Configuration for hipoterapiabydgoszcz

nowa.hipoterapia.bydgoszcz.pl

Deploys from **GitHub** with Astro.

Last update at 8:43 PM.

General

- Project details
- Visitor access
- Status badges
- Project members
- Danger zone

Build & deploy

- Continuous deployment
- Post processing
- Split Testing
- Build plugins

Environment variables

- Environment variables
- Project policies

Visual editor

Notifications

- Slack notifications
- Emails and webhooks

Identity

- Overview

- Users
- Registration
- Emails
- Services
- Danger zone

Access & security

- Firewall Traffic Rules
- Web Application Firewall
- OAuth

Emails

Build & deploy settings

Continuous deployment

Configuration for continuous deployment from a Git repository

Repository

Your project is linked to a Git repository for continuous deployment.

Current repository:

github.com/adamkempczynski/hipoterapia-bydgoszcz

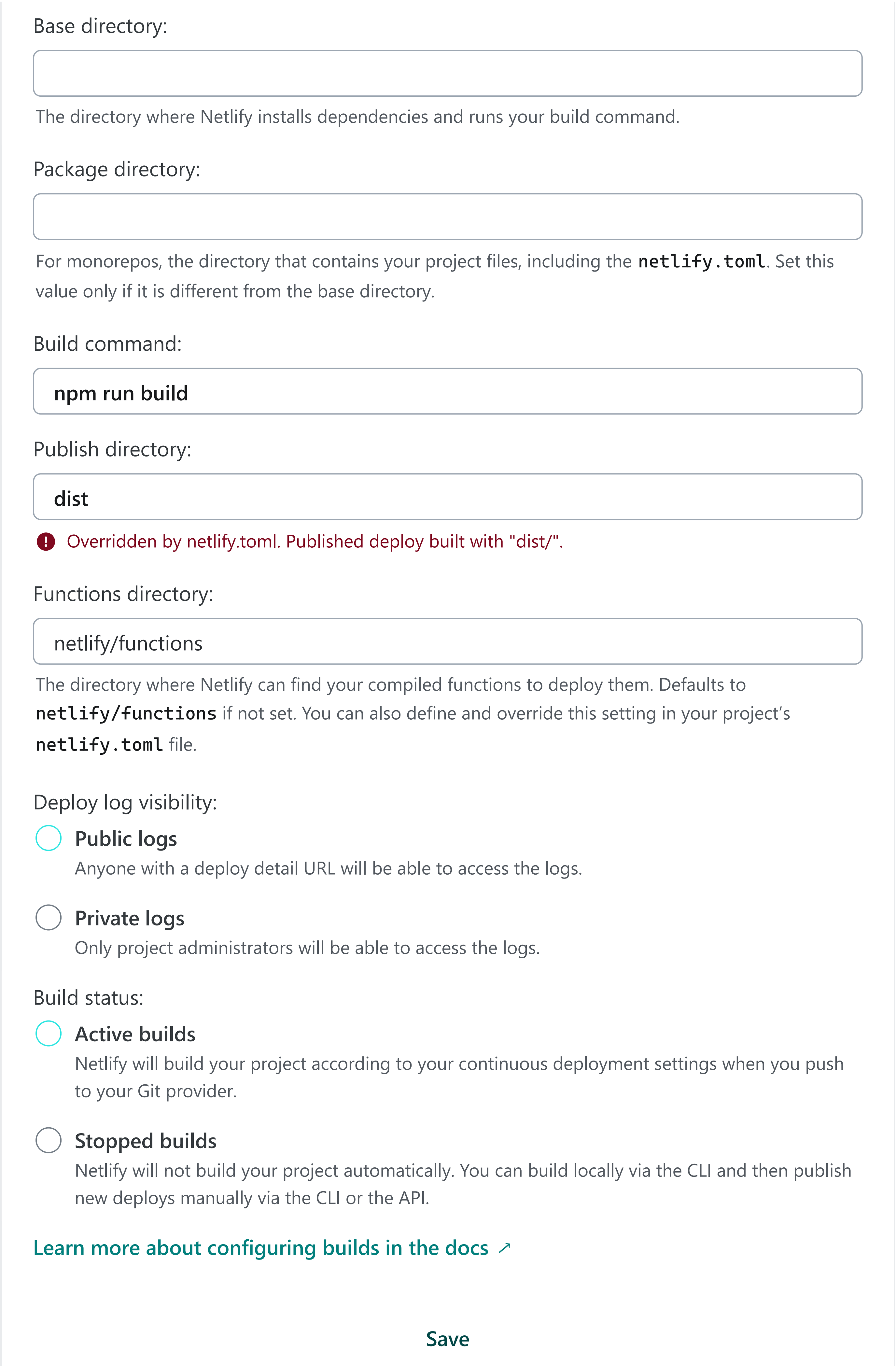
Learn more about continuous deployment in the docs ↗

Manage repository ⌄

Build settings

Runtime: Select ⌄

Runtimes are automatically installed on project creation. You can change this at any time, although it may affect your builds.



