

AI-04635 OpenSEO Docker Setup

Explain the SEO, what's the job? What tools are needed? How to get started? What are the top 5 Docker-deployable open source SEO?

SEO (Search Engine Optimization) is the practice of optimizing a website to increase its visibility in organic (unpaid) search engine results. The goal is to drive relevant traffic to your site by helping engines like Google understand your content and recognize your authority on a subject.

What's the Job? ⇄

An SEO Specialist's main job is to align a website's structure, content, and reputation with search engine ranking algorithms. The job generally breaks down into four key areas:

1. **Technical SEO:** Ensuring search engine bots can crawl, index, and render your site efficiently. This includes fixing broken links, improving page load speeds, implementing schema markup, and configuring site architecture.
2. **On-Page SEO:** Optimizing individual web pages for specific keywords. Tasks include crafting high-quality content, writing meta tags, structuring header tags (H1, H2), and maintaining proper keyword intent.
3. **Off-Page SEO:** Building external trust and authority for your site. This revolves around earning high-quality backlinks from other websites, digital PR, and social signals.
4. **Keyword & Competitor Research:** Identifying what your target audience is searching for, evaluating keyword difficulty, and finding content gaps that your competitors are missing.

What Tools Are Needed?

To do the job effectively, an SEO pro relies on a toolkit covering distinct functions:

- **Search Engine Consoles (First-Party Data):** Google Search Console and Bing Webmaster Tools (tracks real queries, indexing status, and click performance).
- **Analytics Platforms:** Google Analytics 4, Matomo, or Plausible (tracks user behavior after they land on your site).
- **Crawlers & Audit Tools:** Screaming Frog or SEOnaut (scans thousands of URLs to surface broken links, canonical issues, and metadata gaps).
- **Rank Trackers:** SerpBear or AccuRanker (monitors your rankings for thousands of keywords over time).

- **All-in-One Data Suites:** Semrush, Ahrefs, or OpenSEO (provides backlink indexes, competitor domain statistics, and volume estimates).

How to Get Started

1. **Learn the Fundamentals:** Understand basic search engine mechanics (crawling $\rightarrow$ indexing $\rightarrow$ ranking). Read Google's official *Search Essentials* guide.
2. **Set Up Google Search Console & Analytics:** Install tracking on a personal website or blog. Analyzing your own real traffic is the best way to learn.
3. **Conduct an Initial Audit:** Use a crawler tool to run a diagnostic on your site. Practice fixing technical issues like missing alt text, duplicate $H1$ tags, or slow-loading images.
4. **Execute Keyword Research & Write Content:** Identify a handful of low-competition keywords, write detailed content targeting those queries, and optimize the title and headings.
5. **Track & Iterate:** Monitor your positions over 30 to 90 days, refine the content based on ranking movement, and practice earning links.

Top 5 Docker-Deployable Open-Source SEO Tools

If you want complete data privacy and zero monthly per-seat or per-keyword fees, you can self-host these top open-source tools using Docker:

Tool	Category	Key Features	Deployment / Run Setup
SerpBear	Rank Tracking	Tracks unlimited keywords across Google/Bing; provides positional history, domain overview, and email alerts.	Runs via Node.js/SQLite in Docker; requires bringing your own free/cheap SERP scraping API key.
OpenSEO	All-in-One Suite	Open-source alternative to Ahrefs. Does keyword research, site audits, rank tracking, and AI brand visibility tracking.	Deploys natively via Docker Compose; relies on pay-as-you-go DataForSEO API keys for raw index data.
SEOnaut	Technical Site Crawler	Light, automated crawling suite for identifying canonical issues, broken links, missing meta tags, and indexing issues.	Self-hosted web app built on Go/PostgreSQL; pre-packaged Docker image available.

Matomo	Analytics & Core Web Vitals	Enterprise-grade Google Analytics alternative. Offers 100% data ownership, heatmap tracking, and page speed reporting.	Full PHP/MySQL application with official Docker Compose definitions.
Umami	Privacy Analytics	Fast, lightweight (<1KB script) privacy-focused web analytics tool that won't slow down site speed.	Deploys in minutes via Docker with Node.js and a PostgreSQL/MySQL database.

Create a step by step instructions on how to deploy OpenSEO using compose file. Explain the backend process, database and workflow first. I bought a domain name from Cloudflare, do I need to purchase additional product such as Cloudflare Workers? I have an NPM in my dockerlab, a Postgres db with username postgres and password as postgres and in the dockerlab-network.

How OpenSEO Works (Backend, Database, & Workflow)

OpenSEO functions as an **analytics orchestration engine**. Unlike traditional SEO platforms that crawl the entire web independently (requiring massive data center infrastructure), OpenSEO decouples the front-end user experience from raw data collection:

- **The Backend:** Built as a modern TypeScript/Node web service. When you perform a site audit, keyword search, or rank check, the backend translates your requests and proxies them to specialized external APIs—primarily **DataForSEO** (for raw SERP and keyword metrics) and **OpenRouter** (for AI-assisted analysis).
- **The Database:** By default, OpenSEO uses an embedded **SQLite/Wrangler** database volume inside its container to store configuration, cached API responses, project settings, and keyword lists.
- **The Workflow:**
 1. You create a project for a domain in the OpenSEO UI.
 2. The application requests SERP or technical audit data via API connections.
 3. Raw data is processed, structured, and saved locally inside the database.
 4. Nginx Proxy Manager (NPM) routes your incoming HTTPS traffic directly to OpenSEO's container port (3001).

