How to migrate from HubSpot to Mautic

This guide walks you through moving your marketing from HubSpot to a self-hosted Mautic instance: Contacts, unsubscribes, Campaigns and Forms. It follows the order we used in a real migration, and it calls out the mistakes we made so you can avoid them.

By the end, you will have:

- Mautic running on your own server, with backups
- An email delivery provider connected and authenticated
- All your HubSpot Contacts imported, with unsubscribes and bounces still respected
- Your HubSpot workflows rebuilt as Mautic Campaigns
- New signups flowing into Mautic, and HubSpot safely wound down

Prerequisites

Before starting, you need to have:

- **Admin access to HubSpot**, including the ability to export contacts and use the HubSpot API, while your subscription is still active.
- **A server you control**. A small VPS with at least 2 vCPU, 2 GB RAM and 60 GB of storage is enough for most small and mid-size lists.
- **A domain or subdomain for Mautic**, such as mautic.example.com, and access to its DNS settings.
- **An account with an email delivery provider**, such as Postmark, Amazon SES or Mailgun.
- **Basic comfort with the command line**, Docker and DNS editing.

Tip: Do not cancel HubSpot yet. Some data, including marketing email performance history, is lost when you downgrade HubSpot. The "Wind down

HubSpot" section explains how to do it safely.

Cost expectations

A small VPS usually costs $10 to $25 per month, and email delivery often starts around $15 per month. That is typically far less than HubSpot Marketing, but you take on running the server yourself.

Migration steps

1. Set up the server

1. Create a VPS with a current Ubuntu LTS release (22.04 or 24.04). Any provider works, such as Hetzner, DigitalOcean or Linode.
2. Point an **A record** for your Mautic subdomain at the server's IP address.
3. Add swap space. Mautic, MySQL and Docker together can run out of memory on a 2 GB server without it:

```
fallocate -l 2G /swapfile && chmod 600 /swapfile && mkswap /sw
apfile && swapon /swapfileecho '/swapfile none swap sw 0 0' >>
 /etc/fstab
```

4. Install Docker and Docker Compose using the official Docker instructions for your operating system.

2. Install Mautic with Docker

1. Use the official mautic/mautic Docker images with a Compose file,

and install a current Mautic release.
2. **Declare named volumes** (or bind mounts) for everything that must survive a rebuild or upgrade:
    - the media images folder (uploaded logos and email images)
    - the media files folder (uploaded assets)
    - the config folder (contains local.php, your instance settings)
    - the MySQL data directory

```
volumes:  mautic_media_images: {}  mautic_media_files: {
}  mautic_config: {}  mautic_db: {}
```

3. Map these volumes into the containers. Then run the Mautic installer (web wizard or bin/console mautic:install), create your admin User, and connect it to the database container.

Warning

This is the most common setup mistake. If uploads and config live in anonymous Docker volumes, they are wiped the next time you rebuild or upgrade the containers.

3. Configure the domain and SSL

1. Put a reverse proxy such as nginx in front of Mautic and proxy your Mautic subdomain to the Mautic container.
2. Raise the upload limit, for example:

```
client_max_body_size 64M;
```

Without it, image and file uploads can fail with no clear error.
3. Get a free SSL certificate from Let's Encrypt (for example with certbot) and redirect all traffic to HTTPS.

4. Connect an email delivery provider

Mautic does not deliver email by itself. It hands each message to an email provider.

1. Choose a provider (Postmark, Amazon SES, Mailgun or similar) and create a sending stream or domain for marketing email.
2. Add the mailer settings in Mautic (**Settings > Configuration >** Email Settings, or mailer_dsn in local.php).
3. Authenticate your sending domain in DNS. Set up all three of these, because they are required for good deliverability:
    - **SPF**, authorizing your provider to send for your domain
    - **DKIM**, using the keys your provider gives you
    - DMARC, starting with p=none and tightening later
4. Send a test email and confirm it arrives in an inbox, not spam.

