

# Home Lab Applications and Services

This book documents the services, configurations, and infrastructure deployed in my home lab. Built on Proxmox VE, this environment allows me to test, deploy, and manage enterprise-like systems, including identity platforms, monitoring tools, collaboration apps, and file services.

Each section covers installation, configuration, and use case documentation to centralize my learning and provide technical transparency for future reference.

- [Audiobookshelf Installation via Proxmox VE Helper Scripts](#)
- [BookStack Installation on Ubuntu 24.04](#)
- [PairDrop Installation via Proxmox VE Helper-Scripts](#)
- [Paperless-ngx Installation on Debian 12 \(Using Official Script\)](#)
- [Snipe-IT Installation on Ubuntu 24.04](#)

# Audiobookshelf Installation via Proxmox VE Helper Scripts

Audiobookshelf is a self-hosted audiobook and podcast server with a web UI and apps. This guide shows how to deploy it in an **LXC container** using the [Proxmox VE Helper-Scripts](#).

## ? Step-by-Step Instructions

To create a new Proxmox VE Audiobookshelf LXC, run the command below in the Proxmox VE Shell.

```
bash -c "$(curl -fsSL https://raw.githubusercontent.com/community-scripts/ProxmoxVE/main/ct/audiobookshelf.sh)"
```

Location of config file: /usr/share/audiobookshelf/config

This launches an interactive installation wizard for **Audiobookshelf**.

## Fill in the Prompts

You'll be asked to confirm settings for:

Prompt	Recommendation / Example
Container ID	905 (or any available ID)
Hostname	e.g. audiobookshelf
Network Bridge	vmbr0 (default for most setups)
Static IP Address	e.g. 172.30.0.x
Gateway IP	e.g. 172.30.0.x
Storage Location	e.g. local-lvm
Enable Root SSH Access	Optional (yes for remote CLI access)
Media Folder Mount Point	Mount your media library or keep the default

- Access Audiobookshelf Web UI

```
http://<your-static-ip>:13378
```

Transfer File to the Audiobookshelf server:

```
scp "File path" user@<IP of the server>:./<Media Folder/
```

# BookStack Installation on Ubuntu 24.04

## ? Description:

This page documents the complete process for installing **BookStack**, a self-hosted wiki and knowledge base, on a fresh Ubuntu 24.04 server using PHP, Nginx, and MariaDB. This service is used as my primary knowledge repository.

ref. <https://www.bookstackapp.com/docs/admin/installation/>

## ? Prerequisites

- Fresh Ubuntu 24.04 LTS server (minimal install) LXC
- Root or sudo access
- Static IP address

## Ubuntu 24.04 Installation Script

A script to install BookStack on a fresh instance of Ubuntu 24.04 is available. This script is ONLY FOR A FRESH OS it will install Apache, MySQL 8.0 & PHP 8.3 and could OVERWRITE any existing web setup on the machine. It also does not set up mail settings or configure system security, so you will have to do those separately. You can use the script as a reference if you're installing on a non-fresh machine.

## Running the Script

```
# Ensure you have read the above information about what this script does before executing
these commands.

# Download the script
wget https://codeberg.org/bookstack/devops/raw/branch/main/scripts/installation-ubuntu-
24.04.sh

# Make it executable
chmod a+x installation-ubuntu-24.04.sh
```

```
# Run the script with admin permissions
sudo ./installation-ubuntu-24.04.sh
```

The script will output a log file for debugging within your current working directory when running the script. Permissions for the BookStack installation files & folders will be set based on the user used to run the script.

## ? Change your BookStack instance URL

### 1. Set up a domain name (DNS)

Make sure that `wiki.example.com` points to your server's **public IP address** (private/internal). In my case, I used Cloudflare tunnel.

### 2. Configure BookStack's `.env`

Edit the BookStack environment file to reflect the new domain:

```
sudo nano /var/www/bookstack/.env
```

Update this line:

```
APP_URL=https://wiki.example.com
```

Then restart the web server:

```
sudo systemctl restart apache2
```

## ? Set up HTTPS with Let's Encrypt (Optional but Recommended)

Install Certbot:

```
sudo apt install certbot python3-certbot-apache
sudo certbot --apache -d wiki.example.com
```

It will automatically get and install the SSL certificate.

## ? Step-by-Step Mail Setup

### 1. Edit the `.env` File

Open the BookStack environment config:

```
sudo nano /var/www/bookstack/.env
```

Look for or add the following section:

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.example.com
MAIL_PORT=587
MAIL_USERNAME=your@email.com
MAIL_PASSWORD=your_password
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=your@email.com
MAIL_FROM_NAME="BookStack"
```

Replace the values depending on your email provider.

## 2. Clear the config Cache (optional but recommended)

```
cd /var/www/bookstack
php artisan config:clear
```

# PairDrop Installation via Proxmox VE Helper-Scripts

This guide covers installing **PairDrop** in an LXC container using the community-built [Proxmox VE Helper-Scripts](#). These scripts make deploying apps like PairDrop fast, standardized, and repeatable.

## ? Step-by-Step Instructions

- To create a new Proxmox VE PairDrop LXC, run the command below in the Proxmox VE Shell.