Base directory:

The directory where Netlify installs dependencies and runs your build command.

Package directory:

For monorepos, the directory that contains your project files, including the `netlify.toml`. Set this value only if it is different from the base directory.

Build command:

npm run build

Publish directory:

dist

❗ Overridden by netlify.toml. Published deploy built with "dist/".

Functions directory:

netlify/functions

The directory where Netlify can find your compiled functions to deploy them. Defaults to `netlify/functions` if not set. You can also define and override this setting in your project’s `netlify.toml` file.

Deploy log visibility:

○ **Public logs**
Anyone with a deploy detail URL will be able to access the logs.

○ **Private logs**
Only project administrators will be able to access the logs.

Build status:

○ **Active builds**
Netlify will build your project according to your continuous deployment settings when you push to your Git provider.

○ **Stopped builds**
Netlify will not build your project automatically. You can build locally via the CLI and then publish new deploys manually via the CLI or the API.

Learn more about configuring builds in the docs ↗

Cancel

Enforce deployment methods

All deployment methods can deploy to production.

The Netlify **CLI**, **MCP**, **API**, and **Git workflows** will be able to deploy to production.

Configure

Preview Server settings

Configure how Netlify CLI runs your project in the Preview Server. By default, it uses the same settings as when running locally on your machine.

Preview Server command:

Not set

Target port:

Not set

Preview Server size:

1 vCPU, 4 GB RAM, 20 GB Storage

Learn more about configuring the Netlify CLI ↗

Configure

Functions region

Configure how functions are deployed for your project.

Region:

US East (Ohio) - us-east-2

Learn more about functions region in the docs ↗

Upgrade to customize

Dependency management

Manage the software and tool versions installed in the build environment for your project.

Node.js:

22.x

Learn more about dependency management in the docs ↗

Configure

Branches and deploy contexts

Deploy contexts are branch-based environments that enable you to configure builds depending on the context. This includes production and preview environments.

Production branch:

main

Branch deploys:

Deploy only the production branch

Deploy Previews:

Any pull request against your production branch / branch deploy branches

Learn more about branches and deploys in the docs ↗

Configure

Collaboration tools

The Netlify Drawer enables your team to collaborate on preview environments before changes go live to production. Discuss changes, file new issues, or annotate screenshots directly from a preview context.

Netlify Drawer:

Enabled on Deploy Previews

Heads-up display:

Disabled

Learn more about collaboration tools in the docs ↗

Configure

Build hooks

Build hooks give you a unique URL you can use to trigger a build.

Learn more about build hooks in the docs ↗

Add build hook

Preview Server hooks

Preview Server hooks give you a unique URL you can use to trigger a Preview Server operation.

Add Preview Server hook

Deploy key

A deploy key grants permissions to our build system to access your private submodule.

Learn more about deploy keys in the docs ↗

Generate public deploy key

Build image selection

Select a different build image to change the operating system and supported software versions in the environment where our buildbot builds your project.

Build image:

Ubuntu Noble 24.04 (default)

Learn more about build image selection in the docs ↗

Configure

Automatic deploy deletion

Control when deploys are automatically deleted. Netlify will not automatically delete the deploy currently published on your project, nor the most recent successful deploy from a Git branch.

Delete deploys after:

30 days

Learn more about deploy deletion in the docs ↗

Upgrade to customize

Post processing

Control the post processing and optimizations Netlify can do for you

Snippet injection

Inject analytics or other scripts into the HTML of your project.

Learn more about snippet injection in the docs ↗

Add snippet

Pretty URLs

Pretty URLs are enabled. ✓

Learn more about Pretty URLs in the docs ↗

Configure

Prerendering

Beta

Enable prerendering to allow crawlers used by search engines and social networks to see the pages rendered by your app.

