

Going Static in a Dynamic World with Hasura and Nuxt.js



NUXTJS



Debbie O'Brien



Head of Learning and Developer Advocate
at NuxtJS



Writer @Ultimate Courses



Teacher @Vue School

Organiser @MallorcaJS, Vue Spain



Experts



Debbie O'Brien



@debs_obrien

Nuxt

The Intuitive **Vue** Framework

Nuxt Benefits

"file-based" routing

Automatic Code splitting of routes

Prefetching

more than 50 modules

Why Static?

No server: cheaper, safer

Better offline support

Better performance

Reduces Carbon footprint

Nuxt Static Module

You don't have to **expose** your **API** to the public

Decide when to publish **new content** from your API by **re-generating** your website

Fully static is much **faster**

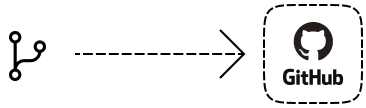
API calls have latency - always takes longer

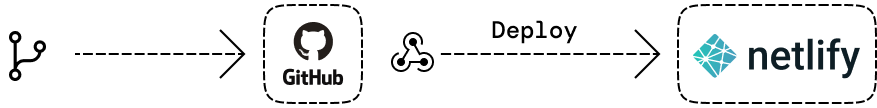
Payload files are **cached** and **offline** support with PWA module

