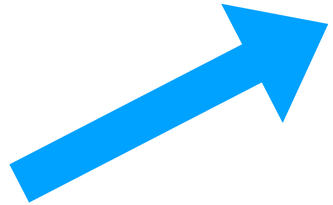
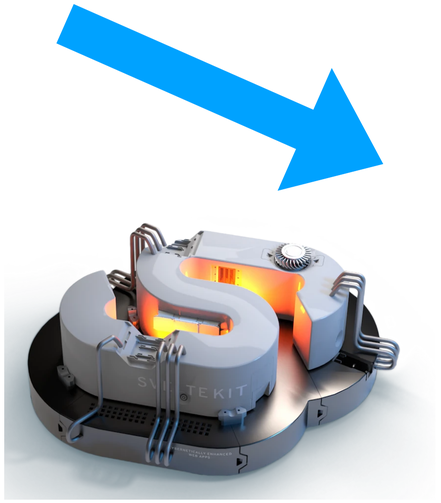
Deploy Front End Apps

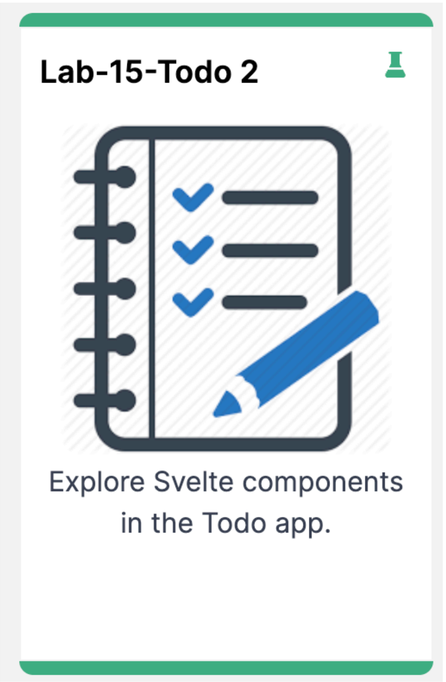
Deploying donation-svelte to Netlify & Vercel services

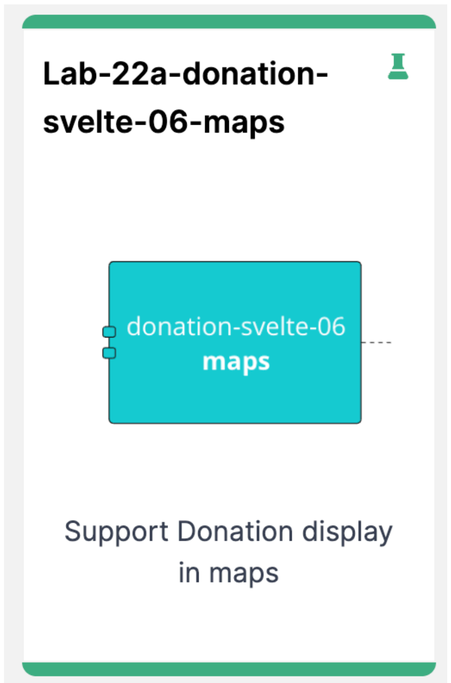
Deployment

- Distinguish between 2 types of deployment:
  - **Svelte**: the base framework, without router
  - **SvelteKit**: the Meta-framework, including router + other framework facilities



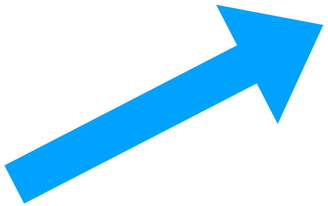


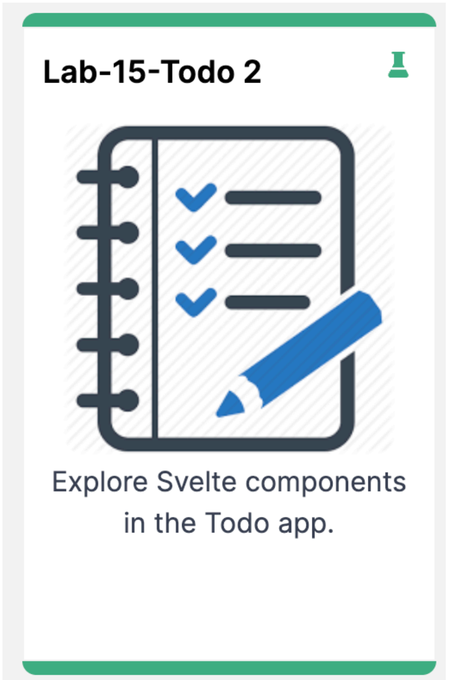




Svelte

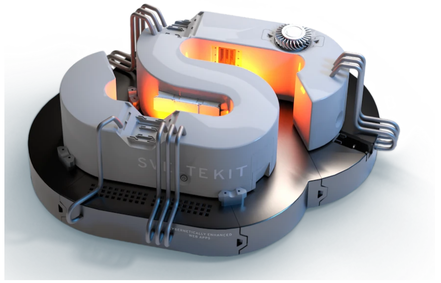
- Svelte applications are typical SPAs
  - Single Page Applications
  - Entire application in “bundled” into a single js download
  - When the application launches, all html/ javascript/css resources are loaded immediately
  - The local application then proceeds to run in the browser, without necessarily needing to load any further resources from the server.

