Setting Up a Home Server Using Debian Linux on an Old Laptop

Guru Swarupa

October 3, 2024

Abstract

This journal documents the process of converting a 10-year-old laptop into a fully functional home server using Debian Linux. The server hosts a variety of services for home automation, personal media streaming, cloud storage, ad-blocking, and more. By leveraging open-source technologies like Docker, Kodi, HomeAssistant, Jellyfin, Nextcloud, and others, I created a stable, secure, and accessible server with minimal overhead. The guide outlines the installation of services, the configuration of Docker containers, and the setup of network management using Nginx Proxy Manager and Tailscale.

Keywords: Home Server, Debian Linux, Docker, HomeAssistant, Jellyfin, Nextcloud, Pi-hole, Kodi, Nginx Proxy Manager, Tailscale, DuckDNS, Photoprism, Webmin, Homarr, qBittorrent

1 Introduction

Building a home server is a cost-effective way to centralize media, automate home appliances, and enhance personal digital storage. This journal explores how an old laptop running Debian Linux can be transformed into a versatile home server. It details the setup of various services using Docker, the importance of network security through Nginx Proxy Manager, and how Tailscale enables secure access via a VPN. By setting up services like Kodi for media streaming, HomeAssistant for automation, and Jellyfin for personal streaming, this project integrates home automation, media storage, and security into a single lightweight machine.

2 Hardware and Initial Setup

2.1 Laptop Specifications

- **Laptop Model:** 10-year-old Latitude 3540, with an Intel Core i3-4010u processor (Dual Core , 1.70GHz, 3M cache, 15W) and 4GB of DDR3 RAM.
- **Storage:** 120GB SSD used for the operating system, paired with a 500GB HDD for file storage and media libraries.
- **Network:** Wired Ethernet connection to ensure stable and high-speed network performance.
- **Power Supply:** Continuous power supply using an uninterruptible power supply (UPS) for reliability during outages.

2.2 Debian Linux Installation

Debian is a stable and lightweight Linux distribution, perfect for a minimal server setup.

1. **Download Debian ISO:** Visit Debian's official website and download the latest stable release (minimal ISO is recommended).

2. **Create a Bootable USB:** Use software like `Rufus` (for Windows) or `dd` (for Linux/macOS) to create a bootable USB from the ISO file.

3. **Installation:** Boot the laptop from the USB and follow the guided steps to install Debian. Select minimal installation (no desktop environment) to reduce system overhead. Partition the disk for flexibility, e.g., separate `/home`, `/var`, and `/srv`.

4. **Post-Installation Configuration:** Update the system:

```
sudo apt update && sudo apt upgrade
```

3 Server Applications Using Docker

Docker is a containerization platform that allows you to run multiple isolated services on your server. The following applications can be set up using Docker, and their installation process can be managed via a script from the `linutil` repository.

3.1 Installation Process

To simplify the installation process for Docker, SSH, and Samba, you can use a single script that I contributed, which handles all three applications.

1. **Run Installation Script:** Execute the following command to install Docker, SSH, and Samba:

```
curl -fsSL https://christitus.com/linux | sh
```

2. **Select Application Installations:** During the script execution, you will have the option to select the installations of Docker, SSH, and Samba under Application Setup.

3. **Automount Drive Script:** Additionally, this script includes an option to set up an automount drive script, which can be used to automatically mount a disk on boot.

3.2 Overview of Applications

- **Docker:** Docker allows you to run multiple isolated services on your server.
- **SSH:** Enabling secure shell (SSH) access allows you to manage your server remotely and securely.
- **Samba:** Samba facilitates file sharing across different operating systems on your network.

3.3 Docker Compose Configuration

Below is the Docker Compose configuration for various services:

3.4 HomeAssistant: Home Automation

Home Assistant is an open-source home automation platform that puts local control and privacy first. It integrates with a wide variety of smart devices and services, allowing users to create automations and manage their smart home effectively. Using Docker simplifies the installation and management of Home Assistant.

- **Docker Compose file:**

```
version: '3'
services:
  homeassistant:
    image: lscr.io/linuxserver/homeassistant:latest
    container_name: homeassistant
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Berlin
    volumes:
      - ./hass/config:/config
    restart: unless-stopped
    ports:
      -  '8123:8123'
```

- **Access HomeAssistant:** Open `http://<your_server_IP>:8123` to access the HomeAssistant dashboard. Configure your smart devices and automations from the web interface.