```
bash -c "$(curl -fsSL https://raw.githubusercontent.com/community-scripts/ProxmoxVE/main/ct/pairdrop.sh)"
```

This command runs a script that launches an interactive menu for installing **PairDrop** in a container.

## Follow the Installation Prompts

The script will guide you through a series of options. Typical options include:

- Set **container ID** (e.g. `904`)
- Choose **hostname** (e.g. `pairdrop`)
- Select a **static IP address** (e.g. `172.30.0.x`)
- Confirm **bridge interface** (e.g. `vmbr0`)
- Choose container **storage** (e.g. `local-lvm`)
- Set **root password**
- Enable optional features (SSH, nesting, etc.)

## Wait for the Script to Finish

Once confirmed, the script will:

- Download the latest **Debian 12** template
- Create the LXC container
- Install **Docker** inside the container
- Deploy the **PairDrop** Docker container

## Access the PairDrop Web Interface

Once complete, open your browser and go to:

<http://<your-static-ip>:3000>

# Paperless-ngx Installation on Debian 12 (Using Official Script)

**Paperless-ngx** is a document management system for indexing, OCRing, and organizing scanned documents and PDFs. This guide walks you through setting it up on **Debian 12** using the official installation script via Docker Compose.

Paperless provides an interactive installation script to set up a Docker Compose installation. The script asks for a couple of configuration options and will then create the necessary configuration files, pull the Docker image, start Paperless-ngx, and create your superuser account.

## ? 1. Install Docker & Compose (Debian12)

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg lsb-release
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

# Install Docker $ Compose plugin
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin

#Enable and Start Docker
sudo systemctl enable --now docker
sudo service docker start
```

## ? 2. Run the Official Paperless-ngx Installation Script

Download and run the setup script:

```
bash -c "$(curl --location --silent --show-error https://raw.githubusercontent.com/paperless-ngx/paperless-ngx/main/install-paperless-ngx.sh)"
```




Follow the prompts.

## ? 4. Access the Interface

Once complete, go to:

```
http://<your-server-ip>:8000
```

## ? Reference

-  Official GitHub Repository:  
<https://github.com/paperless-ngx/paperless-ngx>
-  Docker Installation Guide:  
<https://docs.docker.com/engine/install/debian/>
-  Paperless-ngx Documentation:  
<https://docs.paperless-ngx.com/>

# Snipe-IT Installation on Ubuntu 24.04

This guide explains how to install **Snipe-IT**, a self-hosted IT asset management system, on **Ubuntu 24.04** using **Nginx**, **MariaDB**, and **PHP 8.3**.

Ref. <https://snipe-it.readme.io/docs/installation>

## ? 1. Update the System

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

## ? 2. Install Required Packages

Install Nginx, PHP, MariaDB, and all required PHP extensions:

```
sudo apt install nginx mariadb-server php-bcmath php-common php-ctype php-curl php-fileinfo  
php-fpm php-gd php-iconv php-intl php-mbstring php-mysql php-soap php-xml php-xsl php-zip git  
-y
```

## ?? 3. Install Composer

```
sudo apt install composer -y
```

## ?? 4. Create the Snipe-IT Database

```
sudo mysql
```

Inside the MySQL shell:

```
CREATE DATABASE snipeit;  
GRANT ALL ON snipeit.* TO 'snipeit'@'localhost' IDENTIFIED BY 'yourStrongPassword';  
FLUSH PRIVILEGES;  
EXIT;
```

## ? 5. Download Snipe-IT

```
cd /var/www/html
sudo git clone https://github.com/snipe/snipe-it
cd snipe-it
```

## ?? 6. Create and Configure `.env` File

```
sudo cp .env.example .env
sudo nano .env
```

Edit the following variables:

```
APP_URL=http://your-server-ip
DB_DATABASE=snipeit
DB_USERNAME=snipeit
DB_PASSWORD=yourStrongPassword
```

## ? 7. Set Permissions

```
sudo chown -R www-data: /var/www/html/snipe-it
sudo chmod -R 755 /var/www/html/snipe-it
```

## ? 8. Install Dependencies with Composer

```
sudo composer update --no-plugins --no-scripts
sudo composer install --no-dev --prefer-source --no-plugins --no-scripts
```

## ? 9. Generate App Key

```
sudo php artisan key:generate
```

Make sure you keep a copy of your `APP_KEY` in a safe place, for example, a secure password manager. This key is required to decrypt any encrypted fields in the database.

## ? 10. Check PHP FPM Version

```
sudo systemctl list-units --type=service | grep php
```

Ensure PHP 8.3 (or the current version) is installed and running.

## ?? 11. Enable PHP-FPM

```
sudo systemctl start php8.3-fpm
sudo systemctl enable php8.3-fpm
```

## ? 12. Create Nginx Configuration for Snipe-IT

```
sudo nano /etc/nginx/conf.d/snipeit.conf
```

Paste the following (adjust IP and PHP version if needed):

```
server {
    listen 80;
    server_name 10.0.0.7;
    root /var/www/html/snipe-it/public;

    index index.php;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~ /\.php$ {
include fastcgi.conf;
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.3-fpm.sock;
fastcgi_split_path_info ^(.+\.(php|php5|php7|php8|php9|phpN|cgi|cgi-bin))(/.*)$;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

## ? 13. Enable the Site

```
sudo ln -s /etc/nginx/conf.d/snipeit.conf /etc/nginx/sites-enabled/snipeit.conf
```

## ?? 14. Update Nginx Config

Edit main config:

```
sudo nano /etc/nginx/nginx.conf
```

Add the following line **inside the** `http` **block:**

```
server_names_hash_bucket_size 64;
```

## ? 15. Restart Nginx

```
sudo systemctl restart nginx
```

## ? 16. Access Snipe-IT Web Interface

Open your browser and go to:

```
http://your-server-ip
```

Complete the web-based setup to finish the installation.