**WireGuard VPN Setup on Ubuntu**

**Introduction**

This document outlines the steps involved in setting up a WireGuard VPN on an Ubuntu server. WireGuard is a modern VPN technology known for its simplicity and security. This guide will cover the server configuration, client configuration, and basic troubleshooting.

**Prerequisites**

* An Ubuntu server with internet access.
* Basic understanding of Linux command line.
* A static IP address for the server (recommended).

**Server Configuration**

**Installation**

1. Update the package lists:

Bash

sudo apt update

1. Install WireGuard:

Bash

sudo apt install wireguard

**Generate Keys**

1. Create a private key:

Bash

wg genkey | tee privatekey | wg pubkey > publickey

* + **Note:** Keep the privatekey file secure. It's the server's private key.

1. Create a configuration file (e.g., /etc/wireguard/wg0.conf):

Bash

sudo nano /etc/wireguard/wg0.conf

Paste the following content, replacing placeholders with your actual values:

[Interface]

Address = 10.0.0.1/24

SaveConfig = true

ListenPort = 51820

PrivateKey = YOUR\_PRIVATE\_KEY

[Peer]

PublicKey = CLIENT\_PUBLIC\_KEY

AllowedIPs = 10.0.0.2/32

* + Replace YOUR\_PRIVATE\_KEY with the content of your privatekey file.
  + Replace CLIENT\_PUBLIC\_KEY with the public key of the client you'll generate later.
  + The AllowedIPs line specifies the IP address range for the client.

**Firewall Configuration**

1. Allow WireGuard traffic through the firewall (e.g., UFW):

Bash

sudo ufw allow 51820/udp

**Start WireGuard**

1. Start the WireGuard interface:

Bash

sudo wg-quick up wg0

1. Enable the WireGuard service to start on boot:

Bash

sudo systemctl enable wg-quick@wg0

**Client Configuration**

**Installation**

1. Install WireGuard on the client machine:

Bash

sudo apt update && sudo apt install wireguard

**Generate Keys**

1. Generate a private and public key pair for the client:

Bash

wg genkey | tee privatekey | wg pubkey > publickey

**Configure the Client**

1. Create a configuration file (e.g., /etc/wireguard/wg0.conf):

Bash

sudo nano /etc/wireguard/wg0.conf

Paste the following content, replacing placeholders with your actual values:

[Interface]

Address = 10.0.0.2/24

PrivateKey = YOUR\_PRIVATE\_KEY

[Peer]

PublicKey = SERVER\_PUBLIC\_KEY

AllowedIPs = 0.0.0.0/0

Endpoint = SERVER\_IP:51820

* + Replace YOUR\_PRIVATE\_KEY with the client's private key.
  + Replace SERVER\_PUBLIC\_KEY with the server's public key.
  + Replace SERVER\_IP with the server's public IP address.

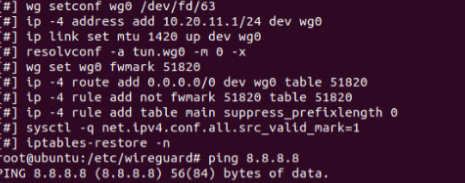
**Start WireGuard**

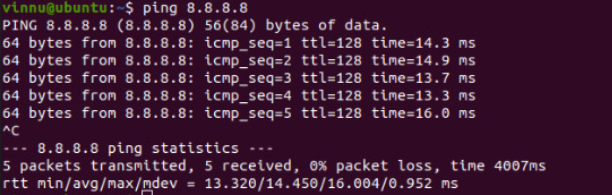
1. Start the WireGuard interface on the client:

Bash

sudo wg-quick up wg0

**Testing the Connection**

****

****

To verify the connection, try pinging the server's IP address from the client, or vice versa.

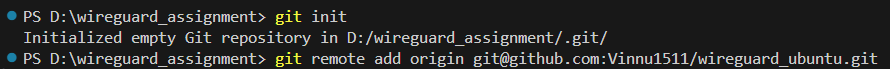
**Uploading Your Document to GitHub from the Terminal**

**Prerequisites**

* A GitHub account
* Git installed on your system
* A text file containing the WireGuard VPN setup documentation

**Steps**

1. **Create a GitHub Repository:**
   * Log in to your GitHub account.
   * Click on the "New repository" button.
   * Give your repository a name (e.g., "wireguard-setup-doc").
   * Optionally, add a description.
   * Choose whether to make the repository public or private.
   * Click "Create repository."
2. **Clone the Repository to Your Local Machine:**
   * Open your terminal to create and initialize git repo.



* + Navigate to the desired directory where you want to clone the repository.
  + Use the following command to clone the repository:

Bash

git clone https://github.com/your\_username/wireguard-setup-doc.git

Replace your\_username with your actual GitHub username and wireguard-setup-doc with your repository name.

1. **Copy the Document to the Repository:**
   * Copy the WireGuard VPN setup documentation text file to the cloned repository directory.
2. **Stage the File:**
   * In your terminal, navigate to the repository directory.
   * Use the following command to stage the file:

Bash

git add your\_document\_name.txt

Replace your\_document\_name.txt with the actual name of your document.

1. **Commit the Changes:**
   * Commit the staged changes with a descriptive message:

Bash

git commit -m "Added WireGuard VPN setup documentation"

1. **Push to GitHub:**
   * Push the committed changes to the remote repository:

Bash

git push origin main

Replace main with the name of your default branch if it's different.

**Additional Tips**

* You can use git add . to stage all files in the current directory.
* For larger files or multiple files, consider using tools like git large-file-storage or git lfs.
* Use descriptive commit messages to track changes effectively.
* Regularly push your changes to keep your remote repository up-to-date.

**Now you have successfully uploaded your WireGuard VPN setup documentation to GitHub.**