# FileBrowser Deployment Guide on Debian 12 Home Server

by GrapheneGoat.com

This guide will walk you through deploying your own home private server to host files that you can access and share from anywhere, on the internet, using your own custom domain.

The Prerequisites section walks you through how to set up a Linux server using Full Disk Encryption (FDE) so that your files are encrypted on the server side. Files, during transfer over the internet, are encrypted with HTTPS through a secure Cloudflare tunnel.

This guide covers the following easy steps:
-Debian 12 Linux install on a desktop computer (your 'host' server)
- Domain registration (which you can also use for many other projects besides this one)
- Cloudflare account creation and setup
- Installing Docker and FileBrowser on Debian 12
- Setting up Cloudflare Tunnel and DNS for public access
- Creating users, setting permissions
- Security (fail2ban, SSH best practices, updates)
- Keeping everything updated

## Prerequisites

### CHAPTER 0: Installing Debian 12 with Full Disk Encryption (LUKS)

**(If you already have a server or know how to install Linux, skip to Chapter 1)**

Before You Begin

**Back up your data!** This process will erase your entire disk on the host server computer.

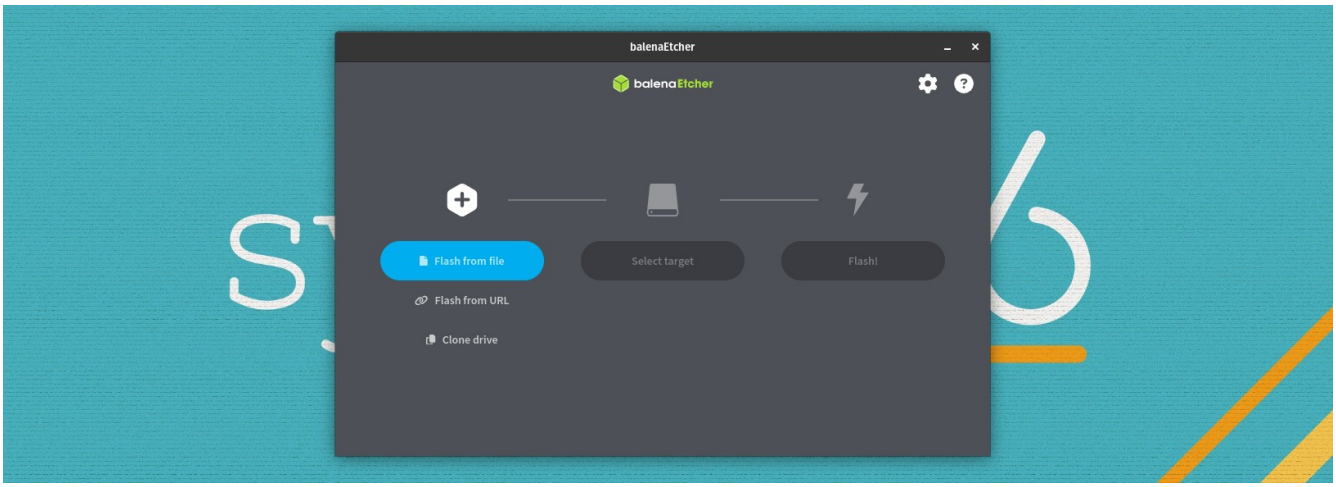
**- You'll need:**
**- A blank USB drive (8GB or more)**
**-Balena Etcher software to create the boot disk from the USB drive**
**- The computer you want to install Debian 12 on (your soon to be host server)**
**-A Password Manager to capture all of your passwords for this project**

Step 0.1: Download Debian 12 ISO

1. On any computer, go to Debian's official download page (https://www.debian.org/download)
2. Download "64-bit PC netinst ISO" (for most desktop/laptops, called `amd64`).

