Home Server Documentation

# 1. Introduction / Overview

This document provides details about the setup and configuration of my home server. It includes hardware specifications, operating system installation, services, configurations, and lessons learned during the process.

# 2. System Setup

## 2.1 Hardware Used

• Device: old acer notebook  
• CPU: inter i3-3110M  
• RAM: 6GB  
• Storage: 1TB HDD

## 2.2 Operating System Installation

• OS/Distro: Ubuntu server 24.04.03 LTS  
• Installation method: USB - Flash  
• Partitioning details: one partition - FULL

## 2.3 Network Setup

• Static IP configuration

Set up rooter static ip to MAC adress of the server  
• Router/firewall rules

None

# 3. Services / Applications

For each service installed, document the following:  
• Name (e.g., Nextcloud, Jellyfin, Nginx, Docker)  
• Purpose  
• Installation steps  
• Configuration details  
• Access method (IP, domain, ports)

## Wireguard

[WireGuard installation and configuration - on Linux](https://www.youtube.com/watch?v=bVKNSf1p1d0)

Purpose:

* Enable remote access to server and its services. Side usage is that it can also be used as a commercial VPN

Installation steps:

Server side:

1. Install wireguard

To install WireGuard on Ubuntu 20.04 LTS we need to execute the following commands on the Server and Client.

sudo apt install wireguard

1. Create a private and public key on Server

Before we can establish a secure tunnel with WireGuard we need to create a private and public key on both, Server and Client first. WireGuard comes with a simple tool that can easily generate these keys. Execute this on the Server and Client.

wg genkey | tee privatekey | wg pubkey > publickey

Be aware, you MUST NOT SHARE the private key with anyone! Make sure to store it in a secure way on both devices.

1. Configure the Server

Now you can configure the server, just add a new file   
 called /etc/wireguard/wg0.conf. Insert the following configuration lines and replace the <server-private-key> placeholder with the previously generated private key.

You need to insert a private IP address for the <server-ip-address> that doesn't interfere with another subnet. Next, replace the <public- interface> with your interface the server should listen on for incoming connections.

1. fff

# 4. Security & Maintenance

• Firewall configuration (UFW, iptables)  
• SSH configuration (keys, disable root login)  
• Backup strategy (tools, frequency, location)  
• Update/upgrade procedure

# 5. What I Have Done / Lessons Learned

• What worked well  
• Problems encountered  
• How I solved them  
• Plans for future improvements

# 6. Appendix

• Command history (if relevant)  
• Configuration snippets (nginx.conf, docker-compose.yml)  
• Useful references or documentation links