3.5 Jellyfin: Personal Media Server

Jellyfin is a free and open-source media server software that allows you to organize, manage, and stream your personal media collection, including videos, music, and photos. It is a community-driven project and is designed to be a self-hosted alternative to proprietary media servers like Plex and Emby.

- **Docker Compose file:**

```
    version: '3'
    services:
      jellyfin:
        image: lscr.io/linuxserver/jellyfin:latest
        container_name: jellyfin
        environment:
          - PUID=1000
          - PGID=1000
          - TZ=Europe/Berlin
        volumes:
          - ./jellyfin/config:/config
          - /media/hdd500/nextcloud/data/swarupa/files/Courses:/
            data/Courses
          - /media/hdd500/nextcloud/data/swarupa/files/Movies:/
             data/movies
        restart: unless-stopped
        ports:
          -  "8096:8096"
```

- **Access Jellyfin:** Go to `http://<your_server_IP>:8096` to set up your media library and start streaming.

3.6 Photoprism: Image Storage

PhotoPrism is an open-source application that allows you to manage, organize, and back up your personal photo collection. Utilizing artificial intelligence, PhotoPrism offers features that enhance photo organization, making it easier to find, tag, and access your images. It's a self-hosted solution, meaning you can run it on your server or local machine, giving you complete control over your photos.