Step 0.2: Create a Bootable USB Installer

- Windows, Linux or macOS: Use Balena Etcher (https://www.balena.io/etcher/).
-Insert USB drive, open Balena Etcher.
-Click **'Flash from file'** button and select the Debian 12 ISO image
-Click **'Select Target'** and (carefully) select the USB drive
-Click **'Flash!'** and in about 5 minutes you'll have a working boot drive to install Debian 12




Step 0.3: Boot the Computer from USB

1. Insert the USB stick into your desktop computer.
2. Turn it on, and press the key for boot menu (often F12, F10, ESC, or DEL).
   -See the chart at the bottom of this page to find the boot key for your computer
3. Select your USB stick as the priority in the boot sequence and boot.

Step 0.4: Start Debian Installation

- On the boot menu, choose **Graphical install**.

Step 0.5: Go Through Basic Setup

- Choose your language, location, and keyboard.
- Set up your hostname (e.g., `debian-server`, default is also fine).
- Choose your root/admin account password and create a regular user.

Step 0.6: Disk Partitioning with Encryption

1. When you reach **Partition disks**, choose:
   - **Guided - use entire disk and set up encrypted LVM**
2. Select the hard drive you want to use (be very careful here!).
3. Choose "**All files in one partition**" (recommended for new users).
4. On the "**Write the changes to disks and configure encrypted volumes**" screen, select **Yes**.
5. **Enter your encryption passphrase** (choose something long and secure, write it down!).
   - You'll need this every time you boot!
6. The installer will now erase the drive and set up LUKS encryption and LVM.

Step 0.7: Finish Installation

- Continue with the installation:
  - Choose your country's mirror (any are fine).
  - Let it complete installing the OS.
  - Select to *install the GRUB boot loader* when prompted.

- When done, eject your USB and reboot!

Step 0.8: First Boot

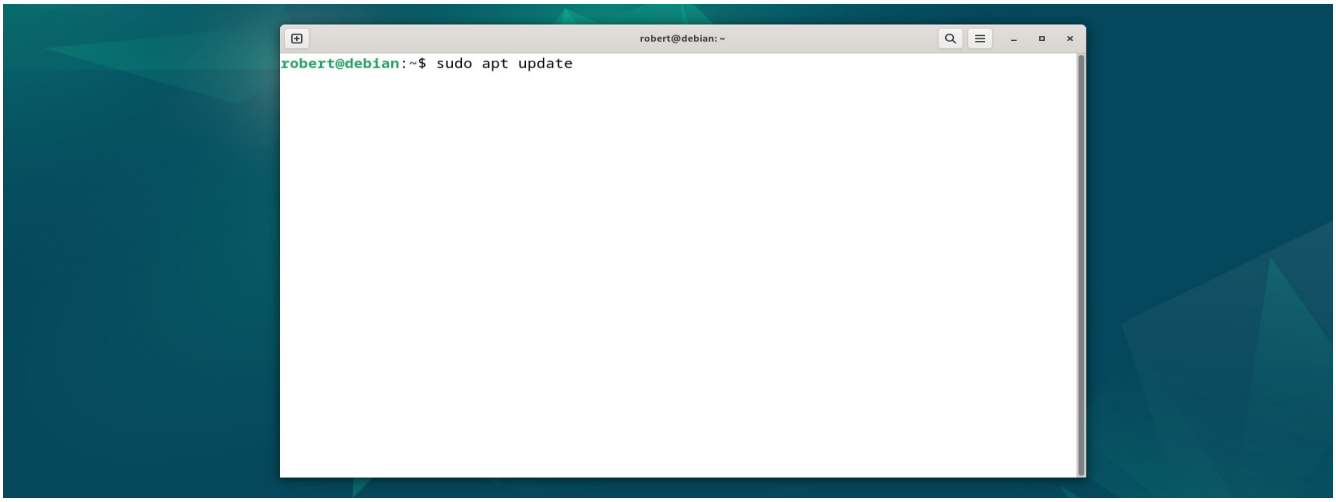
- On startup, you will be prompted for your **encryption passphrase** before the operating system loads. Without this password, no one can access your machine or your data, this is ‘Full Disk Encryption’ using LUKS (Linux Unified Key Setup)
- Enter your passphrase and Debian will unlock the disk and start normally.