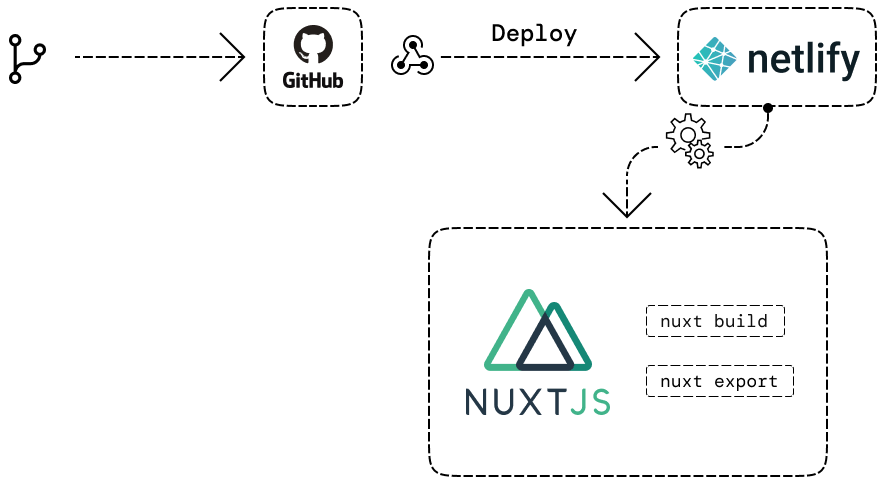
Full Static

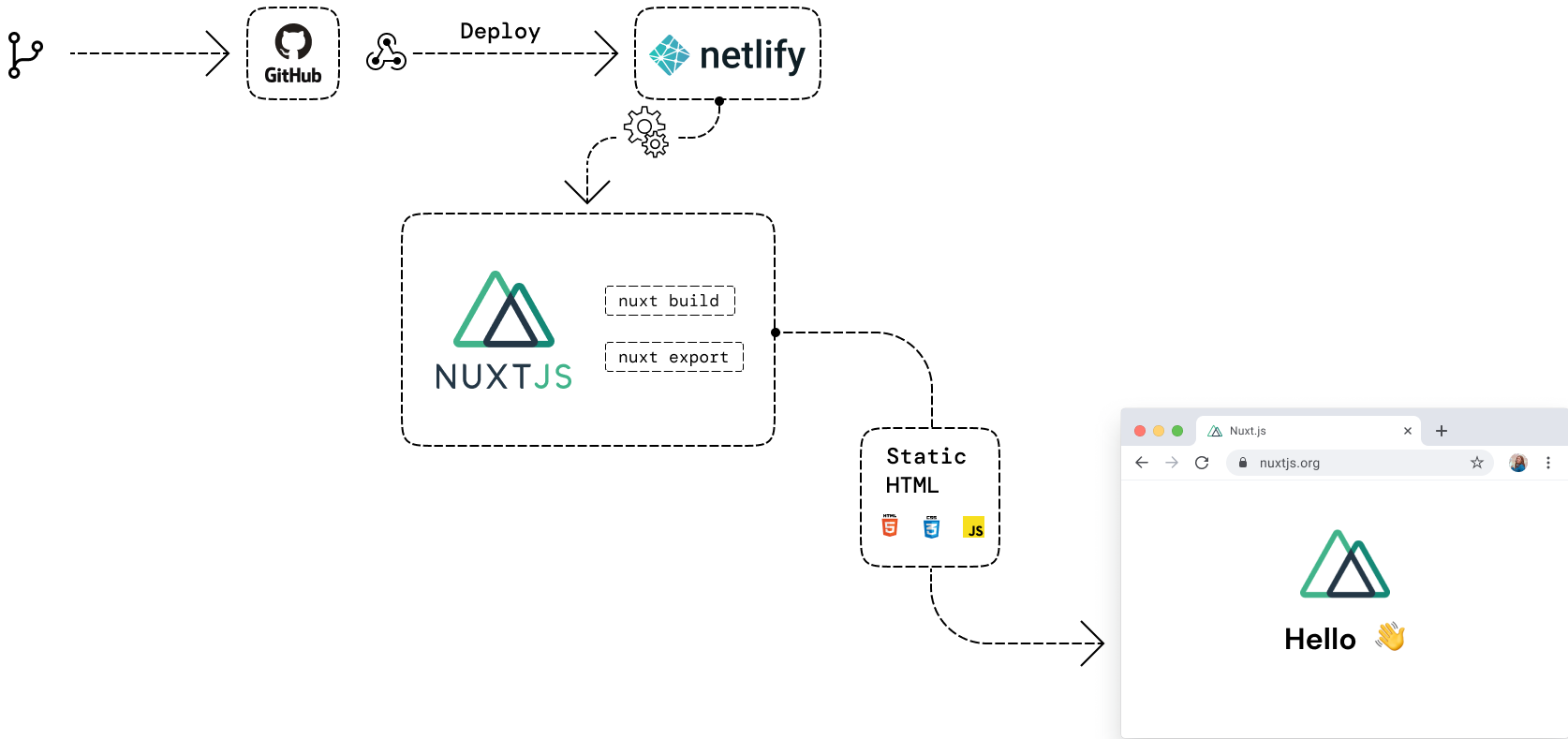
target: 'static',

nuxt build && nuxt export









Data

CONFERENCES

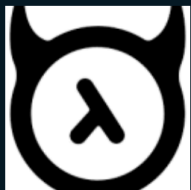
All

Speaker

Lightening Talk

MC

Attendee



10

June
2020

HasuraConf Virtual, Virtual

Going static in a dynamic world with Nuxt and Haura

Speaker



12

May
2020

Dev Around The Sun Virtual

Static Sites === great performance

Speaker



11

May
2020

Vue Zurich Meetup Virtual, Switzerland

Static Sites === great performance

Speaker



7

May
2020

JSVidCon Virtual

Static Sites === great performance

Speaker



Get realtime GraphQL APIs *instantly*

Hasura is an [open source](#) engine that connects to your databases & microservices and auto-generates a production-ready GraphQL backend.

Install and register http

```
npm install @nuxt/http
```

```
modules: [  
  '@nuxt/http'  
],
```


Create a plugin

```
export default function ({ $http, env }, inject) {  
  const $hasura = $http.create({  
    prefixUrl: env.API_HASURA_URL  
  })  
  
  inject('hasura', $hasura.$post.bind($hasura, ''))  
}
```

Register the Plugin

```
plugins: [  
  '@plugins/hasura.js'  
],
```

Import

```
import gql from 'graphql-tag'
```

```
import { print } from 'graphql/language/printer'
```

