hosting mautic on your own open source paas

SETTING UP MAUTIC IN COOLIFY

WHO AM I

Head of Data & Analytics
@VML Germany

Freelance Jack of all Trades Marketing
@moorwald

Team Lead Marketing Team
@Mautic

BEHIND THE CASE

Automated CSRD Certification
SMEs get their ESG certificate through online questionnaire instead of weeks of consulting

Helping with Compliance Pressure in B2B Supply Chain
Suppliers to large corporations increasingly need ESG proof, same applies for public tenders and bank communication

Certificate in <24h
Certificate is immediately ready for download within minutes

Mautic as central marketing hub
Newsletter campaigns, landing pages, marketing automation, and future whitepaper downloads run on Coolify

Transactional Emails via API
Certificate downloads, registration confirmations, and password resets are sent directly from the application through Mautic API

WHAT IT IS

Open-source, self-hostable Platform as a Service (PaaS)

Essentially an alternative to Heroku, Netlify, and Vercel

You run it on your own infrastructure, basically on any machine you can ssh into

Personally I host mine (and the client's one) on Hetzner VPS for < 10€/Month

PROBLEM IT SOLVES

Eliminates expensive cloud PaaS bills

Eliminates vendor lock-in

Provides the same deployment convenience (as those proprietary cloud services)

Easily deploy apps, databases, and 280+ services to any server (or servers)

KEY FEATURES

Git integration (GitHub, GitLab, Bitbucket)

Automatic SSL certificates

One-click database/service deployments

Works on any hardware - VPS, bare metal, Raspberry Pi, or cloud instances

No vendor lock-in - all configurations saved to your server

AN OVERLY SHORT OVERVIEW OF COOLIFY

WHY I LIKE USING IT

Notifications
Email + Pushover when things break or backups succeed

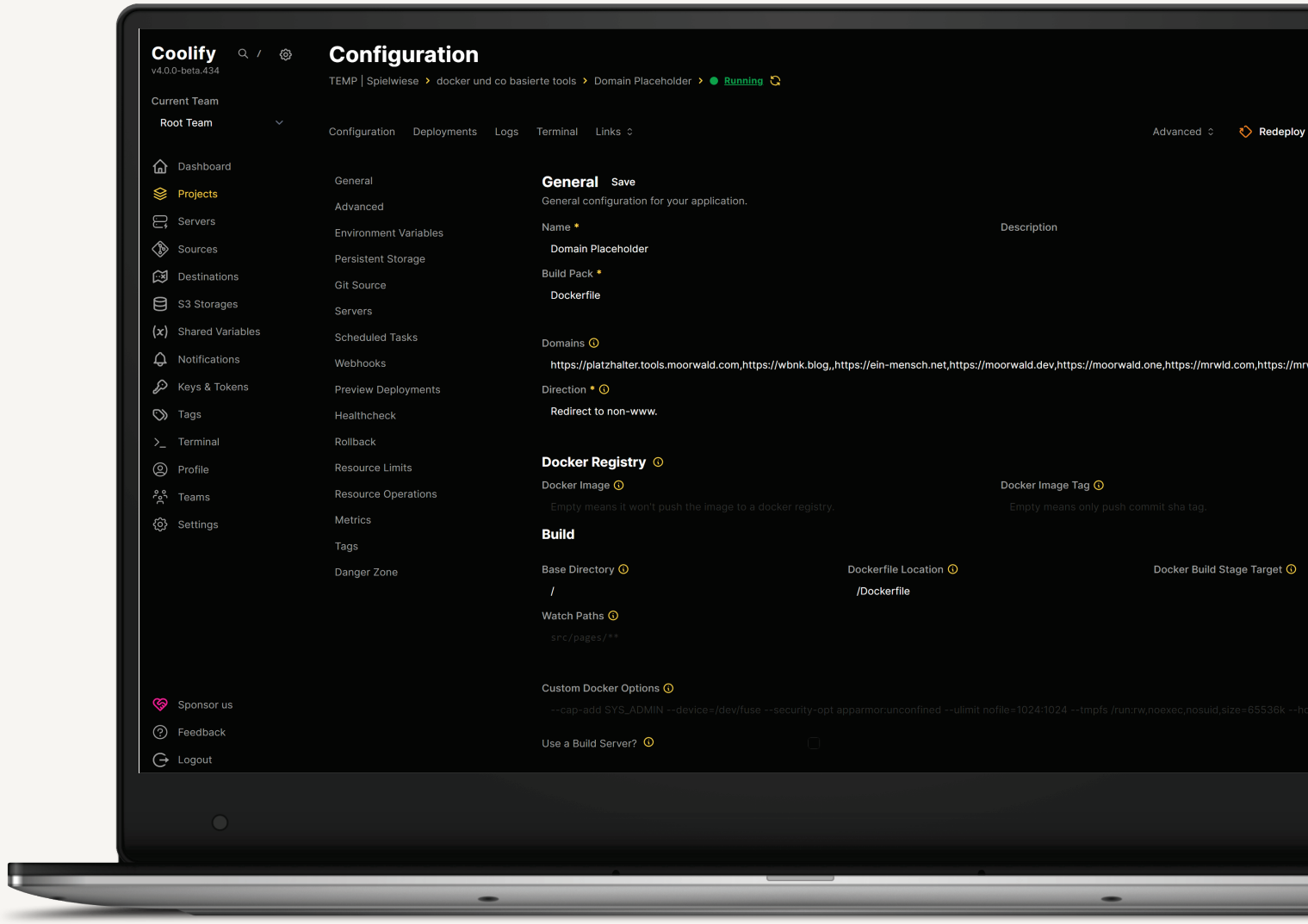
Wildcard setup
*.tools subdomain + auto Let's Encrypt = deploy anything in minutes

