration (Linux Mint)

1. Install WireGuard:
`sudo apt install wireguard`

2. Generate key pairs:
`wg genkey | tee privatekey | wg pubkey > publickey`

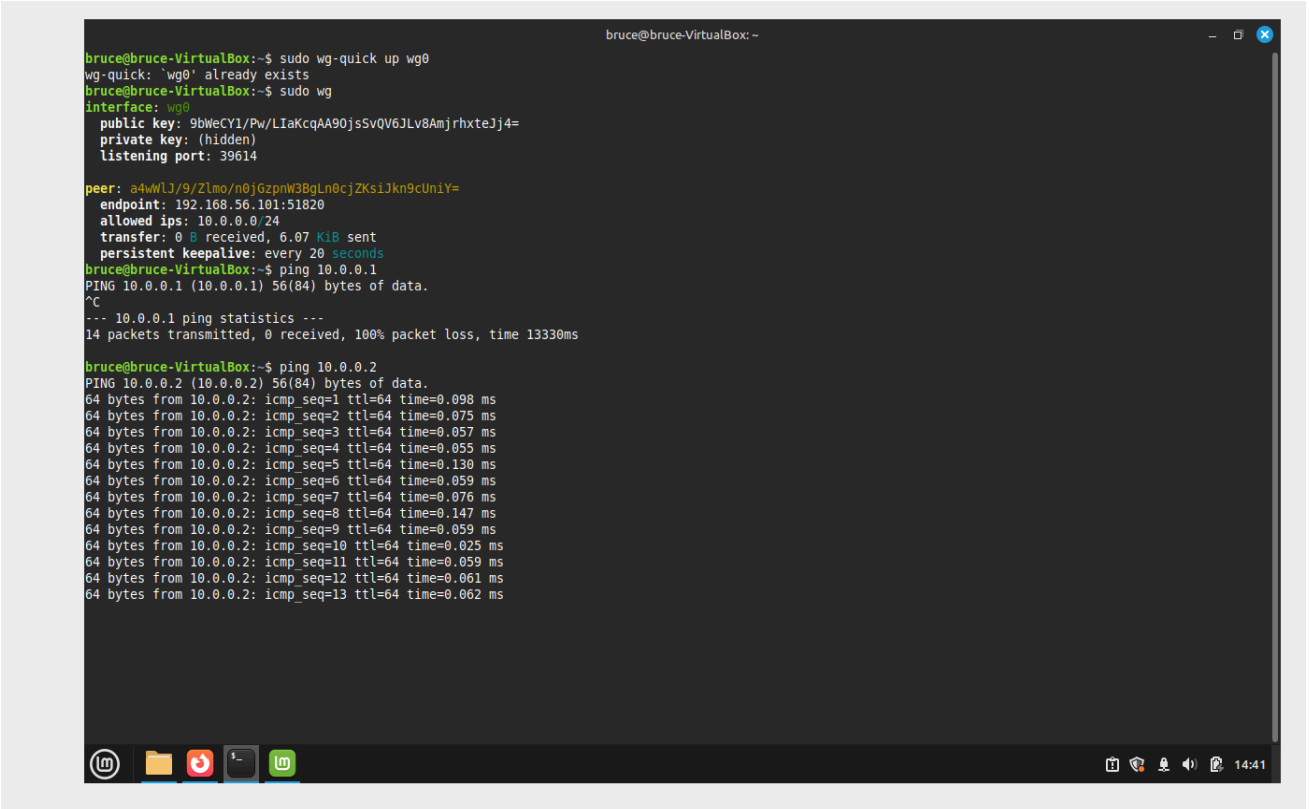
3. Configure /etc/wireguard/wg0.conf:
```
[Interface]
Address = 10.0.0.2/24
PrivateKey = <CLIENT_PRIVATE_KEY>
[Peer]
PublicKey = <SERVER_PUBLIC_KEY>
Endpoint = 192.168.56.101:51820
AllowedIPs = 10.0.0.0/24
```

PersistentKeepalive = 20
```

**4. Bring up the interface:**
`sudo wg-quick up wg0`

**Vérification**

-Check tunnel status:
`sudo wg`


```



- Test connectivity:
From client: `**ping 10.0.0.1**`
From server: `**ping 10.0.0.2**`

The tunnel was successfully established, with encrypted communication verified through basic ping tests.



---

5. Conclusion

This project demonstrated a successful implementation of a secure VPN tunnel using WireGuard between two Linux-based virtual machines. The setup provided a reliable, encrypted communication channel and served as a practical example of network security in action.

Note:

To enhance this project further i will add :
- Add support for multiple peers.
- Implement firewall rules for added security.
- Conduct performance tests (e.g., throughput and latency).