- **Docker Compose file:**

```
version: '3'
services:
  photoprism:
    ## Use photoprism/photoprism:preview for testing preview
        builds:
    image: photoprism/photoprism:latest
    ## Don't enable automatic restarts until PhotoPrism has
       been properly configured and tested!
    ## If the service gets stuck in a restart loop, this
       points to a memory, filesystem, network, or database
       issue:
    ## https://docs.photoprism.app/getting-started/
       troubleshooting/#fatal-server-errors
    restart: unless-stopped
    stop_grace_period: 10s
    depends_on:
      - mariadb
    security_opt:
      - seccomp:unconfined
      - apparmor:unconfined
    ## Server port mapping in the format "Host:Container".
       To use a different port, change the host port on
    ## the left-hand side and keep the container port, e.g.
       "80:2342" (for HTTP) or "443:2342 (for HTTPS):
    ports:
      - "2342:2342"
    ## Before you start the service, please check the
       following config options (and change them as needed):
    ## https://docs.photoprism.app/getting-started/config-
       options/
    environment:
      PHOTOPRISM_ADMIN_USER: "USERNAME"                 # admin login username
      PHOTOPRISM_ADMIN_PASSWORD: "PASSWORD"          # initial admin password (8-72 characters)
      PHOTOPRISM_AUTH_MODE: "password"               # authentication mode (public, password)
      PHOTOPRISM_SITE_URL: "http://localhost:2342/"  # server URL in the format "http(s)://domain.name(:port)/(path)"
      PHOTOPRISM_DISABLE_TLS: "false"                # disables HTTPS/TLS even if the site URL starts with https:// and a certificate is available
      PHOTOPRISM_DEFAULT_TLS: "true"                 # defaults to a self-signed HTTPS/TLS certificate if no other certificate is available
      PHOTOPRISM_ORIGINALS_LIMIT: 5000               # file size limit for originals in MB (increase for high-res video)
      PHOTOPRISM_HTTP_COMPRESSION: "gzip"            # improves transfer speed and bandwidth utilization (none or gzip)
      PHOTOPRISM_LOG_LEVEL: "info"                   # log level: trace, debug, info, warning, error, fatal, or panic
      PHOTOPRISM_READONLY: "false"                   # do not modify originals directory (reduced functionality)
      PHOTOPRISM_EXPERIMENTAL: "false"               # enables experimental features
      PHOTOPRISM_DISABLE_CHOWN: "false"              # disables updating storage permissions via chmod and chown on startup
      PHOTOPRISM_DISABLE_WEBDAV: "false"             # disables built-in WebDAV server
      PHOTOPRISM_DISABLE_SETTINGS: "false"           # disables settings UI and API
      PHOTOPRISM_DISABLE_TENSORFLOW: "false"         # disables all features depending on TensorFlow
      PHOTOPRISM_DISABLE_FACES: "false"              # disables face detection and recognition (requires TensorFlow)
      PHOTOPRISM_DISABLE_CLASSIFICATION: "false"     # disables image classification (requires TensorFlow)
      PHOTOPRISM_DISABLE_VECTORS: "false"            # disables vector graphics support
      PHOTOPRISM_DISABLE_RAW: "false"                # disables indexing and conversion of RAW images
      PHOTOPRISM_RAW_PRESETS: "false"                # enables applying user presets when converting RAW images (reduces performance)
      PHOTOPRISM_SIDECAR_YAML: "true"                # creates YAML sidecar files to back up picture metadata
      PHOTOPRISM_BACKUP_ALBUMS: "true"               # creates YAML files to back up album metadata
      PHOTOPRISM_BACKUP_DATABASE: "true"             # creates regular backups based on the configured schedule
      PHOTOPRISM_BACKUP_SCHEDULE: "daily"            # backup SCHEDULE in cron format (e.g. "0 12 * * *" for daily at noon) or at a random time (daily, weekly)
      PHOTOPRISM_INDEX_SCHEDULE: ""                  # indexing SCHEDULE in cron format (e.g. "@every 3h" for every 3 hours; "" to disable)
      PHOTOPRISM_AUTO_INDEX: 5                       # delay before automatically indexing files in SECONDS when uploading via WebDAV (-1 to disable)
      PHOTOPRISM_AUTO_IMPORT: -1                     # delay before automatically importing files in SECONDS when uploading via WebDAV (-1 to disable)
      PHOTOPRISM_DETECT_NSFW: "false"                # automatically flags photos as private that MAY be offensive (requires TensorFlow)
      PHOTOPRISM_UPLOAD_NSFW: "true"                 # allows uploads that MAY be offensive (no effect without TensorFlow)
      # PHOTOPRISM_DATABASE_DRIVER: "sqlite"         # SQLite is an embedded database that does not require a separate database server
      PHOTOPRISM_DATABASE_DRIVER: "mysql"            # MariaDB 10.5.12+ (MySQL successor) offers significantly better performance compared to SQLite
      PHOTOPRISM_DATABASE_SERVER: "mariadb:3306"     # MariaDB database server (hostname:port)
      PHOTOPRISM_DATABASE_NAME: "photoprism"         # MariaDB database schema name
      PHOTOPRISM_DATABASE_USER: "photoprism"         # MariaDB database user name
      PHOTOPRISM_DATABASE_PASSWORD: "PASSWORD"       # MariaDB database user password
      PHOTOPRISM_SITE_CAPTION: "AI-Powered Photos App"
      PHOTOPRISM_SITE_DESCRIPTION: ""                # meta site description
      PHOTOPRISM_SITE_AUTHOR: ""                     # meta
```

```
site author
      ## Video Transcoding (https://docs.photoprism.app/getting-started/advanced/transcoding/):
      # PHOTOPRISM_FFMPEG_ENCODER: "software"        # H.264/AVC encoder (software, intel, nvidia, apple, raspberry, or vaapi)
      # PHOTOPRISM_FFMPEG_SIZE: "1920"               # video size limit in pixels (720-7680) (default: 3840)
      # PHOTOPRISM_FFMPEG_BITRATE: "32"              # video bitrate limit in Mbit/s (default: 50)
      ## Run/install on first startup (options: update https gpu ffmpeg tensorflow davfs clitools clean):
      # PHOTOPRISM_INIT: "https gpu tensorflow"
      ## Run as a non-root user after initialization (supported: 0, 33, 50-99, 500-600, and 900-1200):
      # PHOTOPRISM_UID: 1000
      # PHOTOPRISM_GID: 1000
      # PHOTOPRISM_UMASK: 0000
    ## Start as non-root user before initialization (supported: 0, 33, 50-99, 500-600, and 900-1200):
    # user: "1000:1000"
    ## Share hardware devices with FFmpeg and TensorFlow (optional):
    # devices:
    #  - "/dev/dri:/dev/dri"                         # Intel QSV
    #  - "/dev/nvidia0:/dev/nvidia0"                 # Nvidia CUDA
    #  - "/dev/nvidiactl:/dev/nvidiactl"
    #  - "/dev/nvidia-modeset:/dev/nvidia-modeset"
    #  - "/dev/nvidia-nvswitchctl:/dev/nvidia-nvswitchctl"
    #  - "/dev/nvidia-uvm:/dev/nvidia-uvm"
    #  - "/dev/nvidia-uvm-tools:/dev/nvidia-uvm-tools"
    #  - "/dev/video11:/dev/video11"                 # Video4Linux Video Encode Device (h264_v4l2m2m)
    working_dir: "/photoprism" # do not change or remove
    ## Storage Folders: "~" is a shortcut for your home directory, "." for the current directory
    volumes:
      # "/host/folder:/photoprism/folder"                # Example
      - "/media/hdd500/nextcloud/data/srivasa/files/Photos:/photoprism/originals"               # Original media files (DO NOT REMOVE)
      # - "/example/family:/photoprism/originals/family" # *Additional* media folders can be mounted like this
      # - "~/Import:/photoprism/import"                  # *Optional* base folder from which files can be
```

```
imported to originals
      - "~/.storage:/photoprism/storage"                  # *Writable* storage folder for cache, database, and sidecar files (DO NOT REMOVE)
  ## MariaDB Database Server (recommended)
  ## see https://docs.photoprism.app/getting-started/faq/#should-i-use-sqlite-mariadb-or-mysql
  mariadb:
    image: mariadb:11
    ## If MariaDB gets stuck in a restart loop, this points to a memory or filesystem issue:
    ## https://docs.photoprism.app/getting-started/troubleshooting/#fatal-server-errors
    restart: unless-stopped
    stop_grace_period: 5s
    security_opt: # see https://github.com/MariaDB/mariadb-docker/issues/434#issuecomment-1136151239
      - seccomp:unconfined
      - apparmor:unconfined
    command: --innodb-buffer-pool-size=512M --transaction-isolation=READ-COMMITTED --character-set-server=utf8mb4 --collation-server=utf8mb4_unicode_ci --max-connections=512 --innodb-rollback-on-timeout=OFF --innodb-lock-wait-timeout=120
    ## Never store database files on an unreliable device such as a USB flash drive, an SD card, or a shared network folder:
    volumes:
      - "./database:/var/lib/mysql" # DO NOT REMOVE
    environment:
      MARIADB_AUTO_UPGRADE: "1"
      MARIADB_INITDB_SKIP_TZINFO: "1"
      MARIADB_DATABASE: "photoprism"
      MARIADB_USER: "photoprism"
      MARIADB_PASSWORD: "PASSWORD"
      MARIADB_ROOT_PASSWORD: "PASSWORD"

  ## Watchtower upgrades services automatically (optional)
  ## see https://docs.photoprism.app/getting-started/updates/#watchtower
  ## activate via "COMPOSE_PROFILES=update docker compose up -d"
  watchtower:
    restart: unless-stopped
    image: containrrr/watchtower
    profiles: ["update"]
    environment:
      WATCHTOWER_CLEANUP: "true"
```

```
          WATCHTOWER_POLL_INTERVAL: 7200 # checks for updates
             every two hours
        volumes:
          - "/var/run/docker.sock:/var/run/docker.sock"
          - "~/.docker/config.json:/config.json" # optional, for
              authentication if you have a Docker Hub account
```

