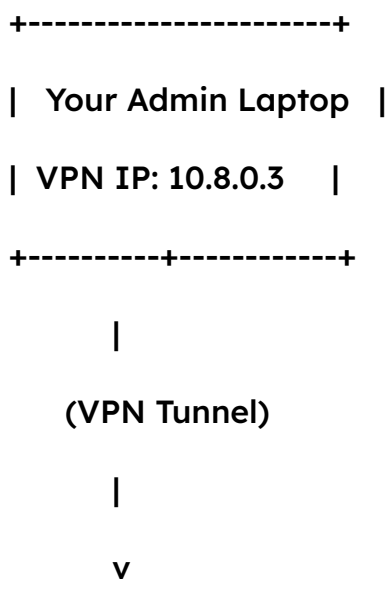
# VPN-secured RDP Access via Ubuntu VPN Gateway

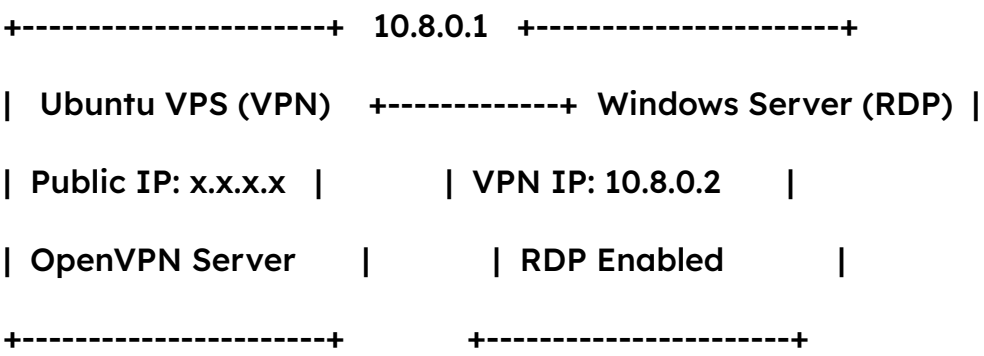
**A private RDP gateway secured via OpenVPN, where only authenticated VPN clients can access the internal Windows RDP server — all communications flow inside a VPN subnet (10.8.0.0/24), ensuring zero public exposure.**

## Purpose

This setup is designed to provide **secure remote access** to a Windows Server via **Remote Desktop Protocol (RDP)**, while ensuring that:

- The Windows Server is **not exposed to the public internet**
- Only **authenticated VPN clients** can access RDP
- All communication is **encrypted end-to-end** using OpenVPN






## How Access Works

1. **VPN Connection**:
   You connect to the OpenVPN server on Ubuntu, receive an IP in the range `10.8.0.0/24` (e.g. `10.8.0.3`).

**RDP Access**:
Since the Windows server is also on the VPN (e.g. `10.8.0.2`), you can connect via RDP:

nginx
CopyEdit
```
Remote Desktop to: 10.8.0.2
```

2.
3. **Access Control**:
   No VPN = No access. Even port scanners on the internet can't reach the Windows Server.

## Benefits

- **Zero Public Exposure** of the RDP server

- **Encrypted Communication** over VPN
- **Access Control via VPN Auth** (e.g. certificates, MFA)
- **Easy to Expand** — more clients or servers can be added into the private VPN network

## How To Make It

So we will use open-source version and will host it on our ubuntu server will make a step by step guide how to do it

1>Bash command:
wget https://git.io/vpn -O openvpn-install.sh

2>Bash command:
sudo chmod +x openvpn-install.sh

3>Bash command:
sudo bash openvpn-install.sh

Now We Have To **NAT forward the OpenVPN port** on our **Ubuntu VPS** through the command line, we'll mainly be using **iptables.**

## 1. Enable IP Forwarding

Edit the system config to allow IP forwarding:

bash
```
sudo sysctl -w net.ipv4.ip_forward=1
```

To make it permanent:

bash
```
echo "net.ipv4.ip_forward = 1" | sudo tee -a /etc/sysctl.conf
```

```
sudo sysctl -p
```

## 2. Use `iptables` to Forward the Port

**Assume:**

- **Your public interface** is `eth0`
- **Your OpenVPN port** is `1194` **(default)**
- **OpenVPN server is running on internal IP** `10.8.0.1`

**Forward UDP port** `1194`**:**

**bash**
**CopyEdit**

```
# Forward incoming VPN requests to internal VPN server
sudo iptables -t nat -A PREROUTING -i eth0 -p udp --dport 1194 -j DNAT
--to-destination 10.8.0.1:1194

# Allow forwarding traffic from public interface to internal VPN
sudo iptables -A FORWARD -i eth0 -p udp --dport 1194 -d 10.8.0.1 -j
ACCEPT

# Enable masquerading for outgoing traffic from VPN server
sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j
MASQUERADE
```

> **Adjust** `eth0`**,** `10.8.0.1`**, and port if different.**

## 3. Save iptables Rules (Optional)

**If using** `iptables-persistent`**:**

bash
CopyEdit

```
sudo apt install iptables-persistent
sudo netfilter-persistent save
```

Or manually save with:

bash
CopyEdit

```
sudo iptables-save > /etc/iptables/rules.v4
```

---

✅ **Confirm Forwarding Works**

- Verify with `iptables -t nat -L -n -v`
- Use `sudo tcpdump -i eth0 udp port 1194` to watch for incoming traffic

## AFTER EVERY THING IS DONE WE CAN MAKE THE CLIENT WITH SIMPLY RUNING 1 COMMAND sudo bash openvpn-install.sh THEN SELECT MAKE A NEW CLIENT

## AND TO GET CLIENT .OVPN FILE TO HOME DIRECTORY:

Bash:

```
sudo cp /root/client1.ovpn ~
```