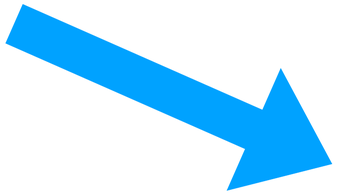


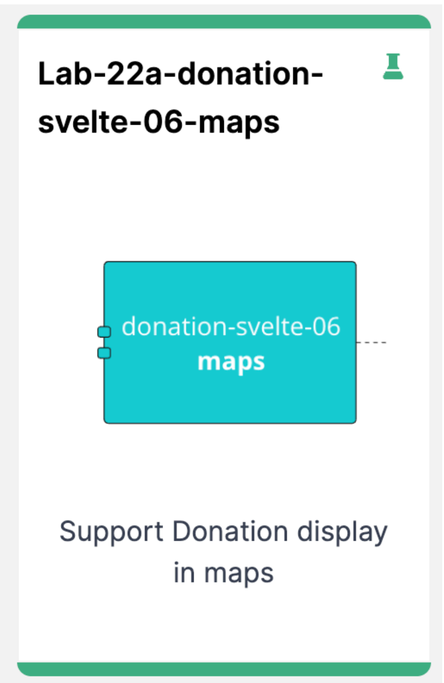


SvelteKit

- SvelteKit applications are typically MPAs
  - Multi Page Applications
  - Each route is potentially “bundled”
  - As each route is loaded, just that route is download and launched in the browser
  - Navigating to another route will download a different bundle
  - Browser cache will be heavily used to limit downloads







Svelte

Bundlers

- Module Bundler: take all of javascript files, bundle & compress.
- This monolithic file is then added to your index.html file as a script tag.
- When the browser loads your page, it only has to make one additional HTTP request.

PARCEL

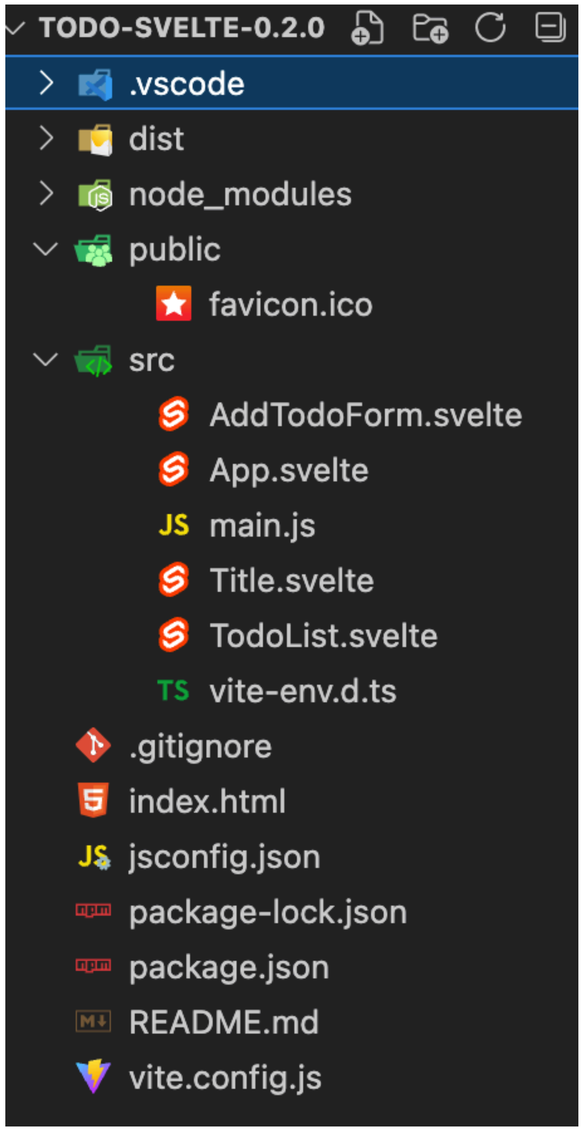
Blazing fast, zero configuration web application bundler

