

# Applications

**Tips and Tricks** about some of my Docker Applications.

- [Deployment Checklist](#)
- [Apache Guacamole](#)
  - [Docker Installation on ARM64](#)
- [Bookstack](#)
  - [Changing the Base URL](#)
- [Paperless-ngx](#)
  - [Backup & Restore](#)
- [ntfy](#)
  - [📁 Low Space](#)
  - [🔑 SSH Access](#)
  - [📁 Useful Commands](#)
- [Vikunja](#)
  - [Quick Add Magic](#)

# Deployment Checklist

*Here is a checklist to make sure to not forget anything when deploying a new container on my Docker.*

---

- ☐ Create a CNAME record on Cloudflare
- ☐ Create a CNAME record on PiHole
- ☐ Write the `compose.yaml` and `.env` files
- ☐ Add the container to the `Caddyfile`
- ☐ Add the container port to Portall
- ☐ Add the application to homepage
- ☐ Add the container to Uptime Kuma
- ☐ Add the container to Guacamole

# Apache Guacamole

**Apache Guacamole** is a clientless remote desktop gateway. It supports standard protocols like VNC, RDP, and SSH.

# Docker Installation on ARM64

*Apache Guacamole is a powerful tool for managing remote connections, and installing it on a Raspberry Pi 4 with Docker allows for easy, remote access from almost anywhere*

---

## Prerequisites

1. Raspberry Pi 4
2. Docker installed

## Docker Compose Setup

For this installation, we will use a `compose.yaml` file with Docker Compose to manage the Guacamole installation.

I'm using the [flcontainers/guacamole](#) Docker image, which is compatible with the ARM64 architecture, unlike the official image.

Here's what the `compose.yaml` file should look like:

```
services:
  guacamole_app:
    container_name: guacamole_app
    image: flcontainers/guacamole:latest
    restart: unless-stopped
    ports:
      - 8094:8080
    volumes:
      - guacamole_app-config:/config
      - /etc/localtime:/etc/localtime:ro
    environment:
      TZ: 'Europe/Brussels'
    healthcheck:
```

```
test: curl -f -k http://127.0.0.1:8080/ || exit 1
interval: 15s
timeout: 10s
retries: 5
```

volumes:

```
guacamole_app-config:
  name: guacamole_app-config
```

# Installation

Once you've created the `compose.yaml` file, navigate to the directory where it's stored and run the following command:

```
docker compose up -d
```

This command will pull the Guacamole image and install it on your Raspberry Pi.

# Using Apache Guacamole

After the installation, you can access the Guacamole web interface by visiting `http://<YOUR_RPI_IP>:8094`

**Username:** guacadmin  
**Password:** guacadmin

With this setup, you now have Apache Guacamole running on your Raspberry Pi 4, allowing easy remote desktop access and management.

Happy me! ☺

# Bookstack

**BookStack** is a simple, open-source, self-hosted, easy-to-use platform for organising and storing information.

# Changing the Base URL

*Sometimes, you need to change the base URL of Bookstack, for example, when you switch from the localhost address to the internet exposed address.*

---

## Docker Compose

In your `compose.yaml`, modify the following environment variable:

```
APP_URL=<new_url>
```

## Bookstack Container

Open a terminal and type:

```
docker exec -it <bookstack_container> php /app/www/artisan bookstack:update-url <old_url> <new_url>
```

## Clear Cache

Open a terminal and type:

```
docker exec -it <bookstack_container> php /app/www/artisan cache:clear
```

# Paperless-ngx

**Paperless-ngx** is a document management system that transforms your physical documents into a searchable online archive.



# Backup & Restore

*Here is the procedure to backup and restore the Paperless-NGX application and all of its data.*

---

## Backup

On a terminal, enter the following command:

```
docker compose exec -T <paperless_webserver> document_exporter -z ../export
```

Where:

- `-T` is used to suppress "The input device is not a TTY" error ;
- `-z` is used to zip the export ;
- `../export` is used because this path inside the container is automatically mounted on your host on the folder export.

## Restore

**You'll need to unzip the previous export!**

On a terminal, enter the following command:

```
docker compose exec -T <paperless_webserver> document_importer ../export/<unzipped_directory>/
```

Where:

- `-T` is used to suppress "The device is not a TTY" error ;
- `../export/<unzipped_directory>/` is the path to your previous backup unzipped.



**ntfy** lets you send push notifications to your phone or desktop via scripts from any computer, using simple HTTP PUT or POST requests.

ntfy

# ? Low Space

## Script

```
#!/bin/bash

mingigs=20
avail=$(df | awk '$6 == "/" && $4 < '$mingigs' * 1024*1024 { print $4/1024/1024 }')

if [ -n "$avail" ]; then
  curl \
    -H "Title: Low Disk Space" \
    -H "Priority: urgent" \
    -H "Tags: warning,cd" \
    -d "Hello TTBB! Only $avail GB available on the root disk. Better clean that up." \
    https://notify.boreux.work/raspberrypi?auth=<TOKEN>
fi
```

## Cron

```
0 * * * * root /bin/bash /root/ntfy/low-disk-space.sh
```

ntfy

# ?? SSH Access

## Script

```
#!/bin/bash
if [ "${PAM_TYPE}" = "open_session" ]; then
  curl \
    -H "Title: SSH Login" \
    -H "Priority: urgent" \
    -H "Tags: warning,door" \
    -d "Hello TTBB! There was an SSH login with user ${PAM_USER} from ${PAM_RHOST} host" \
    https://notify.boreux.work/pihole?auth=<TOKEN>
fi
```

`/etc/pam.d/sshd`

At the end of the file, add the following line:

```
session optional pam_exec.so /root/ntfy/ssh-login.sh
```

ntfy

# ? Useful Commands

## Creating an Administrator

```
docker exec -it <NTFY_CONTAINER_NAME> ntfy user add --role=admin <USERNAME>
```

# Vikunja

The to-do app to organize your life

# Quick Add Magic

## Quick Add Magix Syntax Cheatsheet

Labels - \*

\*"Label with spaces"

Project - +

+"Project with spaces"

## Due Date and Time

The date format is: MM/DD/YYYY

02/17/2021 at 17:00

## Assignee

@ttbb

## Examples

- dishwasher +HOUSE \*kitchen 04/05/2025 at 09:00 @ttbb
- washing machine +HOUSE \*bathroom 04/05/2025 at 09:00 @ttbb

- clean surfaces +HOUSE \*kitchen 04/05/2025 at 09:30 @ttbb
- trash +HOUSE \*kitchen 04/05/2025 at 09:45 @ttbb
- clean surfaces +HOUSE \*living 04/05/2025 at 10:00 @ttbb
- clean toilet +HOUSE \*toilets 04/05/2025 at 10:30 @ttbb
- clean surfaces +HOUSE \*bathroom 04/05/2025 at 11:00 @ttbb
- vacuum +HOUSE \*house 04/05/2025 at 11:30 @ttbb
- wash the floors +HOUSE \*house 04/05/2025 at 12:00 @ttbb
- clean shower +HOUSE \*bathroom 04/05/2025 at 12:30 @ttbb