After Boot: Add user to **sudoers** file and Update the System

Open a Terminal and run each command:

```
su -
usermod -aG sudo $USER
exit  (returns you to normal user mode)

sudo apt update
sudo apt upgrade -y
```




Now you can proceed to the next section of the guide (Docker, FileBrowser, etc).

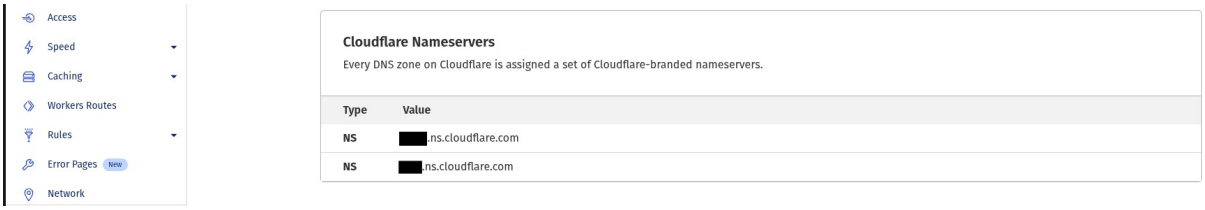
Now your server is securely encrypted – proceed with FileBrowser and Cloudflare setup!

## Deploying FileBrowser Securely on Debian 12

### CHAPTER 1: Domain Registration & Cloudflare Setup

Step 1: Buying a Domain Name
1. Visit a domain registrar (like Namecheap, Porkbun, OrangeWebsite, etc).
2. Search for and purchase your domain (e.g., `yourcustomdomain.com`).
3. Recommend picking a sub-domain to use (e.g., ‘files.yourcustomdomain.com’).

Step 2: Create a Cloudflare Account
1. Go to Cloudflare (https://cloudflare.com) and sign up.
2. Add your new domain to your Cloudflare account.
3. Follow Cloudflare's step-by-step instructions.
****Update your domain's nameservers at your registrar to point to Cloudflare's nameservers** (Cloudflare will tell you which ones).
4. Wait about 10–20 minutes for domain verification. (In rare cases this can take up to 48hrs)




*Copy/Paste the nameservers from Cloudflare into the nameservers on your registrar*

Cloudflare now manages your domain's DNS (Domain Naming System, think of it as the phone book of the internet. When you enter 'files.yourcustomdomain.com' DNS translates that into an IP address to find the appropriate server.) You don't need to change anything else at your domain registrar for this tutorial.

## CHAPTER 2: Set Up Debian 12 Server

Step 3: Log Into Your Server

- If local: Open the Terminal application.
- If remote: Use SSH from another computer (`ssh your_username@your_server_ip`).

Step 4: Update Your Server

Copy and paste these commands, one at a time:

```
sudo apt update
sudo apt upgrade -y
```

(Always keep your server system up to date).

## CHAPTER 3: Install Docker

Step 5: Install Docker and Docker Compose

```
sudo apt install -y docker.io
```

After install, make sure Docker starts automatically:

```
sudo systemctl enable --now docker
```

Add your user to the docker group so you don’t need sudo (recommended for local setup):

```
sudo usermod -aG docker $USER
```

Log out and back in, or restart your computer, for this to take effect.

