

[Cloudflare Workers](#) is a serverless platform for creating entirely new applications or augmenting existing ones without configuring or maintaining infrastructure. With [Workers Sites](#) you can build any static site locally (e.g. your Gatsby project), and deploy to a [workers.dev](#) subdomain or custom domain.

This guide will get you started in a few steps:

1. Installing wrangler CLI

Workers Sites requires [wrangler](#), a tool to upload your static files. The most straight forward way to install wrangler is with [npm](#), run the following command:

```
npm install -g @cloudflare/wrangler
```

2. Initialize the Project

To create the Worker code that will serve your Gatsby files, from the root of your Gatsby project run:

```
wrangler init --site
```

You'll notice your project structure should now look something like:

```
+ | wrangler.toml
+ | workers-site
+ | | index.js
+ | | package-lock.json
+ | | package.json
  | | src
  | | | components
  | | | images
  | | | pages
  | | gatsby-config.js
  | | package.json
```

3. Build the project

To run your application on Cloudflare Workers Sites, you first need to build your application locally. This should generate your files in `./public`.

```
npm run build # Or npx gatsby build
```

4. Configure

To authenticate into your Cloudflare account run:

```
wrangler config
```

Follow the [Quick Start](#) for steps on gathering the correct account ID and API token to link wrangler to your Cloudflare account.

If you don't already have a workers.dev domain run:

```
wrangler subdomain
```

Then, add your account ID to the `wrangler.toml` file, and set `bucket` to `"./public"`, which is where Gatsby's built files are output by default:

```
name = "gatsby-project"
type = "webpack"
account_id = "abcd..."
workers_dev = true

[site]
bucket = "./public"
entry-point = "workers-site"
```

This deploys to your workers.dev subdomain. For a custom domain see [Quick Start](#).

5. Deploy

You can deploy your application by running the following command in the root of the project directory:

```
wrangler publish
```

Now your site is available at `gatsby-project.subdomain.workers.dev`!

6. CI with GitHub Actions

Use wrangler's GitHub action [plugin](#) to automatically deploy to Workers every time you push to master. Alternatively, you can use [Gatsby Cloud Hosting](#) or [Cloudflare Pages](#).

Once GitHub Actions is enabled on your repo, add a file to your project's root called `.github/workflows/main.yml` with the contents:

```
name: Deploy production site

on:
  push:
    branches:
      - master

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v1
      - name: Navigate to repo
        run: cd $GITHUB_WORKSPACE
      - uses: actions/setup-node@v1
        with:
          node-version: "10.x"
      - name: Install deps
        run: npm install
```

```
- name: Build docs
  run: npm run build
- name: Publish
  uses: cloudflare/wrangler-action@1.1.0
  with:
    apiToken: ${ secrets.CF_API_TOKEN }
    environment: "production"
```

Set up `CF_API_TOKEN` in GitHub secrets with appropriate values from [Quick Start](#).

Additional resources

- [Quickstart for Workers Sites](#)
- [GitHub Action wrangler plugin](#)