15+ services running
Mautic, Matomo, custom tools, monitoring, DBs, DNS - mail server next

Variables & secrets
Per-project ENVs, shared secrets, team access, centralized token management

S3 backups
Set once, forget about it

Scaling
Quickly & easily add additional servers for additional services/projects, if the need arises



AVOIDING THE ONE CLICK COMPLEXITY

Or how I recovered from shoting myself in the foot.

QUEST FOR MORDOR

Two days of trying to get this to work
Debugging Rabbitmq & Mautic

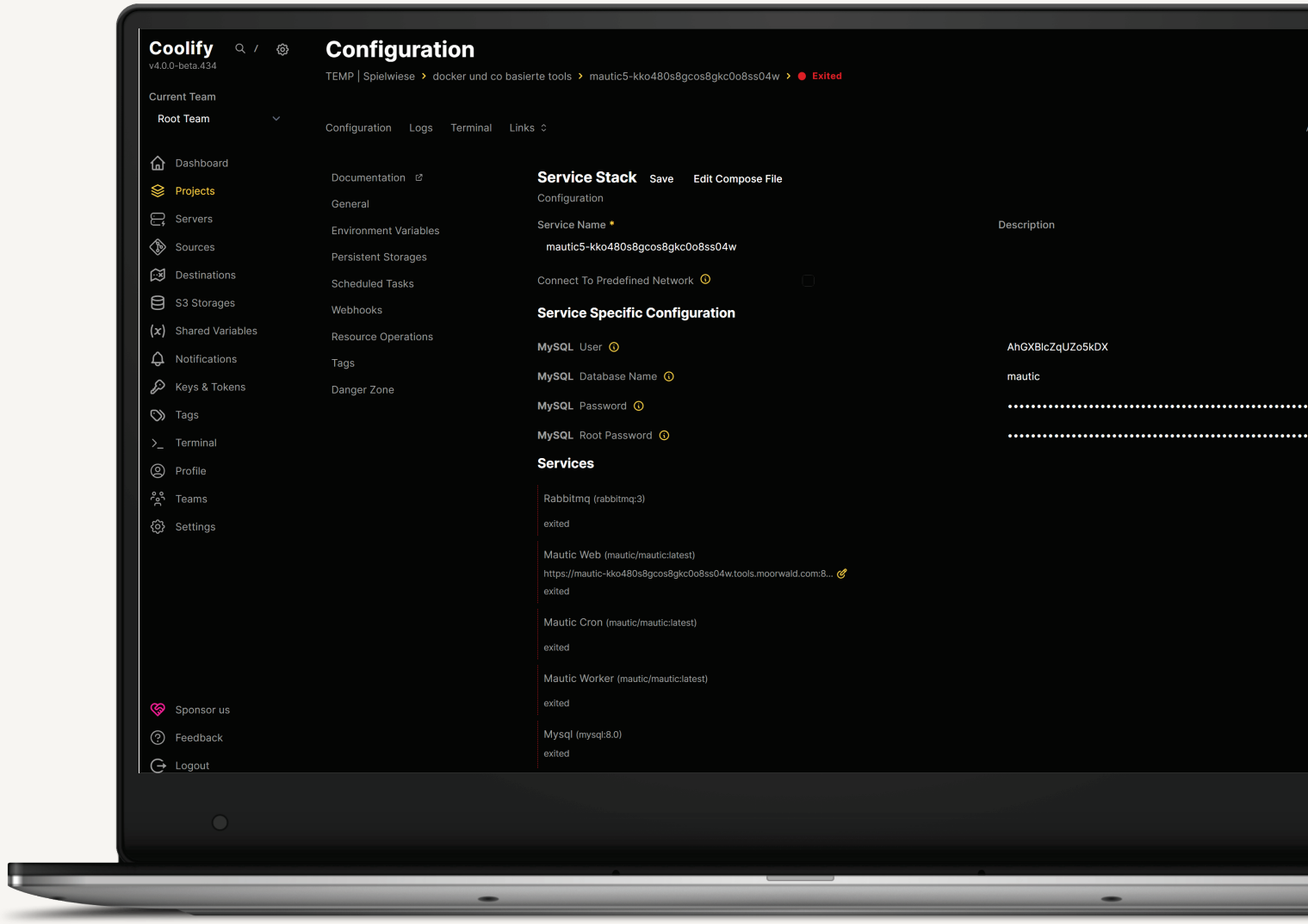
Throwing the towel
Finally gave up, decided that I don‘t need queueing and that things must be easier