- **Access Photoprism:** Visit http://<your_server_IP>:2342 to upload and manage your photos.

3.7 Nextcloud: Personal Cloud Storage

Nextcloud is an open-source platform that enables you to store, synchronize, and share files securely across multiple devices. Unlike proprietary solutions like OneDrive or Google Drive, Nextcloud gives you complete control over your data by allowing you to host it on your own server. This makes it a popular choice for individuals and organizations that prioritize privacy and security.

- **Docker Compose file:**

```
version: '3'
services:
  nextcloud:
    image: lscr.io/linuxserver/nextcloud:latest
    container_name: nextcloud
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Berlin
    volumes:
      - /media/hdd500/nextcloud/appdata:/config
      - /media/hdd500/nextcloud/data:/data
    restart: unless-stopped
```

- **Access Nextcloud:** Visit http://<your_server_IP>:8080 to configure Nextcloud for cloud storage.

4 Reverse Proxy and VPN Setup

4.1 Nginx Proxy Manager: Reverse Proxy

Nginx Proxy Manager is an open-source application that provides a user-friendly interface to manage Nginx proxies. It allows you to easily configure and manage multiple web services hosted on your server, ensuring secure access with HTTPS and customizable routing. It's particularly useful for users who want to access their self-hosted applications remotely without dealing with complex Nginx configurations.

- **Docker Compose file:**

```
version: '3'
services:
  nginxproxymanager:
    image: 'jc21/nginx-proxy-manager:latest'
    container_name: nginxproxymanager
    restart: unless-stopped
    ports:
      - '100:100'
      - '81:81'
      - '443:443'
    volumes:
      - ./nginx/data:/data
      - ./nginx/letsencrypt:/etc/letsencrypt
```

- **Configure Domains and SSL:** Access the UI at `http://<your_server_IP>:81`, configure your domain names, and add SSL certificates via Let's Encrypt.