Add your Query

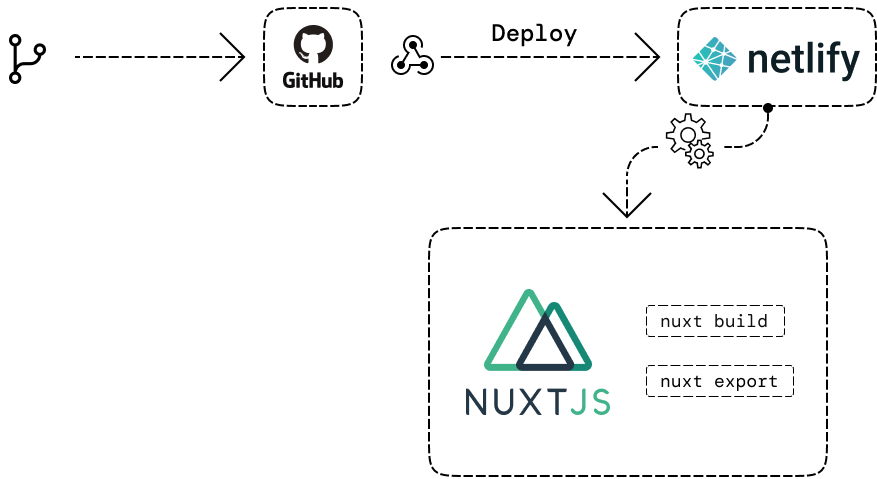
```
const QUERY = gql`  
  query {  
    conferences(order_by: { date: desc }) {  
      name  
      place  
      date  
    }  
  }  
`
```