Using a Compose File
Decided on a minimal docker compose setup of Mautic and MySQL*

Cronjobs via UI
Coolify calls them „Scheduled Task“

Fit to scale
My client does not have many thousands of contacts (hopefully) yet. We‘ll cross that bridge, when we get to it

Result
Within 30 minutes I was running smoothly and had sent my first test mail via Mautic



* There‘s a slightly more robust solution at the end

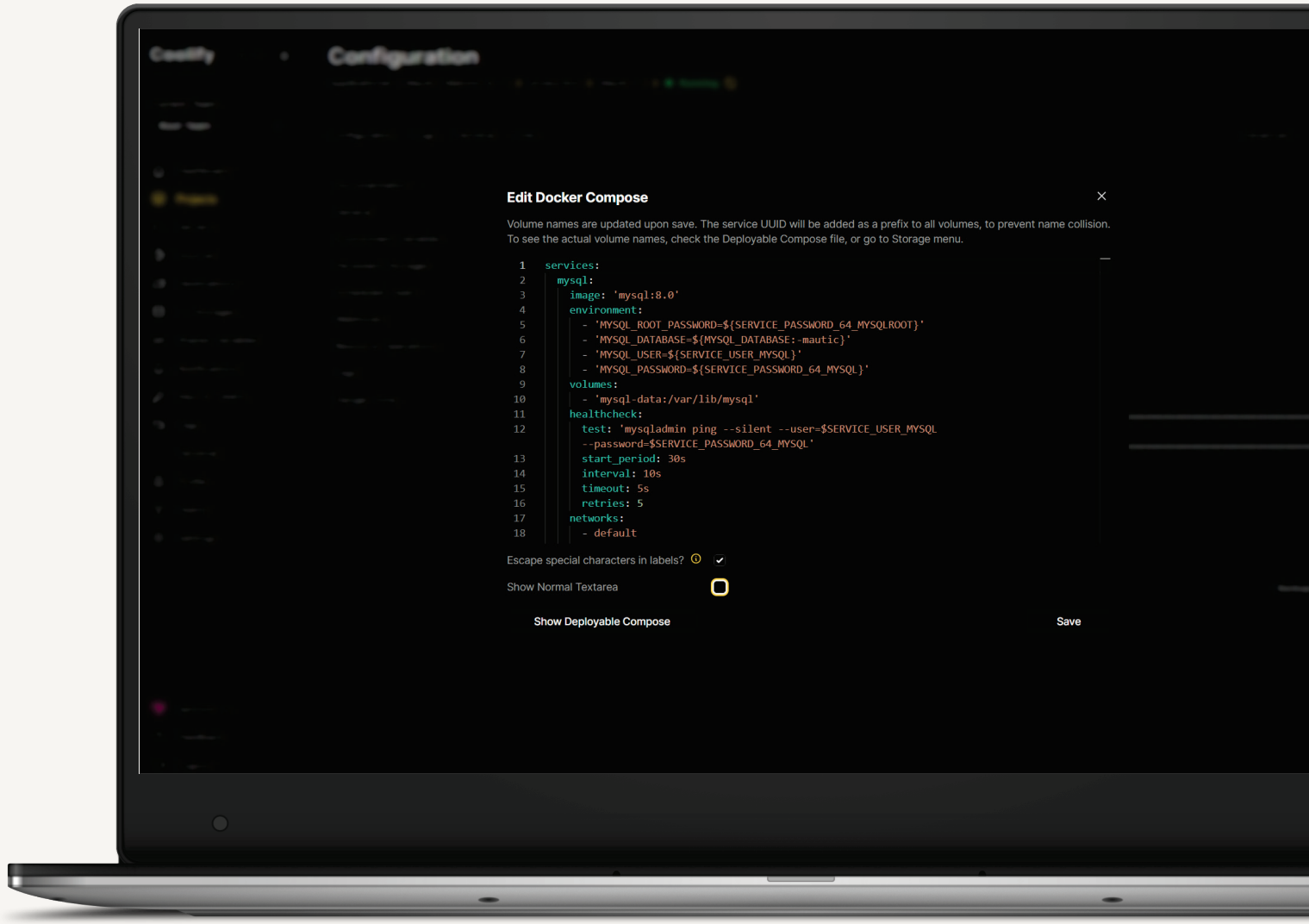
A MORE EASY SOLUTION

```
services:
  mysql:
    image: 'mysql:8.0'
    environment:
      [[ A FEW ENV VARS]]
    volumes:
      - 'mysql-data:/var/lib/mysql'
    healthcheck:
      [[ healthcheck definition ]]
    networks:
      - default
      - analytics-network

  mautic_web:
    image: 'mautic/mautic:latest'
    volumes:
      [[ A FEW VOLUMES ]]
    environment:
      [[ A FEW ENV VARS]]
    healthcheck:
      [[ healthcheck definition ]]
    labels:
      [[ A FEW LABELS ]]
```

```
volumes:
  mysql-data: null
  mautic-dataconfig: null
  mautic-datalogs: null
  mautic-datamediafiles: null
  mautic-datamediaimages: null
  mautic-dataplugins: null
  mautic-datathemes: null
  mautic-datavendor: null
  mautic-databin: null

networks:
  analytics-network:
    name: analytics-network
    external: false
```



```
mautic_web:
    image: 'mautic/mautic:latest'

    volumes:
      - 'mautic-dataconfig:/var/www/html/config'
      - 'mautic-datalogs:/var/www/html/var/logs'
      - 'mautic-datamediafiles:/var/www/html/docroot/media/files'
      - 'mautic-datamediaimages:/var/www/html/docroot/media/images'
      - 'mautic-dataplugins:/var/www/html/docroot/plugins'
      - 'mautic-datathemes:/var/www/html/docroot/themes'
      - 'mautic-datavendor:/var/www/html/vendor'
