

KNOWLEDGE BASE · PUBLIC DOCUMENT

Deploying a Next.js app: static export or Node server, and where each one fits

Node.js, Python & Docker

CODE

RLX-INT-KB-3336

VERSION

v2026.10.10

COUNTRY · LANGUAGE

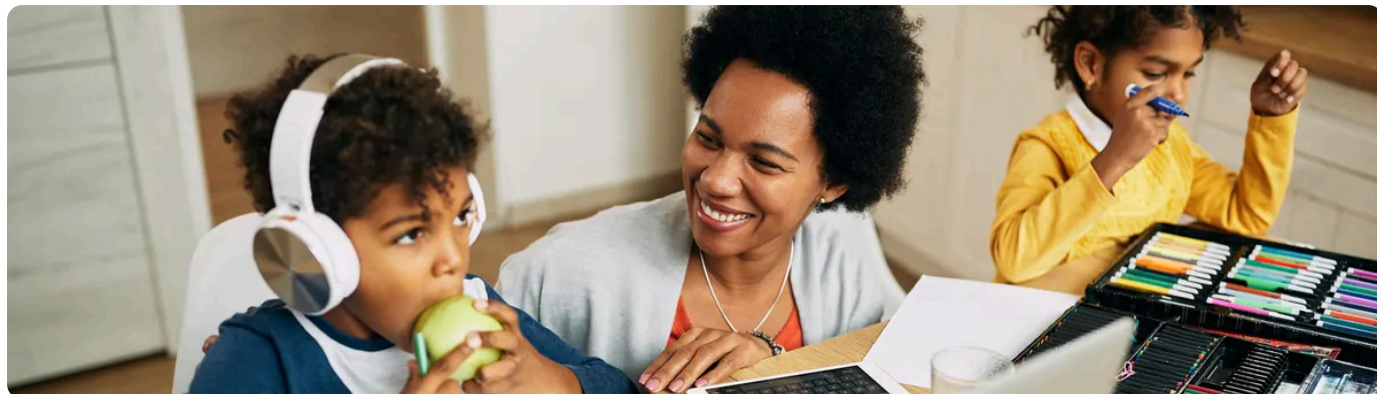
International · English

MozOut International

RLX Holdings (Pty) Ltd. · Group RLX Holdings

Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

<https://www.my.mozout.com/knowledgebase/3336>



Next.js can come out in two very different ways, and picking the wrong one lies behind almost every deployment problem. **The short answer:** if your site does not need code running on the server for every visit, make a **static export** and publish it as plain files, which fits any hosting plan. If it does (pages rendered on the server, API routes, middleware), you need an **always-on Node process**: it fits “Setup Node.js App” when it is simple, and sits better on a VPS.

The possible outputs

Mode	What it produces	Where it runs	What you give up
Static export	A folder of HTML, CSS and JavaScript files (called out by default).	On any hosting plan, like an ordinary site, in the domain’s public folder.	API routes, server-side rendering and middleware. According to the Next.js documentation, <code>next/image</code> images need <code>unoptimized: true</code> .
Node server (next start)	A Node process that answers every request.	In cPanel’s “Setup Node.js App”, or on a VPS with PM2 and nginx.	It uses memory all the time and is subject to the account limits.
standalone mode	A <code>.next/standalone</code> folder with a <code>server.js</code> and only what is needed.	Like the Node server; it is the lightest thing to copy to a VPS or into a Docker image.	You must copy the <code>public</code> and <code>.next/static</code> folders by hand, according to the documentation.

Static export, step by step

1

Ask for the export. In Next's configuration file (`next.config.js`) set `output: 'export'` . If you use Next's images, also add `images: { unoptimized: true }` .

2

Build on your own computer:

```
npm run build
```

The `out` folder appears with the finished site.

3

Upload the contents of `out` (not the folder itself) to the domain's public folder, over FTP: **uploading the site with FileZilla**.

4

Test by opening an internal link and refreshing the page. If you get a 404, it is the routing problem: **publishing a React or Vue site**.

A Node server in cPanel

1

Build on your computer and upload only the result: the `.next` and `public` folders, `package.json`, `package-lock.json` and the configuration file. **Do not upload** `node_modules`. Building on the server uses memory and thousands of files, and the account has limits for both: **the limits nobody advertises**.

2

Create a startup file (for example `server.js`) in the application root. It starts Next and listens on whatever port Passenger gives it:

```
const { createServer } = require("http");
const next = require("next");
const app = next({ dev: false });
const handle = app.getRequestHandler();
app.prepare().then(() => {
  createServer((req, res) => handle(req, res)).listen(process.env.PORT || 3000);
});
```

3

Create the application in “Setup Node.js App”: the Node version your Next asks for, *production* mode, the root, the address and the startup file. Use “Run NPM Install” for the dependencies. The general walk-through is in **getting a Node.js application running**.

4

Set the variables on the application screen and restart. Note that the ones starting with `NEXT_PUBLIC_` are baked into the code at build time, so changing them means building again.

Not everything fits a shared account. A Next.js server is a process that stays in memory. If it falls over or slows down for no obvious reason, see **the causes of the 503 error**. For background jobs, long connections or heavy traffic, the right place is a VPS: **keeping the app running with PM2**.

When in doubt, start with the static export. It is the cheapest to maintain, the fastest to serve and leaves nothing running that could break. Move to a Node server only when a page really needs server data on every request.

Does your application need a process of its own and more room? Have a look at our VPS plans.

[See the VPS servers](#)

SEE ALSO

[Publishing a React or Vue site: the build folder and the 404 on refresh](#)

[Getting a Node.js application running](#)

[Node.js in cPanel: choosing the version and using npm](#)



Online version of this document

MozOut is a leading internet services company, with a mission to help businesses to be born and prosper through technology.

2020

FOUNDED

37,000+

CLIENTS

130+

SIGN-UPS A DAY

MozOut International

RLX Holdings (Pty) Ltd. • Group RLX Holdings

TAX NO. (SARS)

943 907 6234

REGISTRATION NO.
(CIPC)

2025/667953/07

REGISTERED OFFICE

1111 Olive Street
Saint Louis 63101, Missouri, United States

CONTACT

info@mozout.com • +27 78 823 3325 • www.mozout.com

OPENING HOURS

Monday to Friday - 08 AM to 04 PM
Saturday & Sunday - Closed

RLX-INT-KB-3336 • v2026.10.10 • Generated on 18/10/2026 08:54