Tip: Most providers bill overage with no hard cap. Consider a small scheduled script that checks your month-to-date send count through the provider's API and pauses Mautic's send job if it passes a limit you set. That way a misconfigured Campaign cannot run up a large bill.

Warning

Many providers pause accounts with high bounce rates. The "Re-engage older Contacts" section explains how to email older Contacts without getting your account paused.

5. Decide how to handle website tracking

Mautic's tracking script (mtc.js) creates an **anonymous Contact for every unknown visitor**. On a busy website, this can add thousands of junk Contacts a day and slow down your database.

- Only add the tracking script to sites where you really need visitor tracking. Use a web analytics tool for general traffic stats.
- If you don't need anonymous tracking, you can block the tracking endpoints at your proxy.
- Schedule a regular cleanup of anonymous Contacts (Mautic's mautic:maintenance:cleanup command, or your own script).

Warning

Imported Contacts need an identified date set, or Mautic treats them as anonymous and a cleanup job can delete them. The "Import Contacts into Mautic" section shows how to set it.

6. Create API credentials

If you connect a website, an app or scripts to Mautic, give each one its own API credentials instead of a person's login.

1. In Mautic, go to **Settings > Configuration >** API Settings and enable the API.
2. Go to **Settings > API Credentials** and create an OAuth2 client for each integration.
3. Store the client ID and secret securely, never in a shared document or code repository.

7. Export everything from HubSpot

Do this while your HubSpot subscription is still fully active. Use HubSpot's exports or its API to collect:

1. **Contacts with all properties**, including:
    - email, name, company, create date and owner
    - last email open date and last email click date (for engagement)
    - subscription and opt-out status, and hard bounce reasons
    - marketing contact status
2. **Workflow memberships**: for each contact, which workflows they are enrolled in, are in the middle of, or have completed. You need this to avoid sending people emails they already received.
3. **Suppression data**: unsubscribes and hard bounces.
4. **Form submissions and notes** you want to keep.
5. **Marketing email performance history** (opens and clicks per email). This is lost when you downgrade HubSpot, so archive it now.

Tip: In HubSpot, 'non-marketing contact' usually just means you chose not to pay to market to them. It is not the same as an unsubscribe. Those contacts can still be emailed unless they opted out or bounced.

8. Import Contacts into Mautic

You can use Mautic's built-in CSV import (Contacts > Import) for simple lists, or a script that uses the Mautic API for more control. Either way, follow these rules:

1. **Test with about 10 Contacts first.** Check them in the Mautic interface, then run the full import.
2. **Set the identified date** on every imported Contact (for example to their HubSpot create date). Otherwise Mautic treats them as anonymous.
3. **Set the last active date** from their most recent email open or click, falling back to the create date. This keeps engagement-based Segments

accurate.

4. **Import unsubscribes and hard bounces as Do Not Contact** for the Email Channel. Mautic then skips them automatically on every send.
5. **Map HubSpot owners to Mautic Users** so Contact ownership carries over.
6. **Do not add imported Contacts to welcome or onboarding Campaigns.** They already received those emails in HubSpot.
7. Use the workflow membership export to decide who belongs in which Campaign. Do not guess from email domains or other fields.
8. After the import, update Segments and Campaigns:

```
bin/console mautic:segments:updatebin/console mautic:campaigns:updatebin/console cache:clear
```

9. Rebuild Campaigns and Forms

1. **Rebuild each active HubSpot workflow as a Mautic Campaign.** Use Segments as Campaign sources and Campaign conditions in place of HubSpot enrollment and branching rules.
2. **Activate the Campaigns before you switch new signups over**, so no one falls through the gap.
3. **Move your Forms.** Either use Mautic Forms directly, or have your website or app submit to a Mautic Form or the Mautic API when someone signs up. Going through a Form or the API (not writing to the database) makes sure Campaigns trigger.
4. **Send account changes too.** If your own app knows things like whether someone is a paying customer, submit those updates to Mautic as well. Your Segments can then move people in and out of Campaigns automatically, for example removing paying customers from prospect emails.