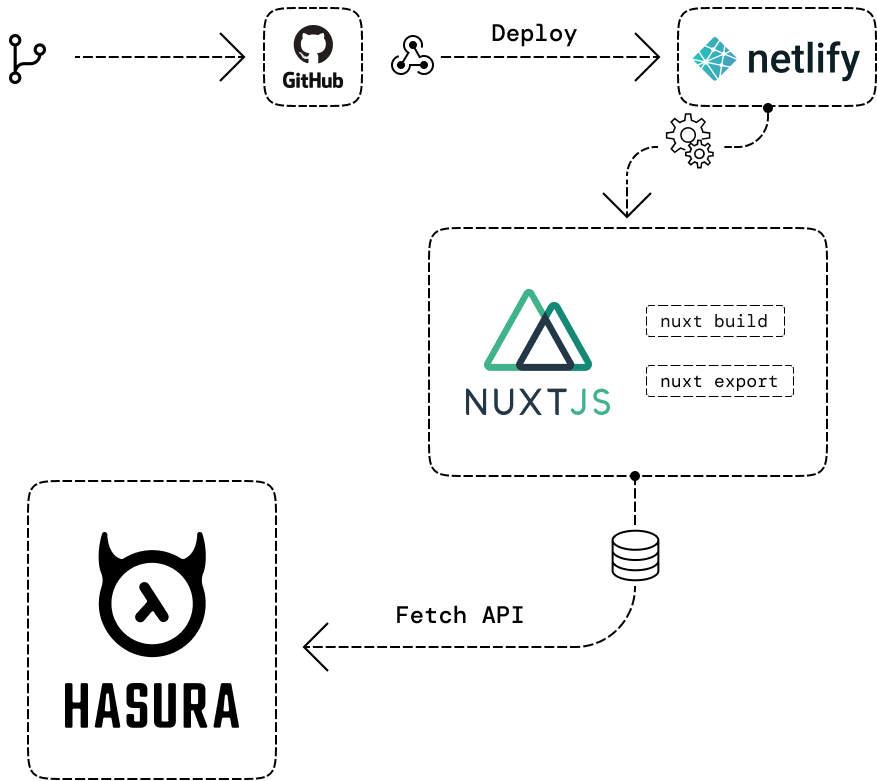
AsyncData

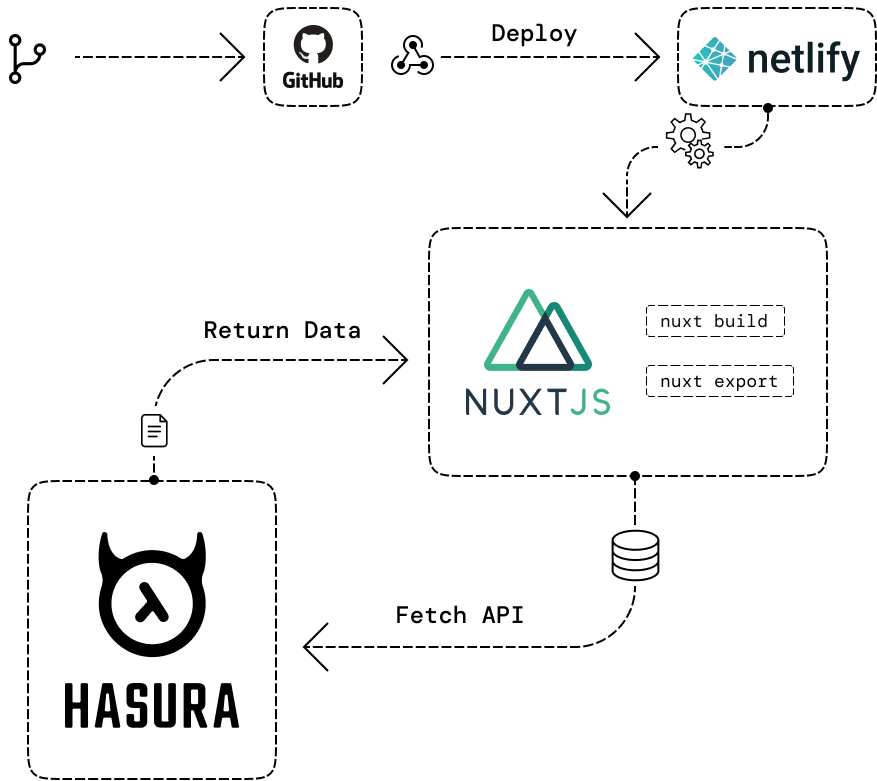
```
async asyncData({ app }) {  
  const { data } = await app.$hasura({  
    query: print(QUERY)  
  })  
  return {  
    conferences: data.conferences  
  }  
},
```

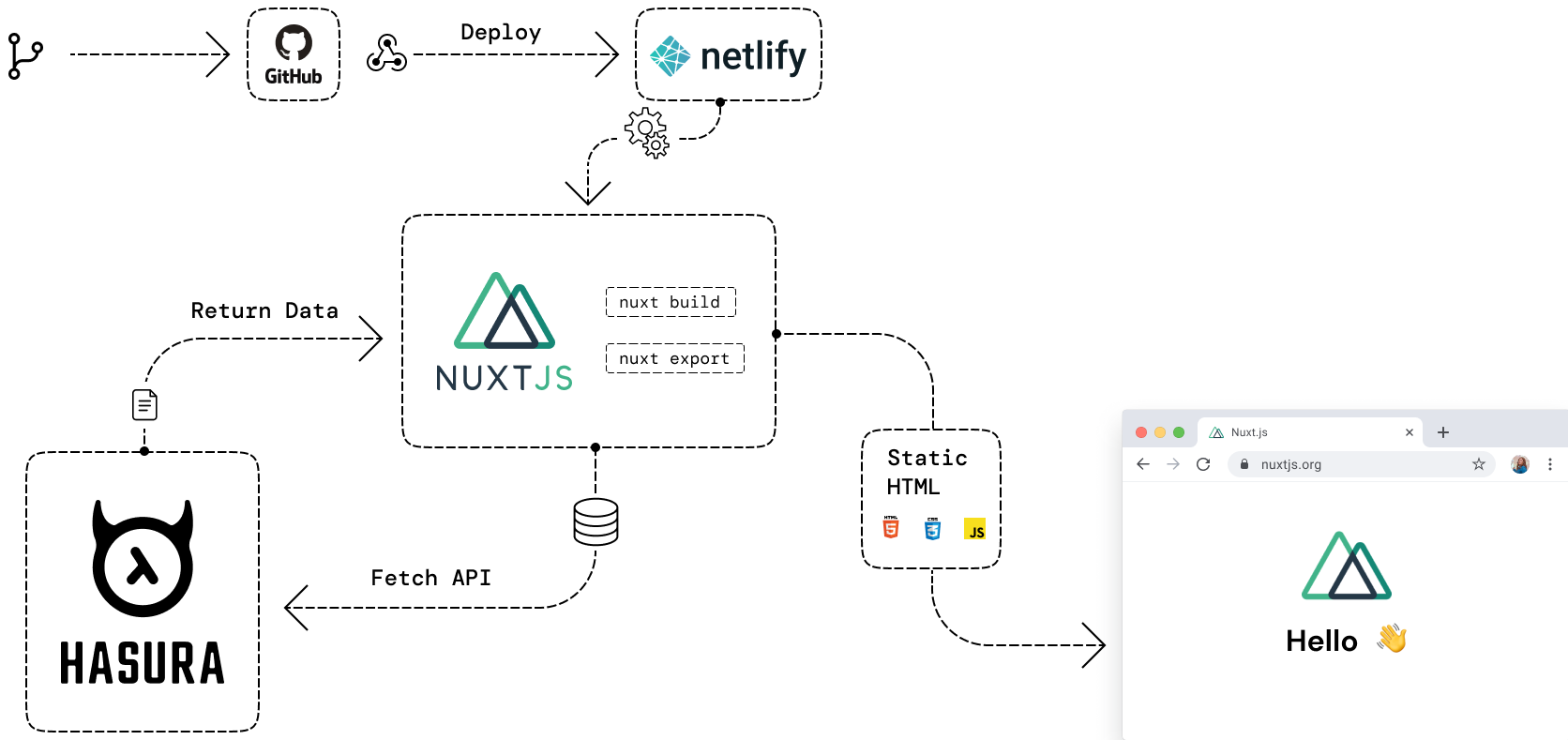
Use in your template

```
<ul v-for="conference in conferences">  
  <li> {{conference.name}}</li>  
</ul>
```