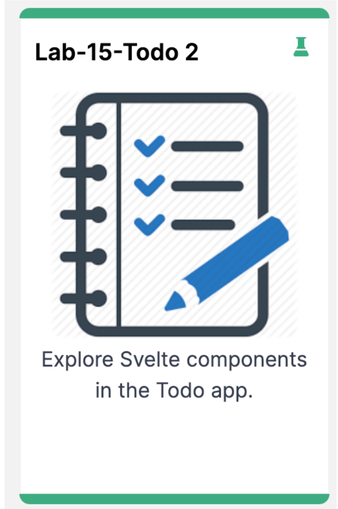
webpack

MODULE BUNDLER

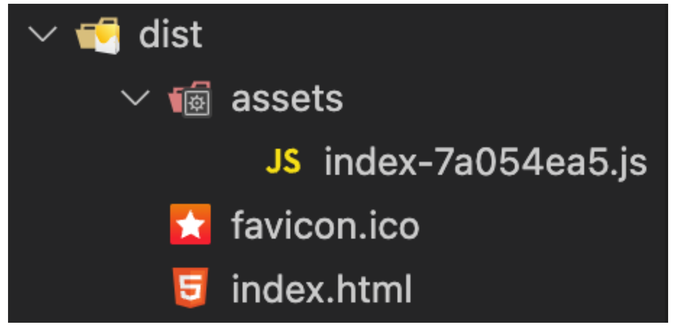
rollup.js

Vite





npm run build



- Compile all JS into index*.js
- Generate index html to reference load these scripts

```json
{
  "name": "todo-svelte",
  "private": true,
  "version": "0.2.0",
  "type": "module",
  ▷ Debug
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "devDependencies": {
    "@sveltejs/vite-plugin-svelte": "^3.0.2",
    "svelte": "^4.2.12",
    "vite": "^5.1.4"
  },
  "dependencies": {
    "uuid": "^9.0.1"
  }
}
```

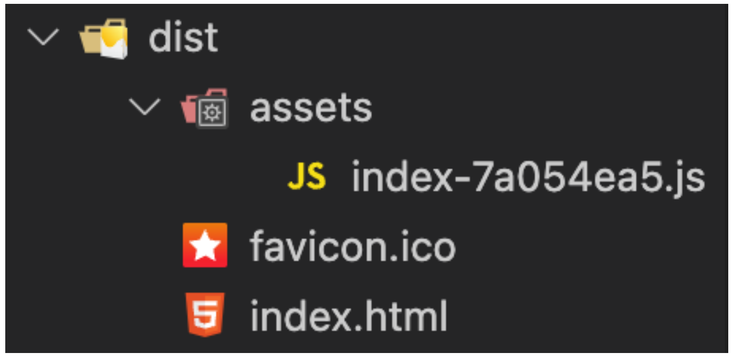
- Build = ‘vite build’

Build/Bundle the App

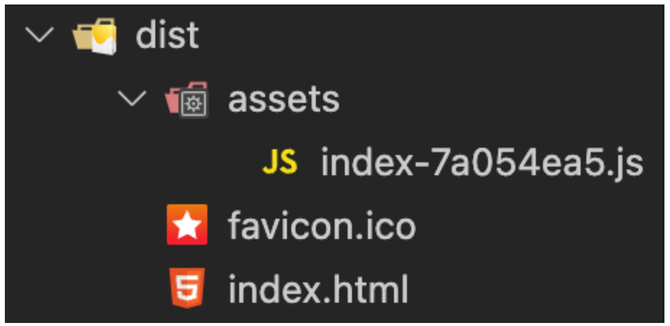
```
npm run build
```

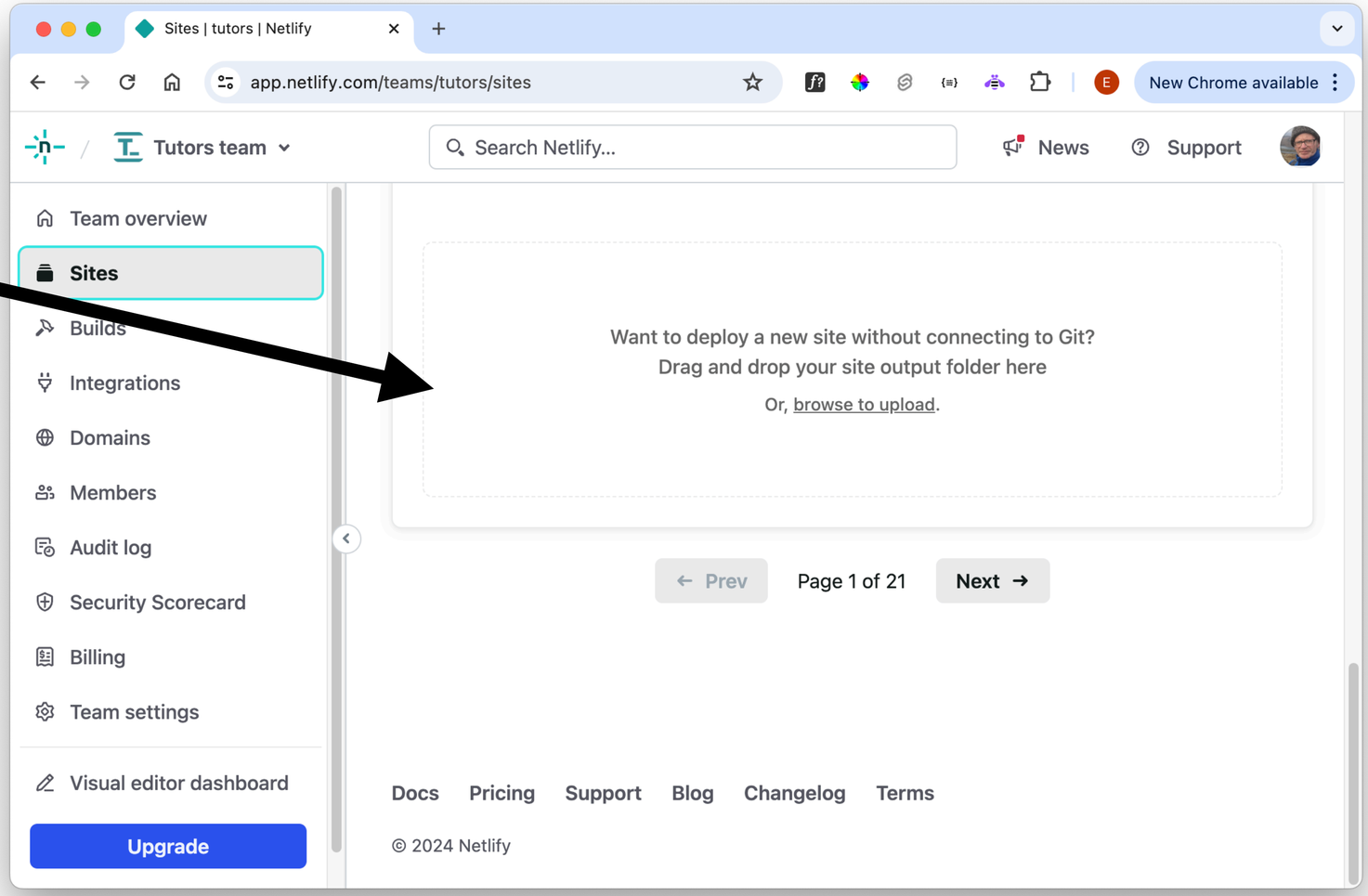
```
> todo-svelte@0.2.0 build
> vite build

vite v5.2.9 building for production...
✓ 45 modules transformed.
dist/index.html                  0.47 kB │ gzip: 0.32 kB
dist/assets/index-DvAsxv8I.js    9.70 kB │ gzip: 4.06 kB
✓ built in 184ms
```