**CHAPTER 4: Install File Browser via Docker**

Step 6: Prepare Directories for File Browser

Pick where you want File Browser to "see" your files and where its database/configuration lives. For this example, we’ll use `/srv/filebrowser-data` for your files and config.

```
sudo mkdir -p /srv/filebrowser-data/files
sudo mkdir -p /srv/filebrowser-data/config
sudo touch /srv/filebrowser-data/config/filebrowser.db
sudo chown -R $USER:$USER /srv/filebrowser-data
```

Step 7: Run File Browser in Docker

Copy and paste:

```
docker run -d \
  --name filebrowser \
  -v /srv/filebrowser-data/files:/srv \
  -v /srv/filebrowser-data/config/filebrowser.db:/database.db \
  -p 8081:80 \
  --restart unless-stopped \
  filebrowser/filebrowser
```

This command:
- Downloads the filebrowser image
- Stores your files under `/srv/filebrowser-data/files`
- Stores your database under `/srv/filebrowser-data/config/filebrowser.db`
- Makes File Browser available on port 8081 on your local server

Step 8: Test File Browser Locally (on the host server machine)

Open a browser on your server (or another computer on the LAN) and visit:

http://SERVER_IP:8081  (replace SERVER_IP with your IP or Domain):

http://files.yourcustomdomain.com:8081 (notice it’s HTTP and not HTTPS)

- First login:
  - Username: ****admin****
  - Password: ****admin****

- Ensure to change the password immediately! (Critical)

**CHAPTER 5: Expose File Browser Securely to the Internet using Cloudflare Tunnel**

Step 9: Install Cloudflare Tunnel ("cloudflared")

```
sudo apt install -y curl

curl -L https://github.com/cloudflare/cloudflared/releases
/latest/download/cloudflared-linux-amd64.deb -o cloudflared.deb

sudo dpkg -i cloudflared.deb
```

Step 10: Login to Cloudflare Tunnel

```
cloudflared tunnel login
```

Follow the URL displayed to authenticate this server with your Cloudflare account.

Step 11: Create and Configure a Tunnel

Create a tunnel named `filebrowser-tunnel`:

```
cloudflared tunnel create filebrowser-tunnel
```

**Note the tunnel ID in the output of this command, you'll need this in a second.

Step 12: Configure The Tunnel To Forward to File Browser

Create configuration for the tunnel:

```
sudo mkdir -p /etc/cloudflared
sudo nano /etc/cloudflared/config.yml
```

Copy-paste into the file (change `<TUNNEL_ID>` and `<YOUR_USER>` as shown in the output and your home dir):

```
tunnel: <TUNNEL_ID>
credentials-file: /home/<YOUR_USER>/.cloudflared/<TUNNEL_ID>.json

ingress:
  - hostname: files.YOURDOMAIN.com
    service: http://localhost:8081
  - service: http_status:404
```

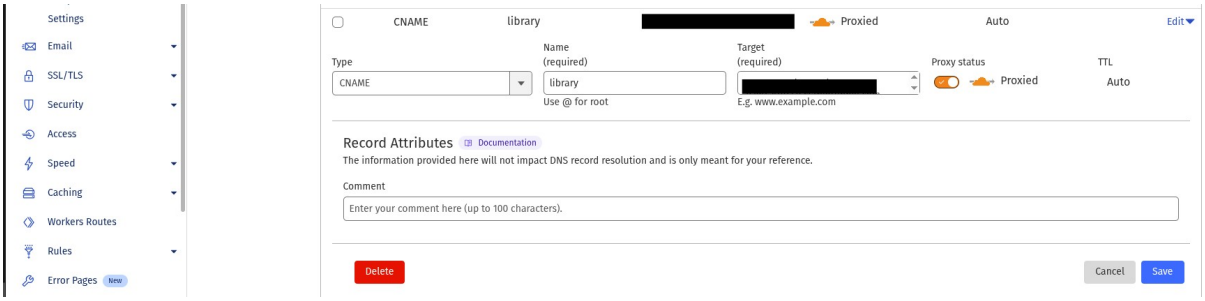
*Replace files.YOURDOMAIN.com with your custom subdomain that you want*

Step 13: Create the DNS Record (This Step is Magic with New Cloudflared)

Run (replace as needed):

```
cloudflared tunnel route dns filebrowser-tunnel files.YOURDOMAIN.com
```

This command automatically makes a **CNAME** record in Cloudflare pointing the subdomain to the tunnel. (Pictured below):




Step 14: Start the Tunnel as a Service

```
sudo cloudflared service install
sudo systemctl start cloudflared
sudo systemctl enable cloudflared
```

Now, **wait a minute**, then visit `https://files.YOURDOMAIN.com` from anywhere in the world!