      - 'mautic-databin:/var/www/html/bin'

    environment:
      - SERVICE_FQDN_MAUTIC_80
      - 'DOCKER_MAUTIC_LOAD_TEST_DATA=${MAUTIC_LOAD_TEST_DATA:-false}'
      - 'DOCKER_MAUTIC_RUN_MIGRATIONS=${MAUTIC_RUN_MIGRATIONS:-false}'
      - 'MAUTIC_DB_HOST=${MYSQL_HOST:-mysql}'
      - 'MAUTIC_DB_PORT=${MYSQL_PORT:-3306}'
      - 'MAUTIC_DB_DATABASE=${MYSQL_DATABASE:-mautic}'
      - 'MAUTIC_DB_USER=${SERVICE_USER_MYSQL}'
      - 'MAUTIC_DB_PASSWORD=${SERVICE_PASSWORD_64_MYSQL}'
      - 'MAUTIC_MESSENGER_DSN_EMAIL=sync://'
      - 'MAUTIC_MESSENGER_DSN_HIT=sync://'

    healthcheck:
      test:
        - CMD
        - curl
        - '-f'
        - 'http://localhost'
      interval: 15s
      timeout: 10s
      retries: 15

    labels:
      - traefik.docker.network=r430kw5wofggs8c0oh4tc8so
```

```
mysql:
    image: 'mysql:8.0'
    environment:
      - 'MYSQL_ROOT_PASSWORD=${SERVICE_PASSWORD_64_MYSQLROOT}'
      - 'MYSQL_DATABASE=${MYSQL_DATABASE:-mautic}'
      - 'MYSQL_USER=${SERVICE_USER_MYSQL}'
      - 'MYSQL_PASSWORD=${SERVICE_PASSWORD_64_MYSQL}'

    volumes:
      - 'mysql-data:/var/lib/mysql'

    healthcheck:
      test: 'mysqladmin ping --silent --user=$SERVICE_USER_MYSQL --
password=$SERVICE_PASSWORD_64_MYSQL'
      start_period: 30s
      interval: 10s
      timeout: 5s
      retries: 5

    networks:
      - default
      - analytics-network

volumes:
  mysql-data: null
  mautic-dataconfig: null
  mautic-datalogs: null
  mautic-datamediafiles: null
  mautic-datamediaimages: null
  mautic-dataplugins: null
  mautic-datathemes: null
  mautic-datavendor: null
  mautic-databin: null

networks:
  analytics-network:
    name: analytics-network
    external: false
```