Netlify Drag & Drop





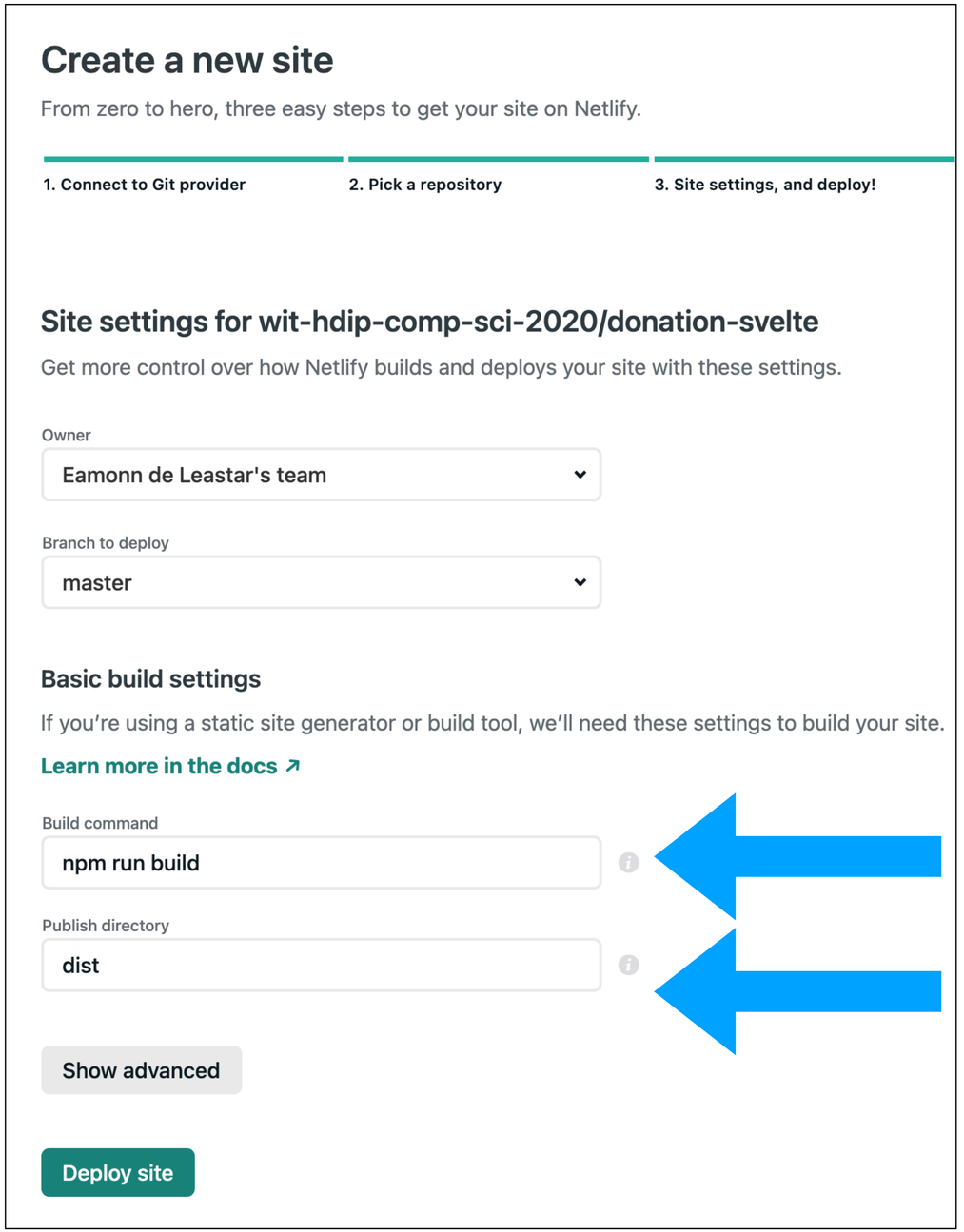
- Drag & Drop **dist** folder to drop zone for project on Netlify project

netlify Platform Pricing Enterprise Jamstack Commu

Modern web apps shipped faster

An intuitive Git-based workflow and powerful serverless platform to build, deploy, and collaborate on web apps

