

Deploy your Astro Site

Ready to build and deploy your Astro site? Follow one of our guides to different deployment services or scroll down for general guidance about deploying an Astro site.

Deployment Guides [🔗](#)

**Netlify**

SSR Static

**Vercel**

SSR Static

**Deno Deploy**

SSR

**GitHub Pages**

Static

**GitLab Pages**

Static

**Cloudflare Pages**

SSR Static

**AWS**

Static

**AWS via Flightcontrol**

SSR Static

**AWS via SST**

SSR Static

**Google Cloud**

SSR Static

**Firebase Hosting**

SSR Static

**Heroku**

Static

**Microsoft Azure**

Static

**Buddy**

Static

**Edgio**

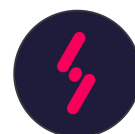
SSR Static

**Fly.io**

SSR Static

**Render**

Static

**Stormkit**

Static

**Surge****Cleavr****Kinsta**



Static



SSR

Static



SSR

Static



Deta Space

SSR

Static



Zeabur

SSR

Static



Zerops

SSR

Static

Quick Deploy Options [↗](#)

You can build and deploy an Astro site to a number of hosts quickly using either their website's dashboard UI or a CLI.

Website UI [↗](#)

A quick way to deploy your website is to connect your Astro project's online Git repository (e.g. GitHub, GitLab, Bitbucket) to a host provider and take advantage of continuous deployment using Git.

These host platforms automatically detect pushes to your Astro project's source repository, build your site and deploy it to the web at a custom URL or your personal domain. Often, setting up a deployment on these platforms will follow steps something like the following:

- 1 Add your repository to an online Git provider (e.g. in GitHub, GitLab, Bitbucket)
- 2 Choose a host that supports **continuous deployment** (e.g. [Netlify](#) or [Vercel](#)) and import your Git repository as a new site/project.

Many common hosts will recognize your project as an Astro site, and should choose the appropriate configuration settings to build and deploy your site as shown below. (If not, these settings can be changed.)

[i](#) Deploy settings

- **Build Command:** `astro build` OR `npm run build`
- **Publish directory:** `dist`

- 3 Click "Deploy" and your new website will be created at a unique URL for that host (e.g. `new-astro-site.netlify.app`).

The host will be automatically configured to watch your Git provider's main branch for changes, and to rebuild and republish your site at each new commit. These settings can typically be configured in your host provider's dashboard UI.

CLI Deployment [↗](#)

Some hosts will have their own command line interface (CLI) you can install globally to your machine using npm. Often, using a CLI to deploy looks something like the following:

- 1 Install your host's CLI globally, for example:

npm pnpm Yarn

```
npm install --global netlify-cli
```

- 2 Run the CLI and follow any instructions for authorization, setup etc.

- 3 Build your site and deploy to your host

Many common hosts will build and deploy your site for you. They will usually recognize your project as an Astro site, and should choose the appropriate configuration settings to build and deploy as shown below. (If not, these settings can be changed.)

Deploy settings

- **Build Command:** `astro build` OR `npm run build`
- **Publish directory:** `dist`

Other hosts will require you to [build your site locally](#) and deploy using the command line.

Building Your Site Locally [↗](#)

Many hosts like Netlify and Vercel will build your site for you and then publish that build output to the web. But, some sites will require you to build locally and then run a deploy command or upload your build output.

You may also wish to build locally to preview your site, or to catch any potential errors and

warnings in your own environment.

Run the command `npm run build` to build your Astro site.

npm

pnpm

Yarn

```
npm run build
```



By default, the build output will be placed at `dist/`. This location can be changed using the [outDir configuration option](#).

Adding an Adapter for SSR [🔗](#)

Note

Before deploying your Astro site with [SSR \(server-side rendering\)](#) enabled, make sure you have:

- Installed the [appropriate adapter](#) to your project dependencies (either manually, or using the adapter's `astro add` command, e.g. `npx astro add netlify`).
- [Added the adapter](#) to your `astro.config.mjs` file's import and default export when installing manually. (The `astro add` command will take care of this step for you!)

 [Edit page](#)

← Previous
Installation and Setup

Next
Migrate to Astro →