

# Recipes: Deploying Your Site

Showtime. Once you are happy with your site, you are ready to go live with it!

## Preparing for deployment

### Prerequisites

- A Gatsby site
- The Gatsby CLI installed

### Directions

1. Stop your development server if it is running (**Ctrl + C** on your command line in most cases)
2. For the standard site path at the root directory (/), run **gatsby build** using the Gatsby CLI on the command line. The built files will now be in the **public** folder.

**gatsby build**

3. To include a site path other than / (such as **/site-name/**), set a path prefix by adding the following to your **gatsby-config.js** and replacing **yourpathprefix** with your desired path prefix:

```
module.exports = {  
  pathPrefix: `/${yourpathprefix}`,  
}
```

There are a few reasons to do this – for instance, hosting a blog built with Gatsby on a domain with another site not built on Gatsby. The main site would direct to **example.com**, and the Gatsby site with a path prefix could live at **example.com/blog**.

4. With a path prefix set in **gatsby-config.js**, run **gatsby build** with the **--prefix-paths** flag to automatically add the prefix to the beginning of all Gatsby site URLs and **<Link>** tags.

**gatsby build --prefix-paths**

5. Make sure that your site looks the same when running `gatsby build` as with `gatsby develop`. By running `gatsby serve` when you build your site, you can test out (and debug if necessary) the finished product before deploying it live.

```
gatsby build && gatsby serve
```

### Additional resources

- Walk through building and deploying an example site in tutorial part one
- Learn about performance optimization
- Read about other deployment related topics
- Check out the deployment docs for specific hosting platforms and how to deploy to them

## Deploying to Netlify

Use `netlify-cli` to deploy your Gatsby application without leaving the command-line interface.

### Prerequisites

- A Gatsby site with a single component `index.js`
- The `netlify-cli` package installed
- The Gatsby CLI installed

### Directions

1. Build your gatsby application using `gatsby build`
2. Login into Netlify using `netlify login`
3. Run the command `netlify init`. Select the “Create & configure a new site” option.
4. Choose a custom website name if you want or press enter to receive a random one.
5. Choose your Team.
6. Change the deploy path to `public/`
7. Make sure that everything looks fine before deploying to production using `netlify deploy -d . --prod`

### Additional resources

- Hosting on Netlify
- `gatsby-plugin-netlify`

## Deploying to Vercel

Use Vercel CLI to deploy your Gatsby application without leaving the command-line interface.

### Prerequisites

- A Vercel account
- A Gatsby site with a single component `index.js`
- Vercel CLI package installed
- Gatsby CLI installed

### Directions

1. Login into Vercel CLI using `vercel login`
2. Change to the directory of your Gatsby.js application in the Terminal if you aren't already there
3. Run `vercel` to deploy it

### Additional resources

- [Deploying to Vercel](#)

## Deploying to Cloudflare Workers

Use `wrangler` to deploy your Gatsby application globally without leaving the command-line interface.

### Prerequisites

- An account on Cloudflare
- A Workers Unlimited plan for \$5/month to enable the KV store, which is required to serve the Gatsby files.
- A Gatsby site set up with Gatsby's CLI
- `wrangler` installed globally (`npm install -g @cloudflare/wrangler`)

### Directions

1. Build your Gatsby application using `gatsby build`
2. Run `wrangler config` where you'll be prompted for your Cloudflare API token
3. Run `wrangler init --site`
4. Configure `wrangler.toml`. First add account ID field and then either
  1. A free workers.dev domain by setting `workers_dev = true`
  2. A custom domain on Cloudflare by setting `workers_dev = false`, `zone_id = "abdc..` and `route = customdomain.com/*`
5. In `wrangler.toml` set `bucket = "./public"`

6. Run `wrangler publish` and your site will be deployed in seconds!

### Additional resources

- Hosting on Cloudflare

## Setting up Google Analytics

Use `gatsby-plugin-google-analytics` to track site activity and provide insights into how users access your website.

### Prerequisites

- A Gatsby site with a `gatsby-config.js` file and an `index.js` page
- The Gatsby CLI installed
- A domain from your provider of choice, e.g. AWS

### Verify the domain in `search.google.com`

1. Navigate to the Google search console to verify the domain by clicking on **Search Property > Add Property**. Type in your domain and press Continue.
2. Add a **TXT** record to your DNS configuration. Follow the directions for your provider, or refer to the Google documentation.

### Linking the domain to Google Analytics admin

1. Log into Google Analytics.
2. Click **Admin**.
3. Select **Create Property** in the Property column.
4. Choose **Web**.
5. Fill in the details and click **Create**.

### Getting your Google Analytics Tracking ID

1. Sign in to your Google Analytics account.
2. Click **Admin**.
3. Select an account from the menu in the ACCOUNT column.
4. Select a property from the menu in the PROPERTY column.
5. Under Property, click **Tracking Info > Tracking Code**. Your Tracking ID is displayed at the top of the page.

### Using the ID in the plugin

1. Run `npm install gatsby-plugin-google-analytics` in your terminal.
2. Add the following to your `gatsby-config.js` file.

```
module.exports = {  
  plugins: [  
    {  
      resolve: `gatsby-plugin-google-analytics`,  
      options: {  
        // replace "UA-XXXXXXXX-X" with your own Tracking ID  
        trackingId: "UA-XXXXXXXX-X",  
      },  
    },  
  ],  
}
```

3. Build and deploy your site to start seeing traffic in your Google Analytics dashboard.

#### **Additional resources**

- Adding Analytics