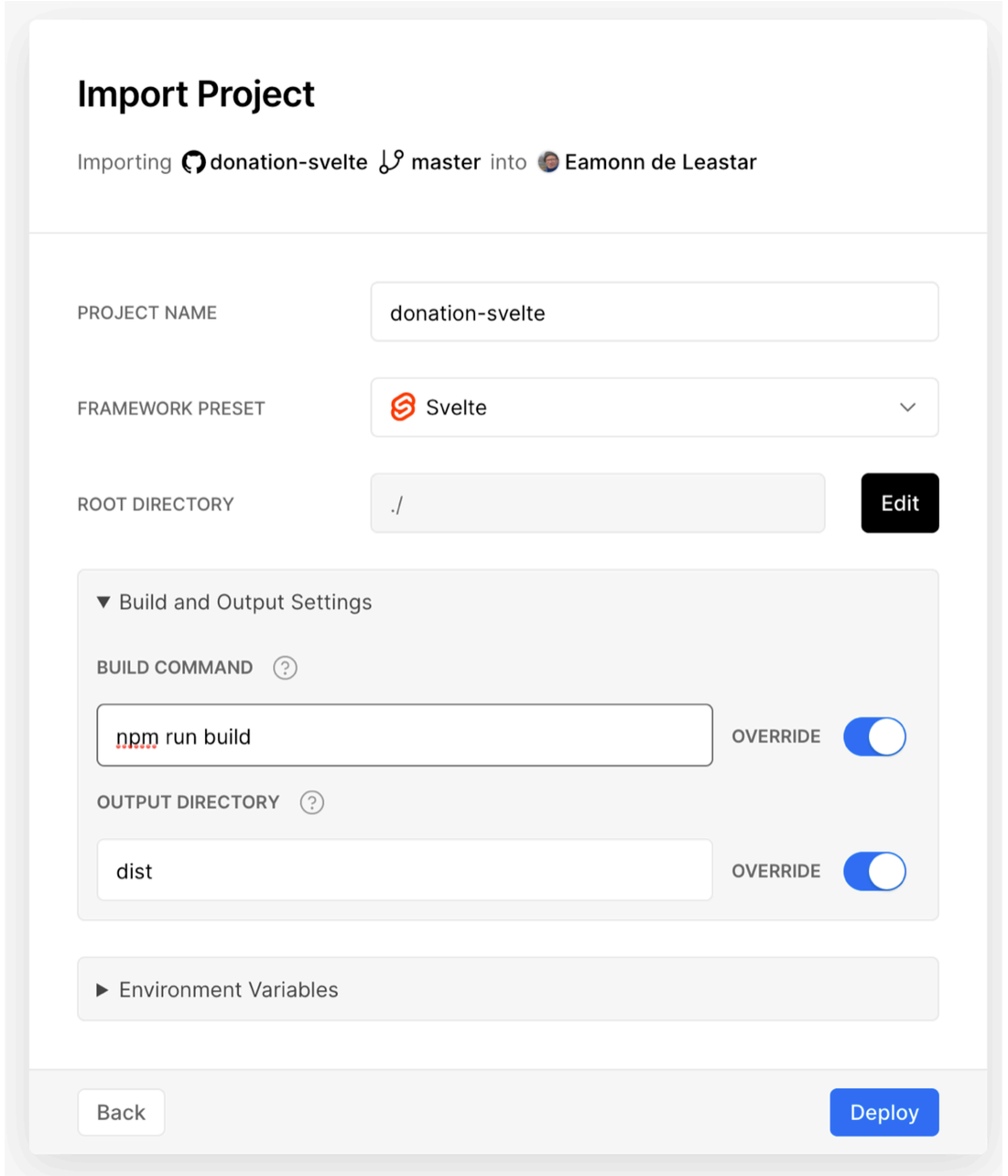
Deploy to Netlify using Github



- Deploy directly from github
- Driven from Netlify web dashboard
- Specify:
  - Build command
  - Build folder

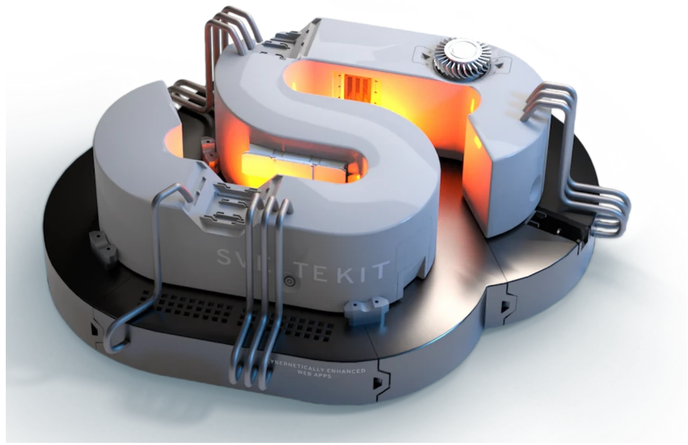
Develop. Preview. Ship.