## CHAPTER 6: Create User Accounts in File Browser

1. Log in as admin at your domain (`https://files.YOURDOMAIN.com`).
2. Open the **Admin Panel** (usually found via your user icon/top right).
3. Go to “Users” and click “Add.”
4. Set usernames, passwords, and directories they’re allowed to see.

Each user has their own login!

## CHAPTER 7: Secure Your Server

Step 15: Set up fail2ban (protects SSH login)

```
sudo apt install -y fail2ban
sudo systemctl enable --now fail2ban
```

- It runs out-of-the-box, protecting SSH.
- If you want, you can edit `/etc/fail2ban/jail.local` for more advanced settings.
- Since Cloudflare is providing front end protection, fail2ban is not critical to install, but offers another layer of security against unauthorized login attempts, ‘defense in depth’.

Step 16: Secure SSH further

1. Disable password logins and require keys:
   Open SSH config:

```
sudo nano /etc/ssh/sshd_config
```

   Find these two lines and set:

```
PermitRootLogin no
PasswordAuthentication no
```

   Save and restart:

```
sudo systemctl restart ssh
```

   *This step is optional but strongly recommended if you know how to use SSH keys.*

Step 17: Keep Everything Updated

- To update your server:

```
sudo apt update
sudo apt upgrade -y
```

- To update Docker containers:

```
docker pull filebrowser/filebrowser
docker stop filebrowser
docker rm filebrowser
```

Then re-run the Docker command from earlier:

```
docker run -d \
  --name filebrowser \
  -v /srv/filebrowser-data/files:/srv \
  -v /srv/filebrowser-data/config/filebrowser.db:/database.db \
  -p 8081:80 \
  --restart unless-stopped \
  filebrowser/filebrowser
```

**CHAPTER 8: Extra Security Recommendations**

- Use strong, unique passwords for all accounts.
- Limit user home directories ("scopes") to their own folders if privacy is needed.
- Enable Cloudflare Access/Zero Trust** in your Cloudflare dashboard for the files domain/subdomain.
  This gives you an extra login wall (Google/Microsoft login, 2FA) in front of File Browser.

