interweb

Generated on 15/11/2026 00:13

KNOWLEDGE BASE · PUBLIC DOCUMENT

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

Node.js & JavaScript

CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-1066	v2026.10.10	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/1066

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

Node.js & JavaScript Created on **October 10, 2026** 3 min read



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, `next/image` images need `unoptimized: true`.
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 `.next/standalone` folder with a `server.js` 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 `public` and `.next/static` 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.

RECOMMENDED

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** →

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-1066 · v2026.10.10 · Generated on 15/11/2026 00:13