Deploy to Vercel



- Deploy directly from github
- Driven from Vercel web dashboard
- Specify:
  - Build command
  - Build folder

SvelteKit

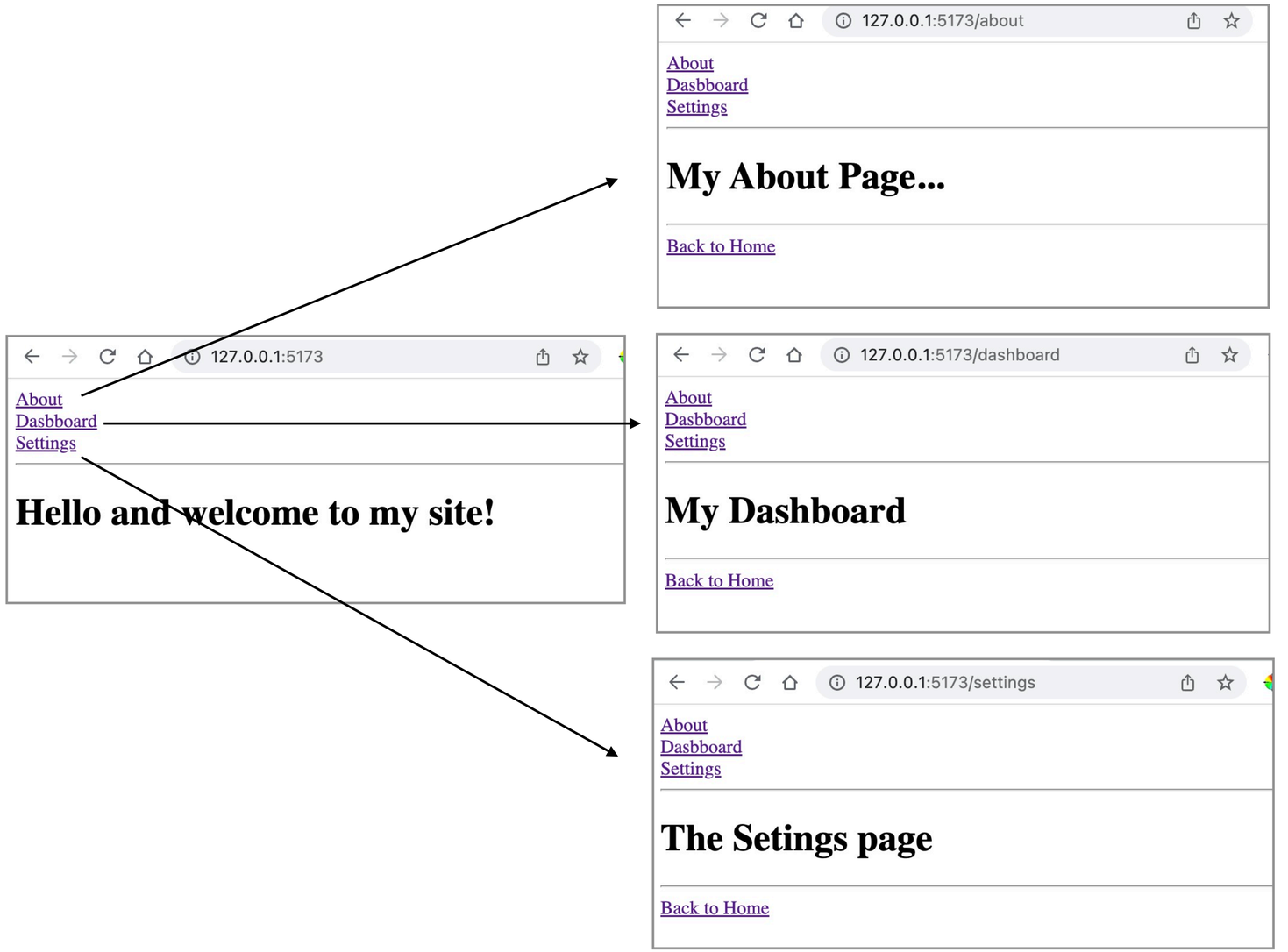


Client Side Rendering (CSR)

- Client-side rendering (CSR) is the generation of the page contents in the web browser using JavaScript.

- Server-side rendering (SSR) is the generation of the page contents on the server.

- SSR is significantly superior for SEO (Search Engine optimisation)



Server Side Rendering (SSR)

CSR

1.

User requests a website

2.

a CDN can quickly serve HTML files with links to JS

3.

Browser downloads the HTML & then the JS, meanwhile the site isn't visible to the user

4.

The browser downloads the Javascripts

5.

The JS is then executed, APIs are called for data, & user sees placeholders

6.

The Server responds with the data asked by the API

7.

The data from the APIs fill the placeholders & the page is now interactive

Infidigit

SSR

1.

User requests a website

2.

Server creates "Ready to Render" HTML files

3.

The browser can quickly render the HTML but the site isn't interactive

4.