4.2 Tailscale: VPN

Tailscale is a secure mesh VPN service that simplifies the process of connecting devices over the internet without the need for traditional VPN configurations or complex networking setups. By leveraging the WireGuard protocol, Tailscale enables secure, private access to your home server or any device in your network from anywhere in the world.

- **Install Tailscale:**

```
curl -fsSL https://tailscale.com/install.sh | sh
```

- **Start and Login:**

```
sudo tailscale up
```

- **Access Your Server:** Connect to your server securely using Tailscale from any device.

4.3 Pihole: Network ad Blocker , DNS sinkhole

Pi-hole is a network-wide ad blocker that acts as a DNS sinkhole. It effectively prevents unwanted content, such as advertisements and tracking scripts, from being loaded on any device connected to your network. By filtering DNS queries, Pi-hole blocks ads at the network level, making it a powerful tool for enhancing privacy and improving browsing speeds.

- **Docker Compose file:**

```
version: '3'
services:
  pihole:
    container_name: pihole
    image: pihole/pihole:latest
    # For DHCP it is recommended to remove these ports and instead add: network_mode: "host"
    ports:
      - "53:53/tcp"
      - "53:53/udp"
      - "67:67/udp" # Only required if you are using Pi-hole as your DHCP server
      - "4040:80/tcp"
    environment:
      TZ: 'America/Chicago'
      WEBPASSWORD: 'PASSWORD'
    # Volumes store your data between container upgrades
    volumes:
      - './etc-pihole:/etc/pihole'
      - './etc-dnsmasq.d:/etc/dnsmasq.d'
    #   https://github.com/pi-hole/docker-pi-hole#note-on-capabilities
    cap_add:
      - NET_ADMIN # Required if you are using Pi-hole as your DHCP server, else not needed
    restart: unless-stopped
```

- **Configure Pihole:** Access the UI at `http://<your_server_IP>:4040`, configure Pihole here.

4.4 QBittorrent: BitTorrent client

qBittorrent is a free and open-source BitTorrent client designed for downloading and uploading files using the BitTorrent protocol. It provides a user-friendly interface while incorporating a variety of advanced features, making it a popular choice among torrent users.

- **Docker Compose file:**

```
version: '3'
services:
  qbittorrent:
    image: lscr.io/linuxserver/qbittorrent:latest
    container_name: qbittorrent
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Etc/UTC
      - WEBUI_PORT=5000
      - TORRENTING_PORT=6881
    volumes:
      - /path/to/qbittorrent/appdata:/config
      - /media/hdd500/nextcloud/data/swarupa/files/Downloads:/downloads #optional
    ports:
      - 5000:5000
      - 6881:6881
      - 6881:6881/udp
    restart: unless-stopped
```

- **Access qbitTorrent:** Access the UI at http://<your_server_IP>:5000,Access QbitTorrent web portal here.

4.5 Webmin: Web-Based Administration Tool

Webmin is a web-based interface for system administration on Unix-like systems. It allows you to manage your server through a web interface, providing access to various services and configuration files without needing to edit them manually.

- **Docker Compose file:**

```
version: '3'
services:
  webmin:
    image: jc21/webmin:latest
    container_name: webmin
    ports:
      - "10000:10000"
    environment:
      - WEBMIN_USER=admin
      - WEBMIN_PASS=PASSWORD
    restart: unless-stopped
```

- **Configure Webmin:** Access the UI at http://<your_server_IP>:10000, and log in with the username and password specified in the environment variables.

4.6 Homarr: Self-Hosted Application Dashboard

Homarr is a self-hosted application dashboard for organizing and accessing your favorite web applications. It supports various authentication methods, including SSH and Samba, allowing for streamlined management of applications.

- **Docker Compose file for Homarr:**

```
version: '3'
services:
  homarr:
    image: homarr/homarr:latest
    container_name: homarr
    environment:
      - NODE_ENV=production
      - HOMARR_DB=/data/homarr.db
      - HOMARR_BACKEND_URL=http://<your_server_IP>:8080
    volumes:
      - /path/to/homarr/data:/data
    ports:
      - "7575:7575"
    restart: unless-stopped
```

- **Access Homarr:** Access the UI at `http://<your_server_IP>:7575`.

5 Conclusion

Setting up a home server using an old laptop and Debian Linux is an affordable and efficient way to centralize media, cloud storage, and home automation. Docker containers simplify the management of services like HomeAssistant, Jellyfin, and Nextcloud. By incorporating a reverse proxy with Nginx Proxy Manager and secure access through Tailscale, the server is both accessible and secure for home use.