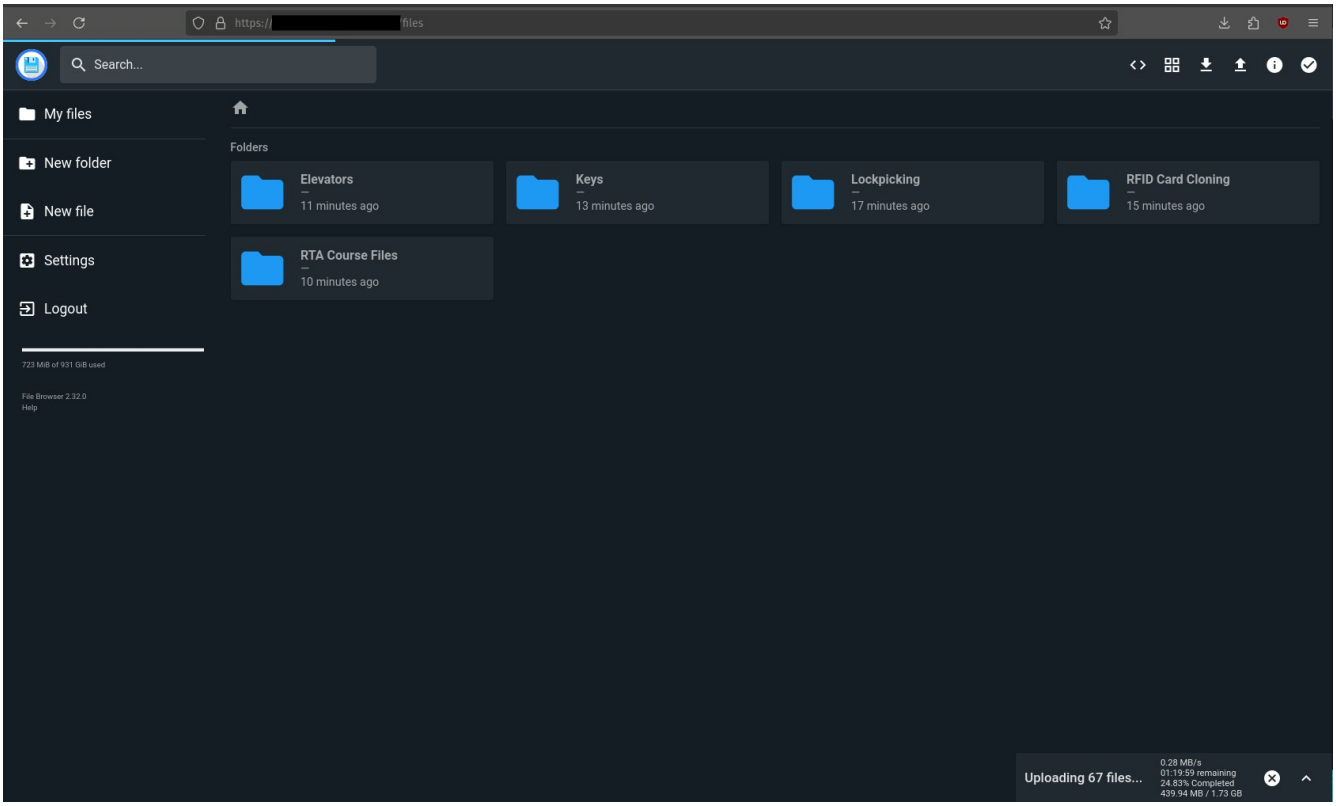
- Back up your `/srv/filebrowser-data/config/filebrowser.db` file regularly!

To do this just run the following command (includes a date/time stamp)

```
cp /srv/filebrowser-data/config/filebrowser.db ~/filebrowser.db.$
(date +%Y-%m-%d).backup
```

Summary Checklist

- [x] Bought a domain and set up Cloudflare
- [x] Set up Debian 12 and updated it
- [x] Installed Docker and File Browser
- [x] Installed and configured Cloudflare Tunnel
- [x] Created Cloudflare DNS record (via CLI or Cloudflare dashboard)
- [x] Created File Browser users and secured admin
- [x] Installed fail2ban and secured SSH
- [x] Learned how to keep everything updated




*Example screenshot of a FileBrowser instance accessed via a browser*

## Conclusion

You've now deployed your own cloud file server in a private, secure, modern way—even as a beginner! It's not super fancy, but is stable, free, open source and works excellent. There is no tracking,

telemetry or collection of your data or files, and you control the hardware as well. This is in staunch contrast to the abusive big tech options that Google, Microsoft, Apple and others offer.

**For those that require more features, consider doing a very similar install with something like Nextcloud, Etesync, Seafile or many other open source projects. You can even host them on the same server as the one you just created if you like, accessible at a different subdomain of your custom domain, e.g., ‘nextcloud.yourcustomdomain.com’ or restrict it to local network access only.

FileBrowser + Cloudflare Tunnel means you have:
- No open ports,
- Convenience of a web interface,
- Custom subdomain,
- Easy updates,
- And strong security.

**If you get stuck at any step, do each command exactly as shown, and restart from the last step that worked. Review terminal messages slowly—if anything says "error" or "permission denied," try to fix it and ask AI for help!

Good Luck, and Enjoy Your Private FileBrowser Cloud! 🚀

**GrapheneGoat.com**