The browser downloads the Javascript

5.

The user can view content & the interactions can be recorded

6.

The browser executes the JS framework

7.

The recorded interactions can be executed & the page is now interactive

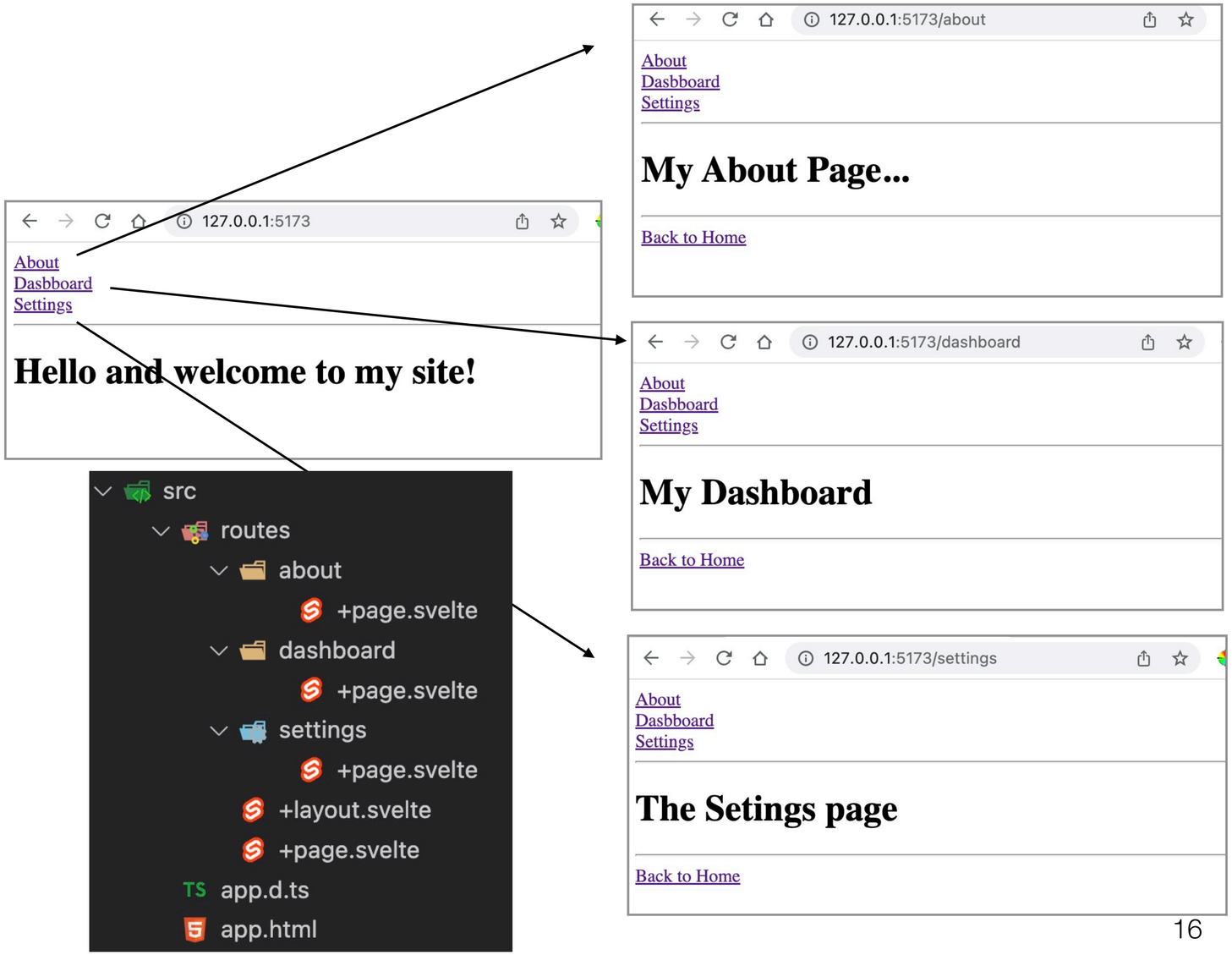
Infidigit

Client Side Routing

- By default, when you navigate to a new page (by clicking on a link or using the browser's forward or back buttons), SvelteKit will intercept the attempted navigation and handle it.

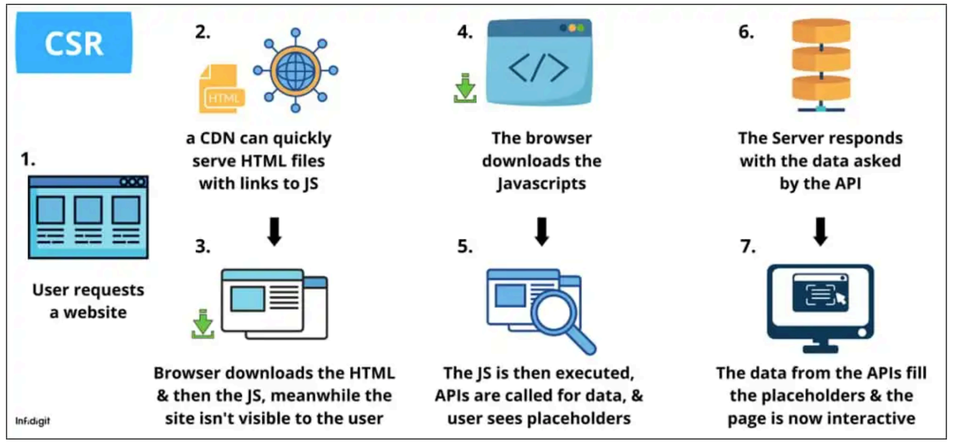
- SvelteKit will then update the displayed contents on the client by rendering the component for the new page