THE COMPOSE FILE IN DETAIL: MAUTIC & MYSQL

BETTER THAN CRON?

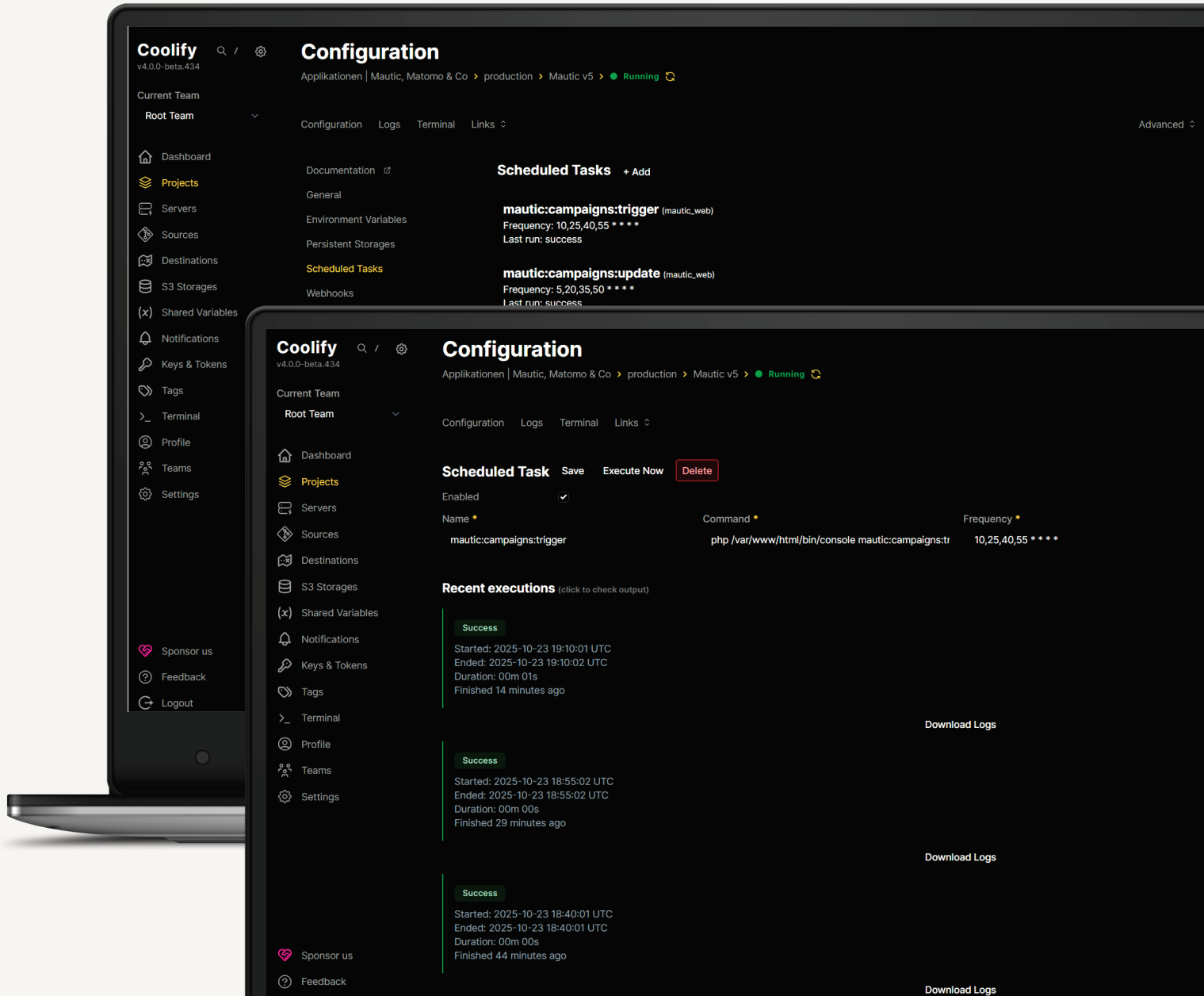
Easy Web Interface Management
Scheduled tasks created and edited directly in the browser. No SSH access or manual crontab editing