Learn more about prerendering in the docs ↗

Configure

Split Testing

Beta

Simple yet powerful split testing at the edge node. Test alternate designs or gradually roll out features by serving multiple Branch Deploys and distributing them across users.

Learn more about Split Testing in the docs ↗

Get started with Split Testing

This feature requires branch deploys ↗

Activate branch deploys and push changes to the branches you want to test.

Activate branch deploys

Build plugins

Discover and configure build plugins

Learn more about build plugins in the docs ↗

Search build plugins

All

21YunBox
by 21YunBox

Deploy to China straight from your Netlify Dashboard

Enable

Options

Algolia Crawler
by Netlify

Automatically crawls your website and generates an Algolia Search index.

Enable

Options

Visual Diff (Applitools)
by Applitools

Require visual changes on production to be manually approved before going live!

Enable

Options

Auth0 extension
by Netlify

Connect your Netlify account to Auth0

Enable Options ⌄

Cloudflare cache purge
by Chris Muktar

Clear Cloudflare cache when build completes.

Enable Options ⌄

Cloudinary
by Cloudinary

Automatically optimize your Netlify site images and deliver them in modern formats with Cloudinary.

Enable Options ⌄

CommandBar
by CommandBar

CommandBar plugin for websites deployed to Netlify.

Enable Options ⌄

Cypress
by Cypress

Runs Cypress end-to-end tests after Netlify builds the site.

Enable Options ⌄

DebugBear web performance
by DebugBear

Run tests on DebugBear to see how your deployments affect page performance and Lighthouse scores

Enable Options ⌄

Formspree

by Formspree

Deploy React forms with formspree.json

Enable

Options ⌄

Ghost Inspector

by Ghost Inspector

Run Ghost Inspector automated browser tests on your Netlify deploys.

Enable

Options ⌄

Inngest

by Inngest

Automatically deploy Inngest functions to Netlify's platform.

Enable

Options ⌄

Lighthouse

by Netlify

Automatically run a Lighthouse audit on your website after every build.

Enable

Options ⌄

a11y

by sw-yx

Build a more accessible web! Run your critical pages through pa11y and fail build if accessibility failures are found.

Enable

Options ⌄

Airbrake

by Bharathvaj Ganesan

Automatically notifies Airbrake of new site deploys.

Enable Options ⌄

AMP server-side rendering

by Martin Bean

A Netlify plugin to server-side render your AMP pages.

Enable Options ⌄

Build Logger

by Amar Parmar

Send build events to a logging service.

Enable Options ⌄

Cecil cache

by Arnaud Ligny

Persist the Cecil cache between Netlify builds.

Enable Options ⌄

Checklinks

by Peter Müller

Checklinks helps you keep all your asset references correct and avoid embarrassing broken links to your internal pages, or even to external pages you link out to.

Enable Options ⌄

Chromium

by soofka

Installs Chromium (installs NPM Chromium package and sets environment variable to location of binaries); useful for other tools requiring Chromium to run, e.g. Lighthouse CI.

Enable Options ⌄

Contextual ENV

by Chris Ball

Replaces ENV vars with ENV vars that are prefixed/suffixed with the context or branch name.

Enable Options

Dareboost

by Boris Schapira

After a successful build, create an event in your Dareboost monitoring. If you have subscribed to API credits, you can automatically launch analyses.

Enable Options

Debug cache

by Netlify

Debug & verify the contents of your Netlify build cache.

Enable Options

Fetch feeds

by Phil Hawksworth

A Netlify plugin to fetch and cache content from remote feeds including RSS and JSON.

Enable Options

Flutter SDK

by Ben Evans

Install the Flutter SDK for Building and Deploying Flutter Web Apps.

Enable Options

Ghost Markdown

by David Darnes

Generates posts, pages, tag pages and author pages from a Ghost publication as markdown files, using the Ghost Content API.

Enable

Options

Gridsome Cache

by Eric McCormick

Persist the Gridsome cache between Netlify builds for huge speed improvements!

Enable

Options

Hashfiles

by Peter Müller

Hashfiles sets you up with an optimal caching strategy for static sites, where static assets across pages are cached for as long as possible in the visitors browser and never have to be re-requested.

Enable

Options

HTML Validate

by Oliver Roick

Validate HTML generated by your build.

Enable

Options

Hugo Cache Resources

by Cassius

Persist Hugo resources folder between Netlify builds for huge build speed improvements!