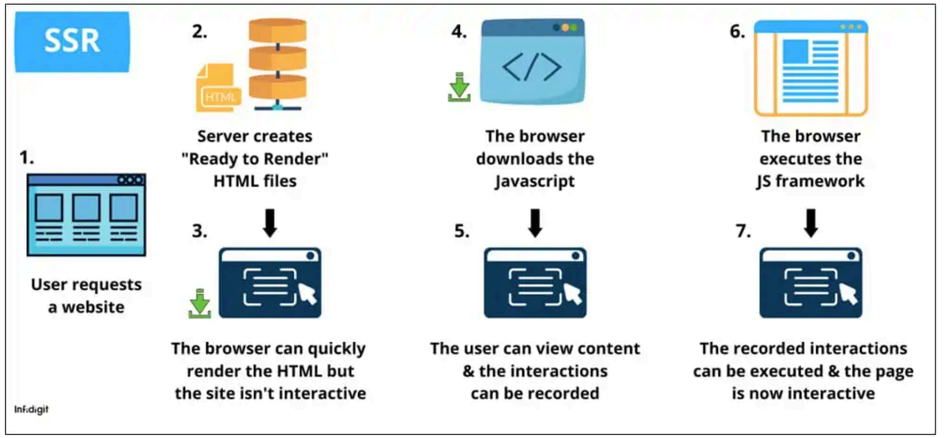
- This process of updating the page on the client in response to attempted navigation is called client-side routing.



SvelteKit Rendering Model

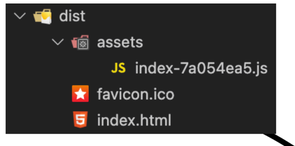
- By default, pages are also server-side (SSR) rendered when you first visit a page.
- When you navigate back to a page that has already been visited, SvelteKit will use client side rendering (CSR)
- This will improve the perceived performance of the app (it will feel like an SPA), but retain potential SEO benefits of an MPA model
- This behaviour can be customised -
  - Full CSR (SPA like)
  - Full SSR (MPA like)
  - Per page behaviour CSR/SSR

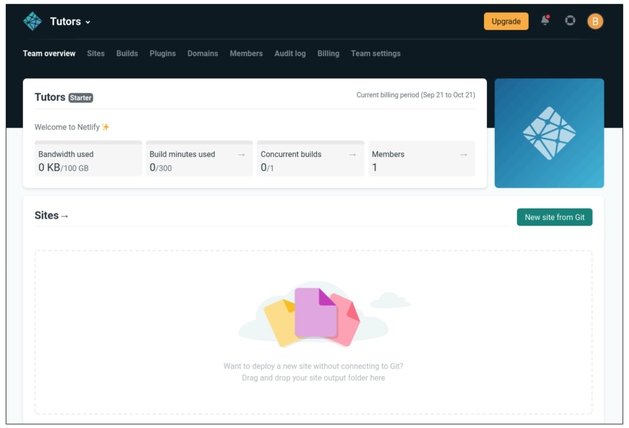




SvelteKit Deployment

- Because of the powerful, hybrid rendering models available, deployment is NOT just a matter of:
  - npm run build
  - Copy build folder to a web server





- Drag & Drop **dist** folder to drop zone for project on Netlify project

- Sveltekit requires an ‘Adapter’ in order to deploy to specific platforms:
- You select and install the adapter for the platform your are deploying to
- This adapter will automatically handle all SSR/CSR configuration

- `@sveltejs/adapter-cloudflare` for Cloudflare Pages
- `@sveltejs/adapter-cloudflare-workers` for Cloudflare Workers
- `@sveltejs/adapter-netlify` for Netlify
- `@sveltejs/adapter-node` for Node servers
- `@sveltejs/adapter-static` for static site generation (SSG)
- `@sveltejs/adapter-vercel` for Vercel

SvelteKit Netlify Deployment

- Install the Netlify Adapter
- Include this adapter in the svelte config
- Include build instructions for netlify
- Create & Import app on Netlify

```
npm install -D @sveltejs/adapter-netlify@next
```

svelte.config.js

```
//import adapter from '@sveltejs/adapter-auto';
import adapter from "@sveltejs/adapter-netlify";
import { vitePreprocess } from "@sveltejs/kit/vite";

/** @type {import('@sveltejs/kit').Config} */
const config = {
  // Consult https://kit.svelte.dev/docs/integrations#preprocessors
  // for more information about preprocessors
  preprocess: vitePreprocess(),

  kit: {
    adapter: adapter()
  }
};

export default config;
```

netlify.toml

```
[build]
  command = "npm run build"
  publish = "build"
```

Deploy Front End Apps

Deploying donation-svelte to Netlify & Vercel services