Execution History Visible
Past task runs are automatically logged. Shows status, timestamp and duration of recent executions

Automatic Failure Notifications
Failed tasks trigger notifications. If a Mautic command fails, you know immediately

Download Log Files
Output of each execution is stored persistently. Logs can be downloaded directly from the interface as files

Container Integration Without Workarounds
Tasks run directly in the Mautic container. No separate cron container configuration or volume mounting for cron files



HOW TO DEPLOY IN COOLIFY

Two viable options...

... to start without the One Click* Installer

* A viable option if you need queueing

OPTION 1: HOW IT WORKS

Copy
Copy your docker-compose.yml to Coolify UI

Configure
Configure Environment Variables, URL, Service Name, Description

Configure Scheduled Tasks, potentially Webhooks

Deploy
Click deploy - Coolify pulls images and starts containers

Update
Updates require manual file edits and redeployment

WHEN WOULD I USE THAT AGAIN?

Testing and experimentation
Quick iterations without Git overhead or setting up a GitHub App
(to me the most painful part anytime I need to do that)

Single instance deployments
Just one Mautic setup to manage

Simple configurations
No need for version history

Learning Coolify
Easiest way to get started

One-person show
You're the only one touching this setup

DIRECTLY USE A COMPOSE FILE

OPTION 2: HOW IT WORKS

Push
Push your docker-compose.yml to a GitHub repository

Connect
Connect Coolify via GitHub App integration

Deploy
After the initial deployment, every push to main branch triggers automatic deployment

Stage
You can easily create a staging setup with a separate branch

Update
Each and any push (or merge) into main triggers an automated deployment

WHEN WOULD I USE THAT AGAIN?

Production environments
If you need an audit trail and reproducibility this option is your friend

Multiple environments
Easy and identical config for stage and prod environments
(or with a bit more effort for multiple tenants/domains)

Team Collaboration
Multiple people working on the setup

Configuration as Code
Everything is beautifully versioned and documented.

Backup and disaster recovery
Your infrastructure is already backed up in Git

GIT OPS IS YOUR FRIEND

IS THAT ALL THERE IS?

One more thing...

PLUGINS

Custom Dockerfile
Setup plugin import in custom Mautic dockerfile

Working on Filebrowser Solution
Setup filebrowser/filebrowser in coolify

Point to shared (plugin) volume

Start in UI when needed for plugin upload

Shut down to minimize surface area afterwards

Install
Install plugin via Mautic UI

UPDATES

I haven't done Mautic updates with this setup yet

Will probably be reporting on hat next year

;-)

BACKUPS

Overall Coolify DB auto backuped every night

Run DB backups regularly

Create a playground for test-deploying with backup data and config

Optionally: Use Hetzner auto backup (+20% cost)