Cloudflare Requirements

No, you do NOT need Cloudflare Workers or any paid Cloudflare additions.

Because you are using Docker Compose, NPM, and your own domain:

- **Cloudflare DNS** is all you need to point your domain name (or a subdomain like `openseo.yourdomain.com`) to your home server's external IP.
- Cloudflare Workers / D1 / R2 are only required if you choose to deploy OpenSEO *serverlessly* on Cloudflare's edge platform rather than inside Docker.

Prerequisites & Network Strategy

You already have an active custom bridge network (`dockerlab-network`) where your Nginx Proxy Manager and PostgreSQL database reside.

Because OpenSEO is designed for self-hosting with `AUTH_MODE=local_noauth` (no built-in login screen when self-hosted), **never expose its port directly to the host network**. We will attach OpenSEO to `dockerlab-network` so NPM can access it internally via container name, keeping it hidden from the open internet.

Step-by-Step Deployment Guide

Step 1: Obtain a DataForSEO API Key

OpenSEO requires a DataForSEO account to pull search engine data.

1. Sign up at [DataForSEO](#) (they provide free initial test credits).
2. Generate your API Credentials (Login Email & API Password).
3. Create your Base64 authorization string by running this command in your terminal (replace with your actual credentials):

Bash

```
echo -n "your_email@example.com:your_api_password" | base64
```

Copy the output string.

Step 2: Create Project Folder & Environment File

On your Docker host machine, create a dedicated folder and a `.env` configuration file:

Bash

```
mkdir -p ~/dockerlab/openseo && cd ~/dockerlab/openseo  
nano .env
```

Paste the following environment variables (adjust `ALLOWED_HOST` to match your domain):

Code snippet

```
# OpenSEO Image & General Settings  
OPEN_SEO_IMAGE=ghcr.io/every-app/open-seo:latest  
PORT=3001  
AUTH_MODE=local_noauth  
  
# Domain & Proxy Settings (Matches your NPM setup)  
ALLOWED_HOST=openseo.yourdomain.com  
  
# DataForSEO API Base64 Credentials  
DATAFORSEO_API_KEY=your_base64_encoded_string_here  
  
# Optional: OpenRouter API key for AI features (SAM Agent)  
OPENROUTER_API_KEY=  
  
# Optional: Disable Telemetry  
OPENSEO_TELEMETRY_DISABLED=1
```

Step 3: Create the `docker-compose.yml` File

In the same directory (`~/dockerlab/openseo`), create your `docker-compose.yml` :

Bash

```
nano docker-compose.yml
```

Paste the following Docker Compose configuration:

YAML

```
services:
  open-seo:
    image: ${OPEN_SEO_IMAGE:-ghcr.io/every-app/open-seo:latest}
    container_name: open-seo
    restart: unless-stopped
    env_file:
      - .env
    environment:
      - CLOUDFLARE_INCLUDE_PROCESS_ENV=true
      - PORT=${PORT:-3001}
      - ALLOWED_HOST=${ALLOWED_HOST:-}
      - AUTH_MODE=local_noauth
      - OPENSEO_TELEMETRY_DISABLED=${OPENSEO_TELEMETRY_DISABLED:-1}
      - DATAFORSEO_API_KEY=${DATAFORSEO_API_KEY}
      - OPENROUTER_API_KEY=${OPENROUTER_API_KEY:-}
      - VITE_SHOW_DEVTOOLS=false
    volumes:
      - openseo_data:/app/.wrangler
    networks:
      - dockerlab-network

volumes:
  openseo_data:
    name: openseo_data

networks:
  dockerlab-network:
    external: true
```

Step 4: Launch the Container

Start the container in detached mode:

Bash

```
docker compose up -d
```

Check the logs to ensure the application starts up cleanly:

Bash

```
docker compose logs -f open-seo
```

Step 5: Configure Nginx Proxy Manager (NPM)

1. Open your NPM Web Admin UI.
2. Go to **Hosts** $\rightarrow$ **Proxy Hosts** $\rightarrow$ **Add Proxy Host**.
3. Fill in the **Details** tab:
 - **Domain Names:** `openseo.yourdomain.com`
 - **Scheme:** `http`
 - **Forward Hostname / IP:** `open-seo` (using the container name on `dockerlab-network`)
 - **Forward Port:** `3001`
 - Enable **Block Common Exploits** and **Websockets Support**.
4. Fill in the **SSL** tab:
 - Select **Request a new SSL Certificate** (Let's Encrypt).
 - Enable **Use a DNS Challenge** (using your Cloudflare API token) OR standard HTTP challenge.
 - Enable **Force SSL** and **HTTP/2 Support**.
5. Save the configuration.

Step 6: Access & Secure Your Deployment

Navigate to `[https://openseo.yourdomain.com](https://openseo.yourdomain.com)` in your browser. Because OpenSEO runs in `local_noauth` mode for Docker deployments, it will drop you straight into the dashboard.

Security Note: Since OpenSEO disables internal password checks in Docker mode, use **NPM's Access Lists** (under the *Access Lists* tab in NPM) to add HTTP Basic Auth (username/password protection) to your `openseo.yourdomain.com` host entry. This guarantees nobody on the web can access your instance.