Enable

Options

Image Optim

by Chris Draycott-Wheatley

Optimize images as part of your Netlify build process.

Enable

Options

Inline Critical CSS

by Tom Bonnike

Automatically extract and inline the critical CSS of your pages in order to render content to the user as fast as possible.

Enable Options ⌄

Inline Functions Environment Variables

by Zhenhua Ben Cao

Inline process.env.VARIABLE in netlify functions with netlify build time environment variables.

Enable Options ⌄

Inline Source

by Tom Bonnike

Improve your site’s performance by inlining some of your assets/sources, reducing the number of HTTP requests your users need to make.

Enable Options ⌄

Is Website Vulnerable

by Erez Rokah

A Netlify plugin that uses Snyk to test for security vulnerabilities in a website's JavaScript libraries.

Enable Options ⌄

JavaScript Obfuscator

by Anish Roy

An obfuscator for your JavaScript source code.

Enable Options ⌄

Jekyll Cache

Persist the Jekyll cache between Netlify builds.

Enable Options

Minify HTML

by Phil Hawksworth

A plugin to add HTML minification as a post-processing optimisation in Netlify.

Enable Options

Netlify Bundle ENV

by Hrishikesh Kokate

A Netlify Build Plugin to inject environment variables in Netlify Functions during Netlify Builds.

Enable Options

No More 404

by sw-yx

Check that you preserve your own internal URL structure between builds, accounting for Netlify Redirects. Don't break the web!

Enable Options

Pushover Notification

by Ashik Nesin

Send real time notification to your devices on build success/error via Pushover.net.

Enable Options

Replace

Replace ENV variables in your publish directory.

Enable Options

Sitemap

by Netlify

Automatically generate a sitemap for your website after it finishes building in Netlify.

Enable **Options** ⌄

Skip nx

by Fahd Allebdi

Skip not affected apps in a nx monorepo.

Enable **Options** ⌄

Subfont

by Peter Müller

Subfont post-processes your web page to analyse you usage of web fonts, then reworks your webpage to use an optimal font loading strategy for the best performance.

Enable **Options** ⌄

Submit Sitemap

by Cassius

Automatically submit your sitemap to Google and Yandex after every production build!

Enable **Options** ⌄

Use Env in Runtime

by Ben Lmsc

Make some environment variables available only at build time in the runtime of your app.

Enable **Options** ⌄

Webmentions

by Luke Bonaccorsi

Automatically discover any webmentions and send them after every production build.

Enable Options ⌄

New Relic
by Aaron Bassett

Automatically notifies New Relic of Netlify build events and installs the New Relic browser agent to monitor your sites.

Enable Options ⌄

Nimbella
by Nimbella

Nimbella plugin to extend Netlify Sites with stateful and portable serverless APIs.

Enable Options ⌄

PerfBeacon
by OnlineOrNot

Tell PerfBeacon to measure page speed after a successful build.

Enable Options ⌄

PageWatch Audit
by PageWatch

Check your site for layout errors, spelling mistakes, broken links and performance issues on every build.

Enable Options ⌄

QA Wolf
by QA Wolf

Run QA Wolf end-to-end tests on Netlify deployments.

Enable Options ⌄

Sentry

by Sentry

The Sentry integration on Netlify helps you quickly identify how code errors and performance issues impact your users.

Enable **Options ⌄**

Snaplet

by Snaplet

Bootstrap preview databases with your deploy previews on Netlify without affecting production.

Enable **Options ⌄**

Snyk Security

by Snyk

A Snyk Netlify plugin to find and monitor new security vulnerabilities in JavaScript libraries.

Enable **Options ⌄**

Speedcurve

by Andy Davies

After a successful build, tell SpeedCurve you've deployed and trigger a round of testing.

Enable **Options ⌄**

StepZen

by StepZen

Deploy a StepZen (http://stepzen.com) GraphQL API with any Netlify build.

Enable **Options ⌄**

Strapi

by Strapi

Strapi v4 plugin to trigger, monitor, and cancel deployments on one or more Netlify sites

Enable Options ⌄

TakeShape
by TakeShape

Automatically create and configure a TakeShape project for use with your Netlify site.

Enable Options ⌄

Very Good Security
by Very Good Security

Securely collect and use sensitive data without having it ever touch your systems.

Enable Options ⌄

Docs Pricing Support Blog Changelog Terms