GOING ONE STEP FURTHER

RESULTS

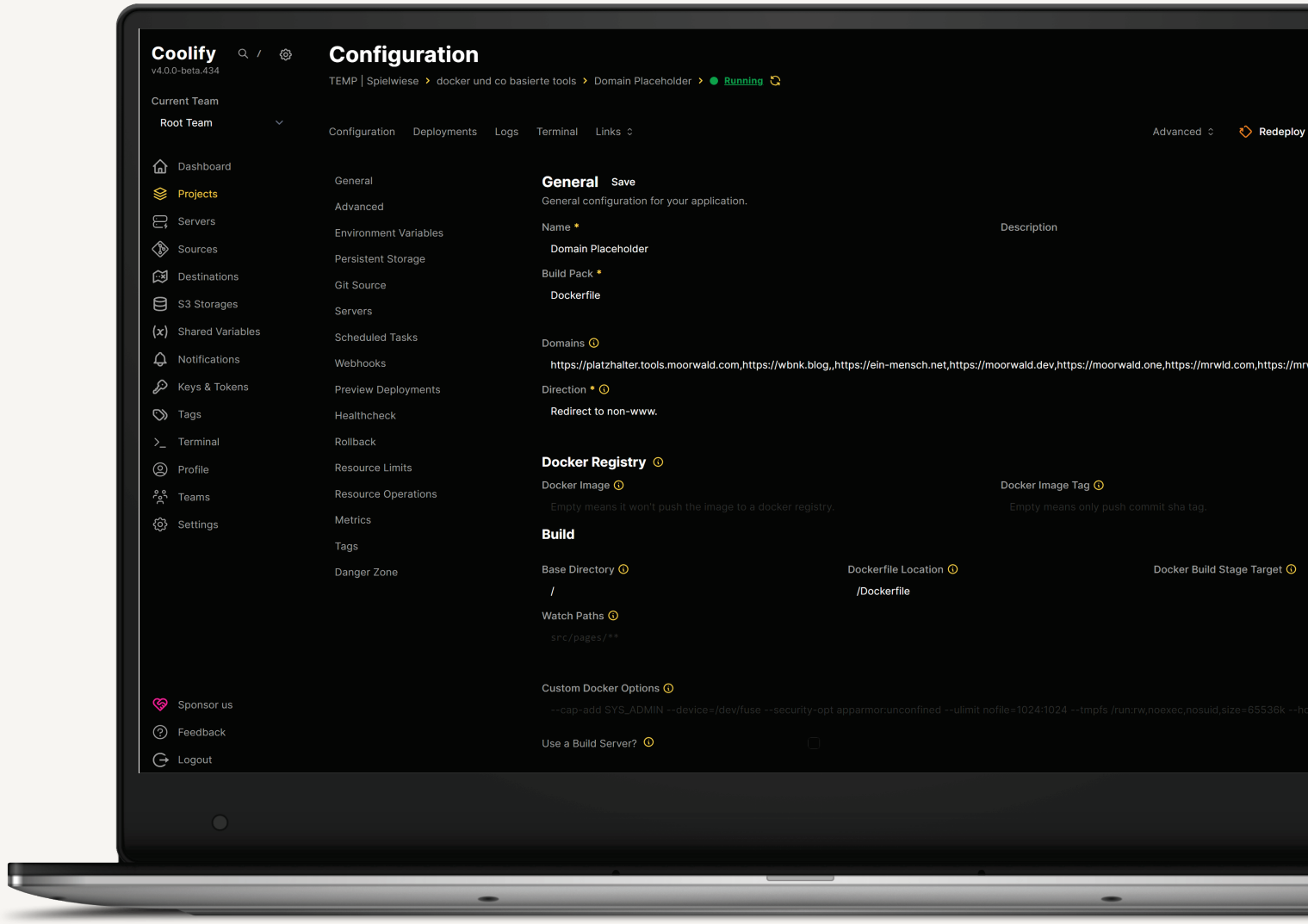
6 month without a hickup
Solution is now running smoothly for > 6 months

Easy two container setup
Reduced complexity, great performance

Inspiration for next iteration
My next iteration will definitely contain an upload capability for plugin files*

Never had less trouble with hosting
I have never slept better when hosting for clients (or myself actually)

Happy ~~wife~~client, happy life
Client is happy and I can focus on helping them grow their business



* If Mautic doesn't gain the file upload capability for plugins like with themes

moorwald
NATÜRLICH ONLINE WACHSEN

Code Repo & Documentation

Github:
/moorwald/mautic-world-conference-2025-hosting-mautic-on-coolify

Thank you!

QUESTIONS?