10. Switch new signups to Mautic without duplicate emails

This is the moment of cutover. The goal is that every new signup gets exactly one welcome sequence.

1. Pick a cutover date and time.
2. At that time, point your website forms and app signups at Mautic.
3. In HubSpot, change each active workflow's enrollment criteria to **include only contacts created before the cutover time**. Existing people finish their HubSpot sequences, and new people only enter Mautic Campaigns.
4. Sign up with a test email address and confirm you receive the Mautic welcome email, and nothing from HubSpot.
5. Watch new signups for a few days to confirm they are arriving in Mautic and entering the right Campaigns.

11. Re-engage older Contacts

Contacts who haven't heard from you in a long time are a deliverability risk. Emailing them all at once can cause a sudden rise in bounces and complaints and get your sending paused. Follow these steps to re-engage them safely:

1. Send in **batches of about 1,500 to 2,000 Contacts**, oldest signups first.
2. **Wait about two weeks between batches** and continue only once the bounce rate has settled.
3. Exclude Contacts who were recently active.
4. Use an honest "It's been a while. Here's what's new" message.
5. Include a clear, friendly unsubscribe link. Unsubscribes are fine; spam complaints are what damage your sending reputation.

Mautic marks hard bounces as **Do Not Contact** automatically, so each batch cleans your list a little more.

12. Wind down HubSpot

1. **Turn off auto-renewal early.** HubSpot usually doesn't let you cancel mid-term or give refunds, so make sure it can't renew for another term.
2. **Let in-progress workflows finish.** Check your workflow membership export for anyone still in the middle of a sequence and give them time to complete.
3. **Archive what a downgrade removes**: marketing email performance history, reports and any files or pages you built in HubSpot. Store the archive somewhere permanent.
4. Once workflows have finished and data is archived, downgrade or cancel.

13. Schedule the cron jobs

Mautic relies on scheduled jobs. A typical setup looks like this:

```
*/5  * * * *  bin/console mautic:segments:update*/5  * * * *  bin/con
sole mautic:campaigns:update*/5  * * * *  bin/console mautic:campaign
s:trigger*/5  * * * *  bin/console messenger:consume email --time-lim
it=2400    3 * * *  bin/console mautic:maintenance:cleanup --days-
old=3650    2 * * *  database and media backup to offsite storage
```

Adjust the timing to your volume. The messenger:consume email job is the one that actually delivers queued email.

Tip: Add a simple health check that alerts you if the server or Mautic stops responding, and test that your backups can actually be restored.

Tips for running Mautic day to day

- **Check delivery at your email provider.** An API 'success' response from Mautic means the email was queued, not delivered. Your

provider's activity log shows what was delivered or bounced.
- **Don't delete Emails that have already been sent.** Deleting an Email removes its send statistics.
- **Clear the cache if the interface shows errors** after API changes: run bin/console cache:clear and make sure the cache folder is owned by the web server user.
- **Follow email law.** Every marketing email needs a working unsubscribe link and your physical mailing address.
- **Keep tags and Segments organized.** A consistent naming pattern (for example Type - Subtype - Value for tags, and numbered Segment names) keeps the account easy to understand as it grows.

Summary of the order

1. Server, swap and Docker
2. Mautic with named volumes
3. Domain, proxy and SSL
4. Email provider and SPF, DKIM, DMARC
5. Tracking decisions
6. API credentials
7. Export from HubSpot
8. Test import, then full import
9. Rebuild Campaigns and Forms, with HubSpot still running
10. Cut over new signups
11. Re-engage older Contacts in batches
12. Wind down HubSpot
13. Cron jobs, monitoring and backups

Take it one step at a time and check each step before moving on.

Online URL:
https://kb.mautic.org/article/how-to-migrate-from-hubspot-to-mautic.html