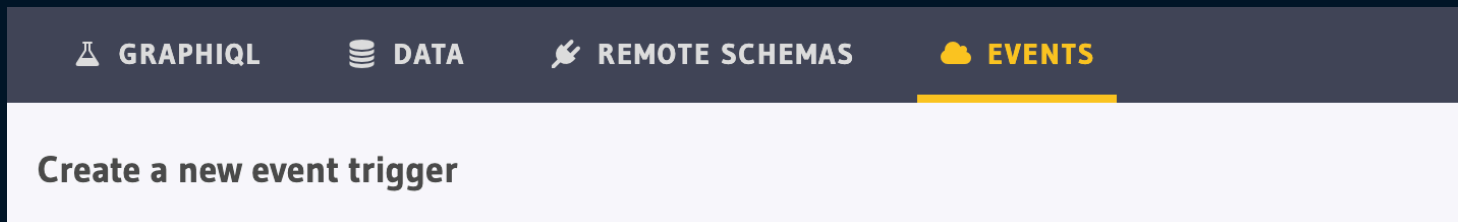






What happens when my **data changes**

Generate on Update



Create your trigger

trigger_name

Schema/Table ?

public ▾ Select table ▾

Trigger Operations ?

Insert Update Delete Via console ? [\(Know more\)](#)

Webhook URL ?

URL ▾ http://httpbin.org/post

Netlify Build Hooks

Build hooks

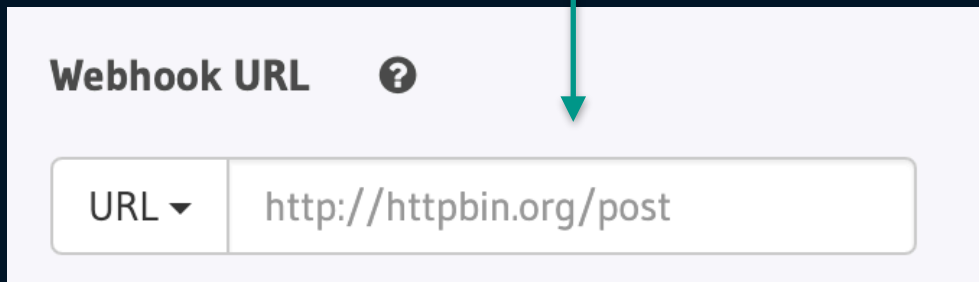
Build hooks give you a unique URL you can use to trigger a build.

Branch to build

[Learn more about build hooks in the docs ↗](#)

Add the URL to Hasura


`https://api.netlify.com/build_hooks/5e4c190828324653346571ad`




Webhook URL ?

URL ▼

Browse, Modify, Delete

 HASURA v1.0.0

[GRAPHIQL](#) **[DATA](#)** [REMOTE SCHEMAS](#) [EVENTS](#) [SECURE YOUR ENDPOINT](#) 

Schema:

Search table/view/function

Tables (2)

- conferences
- workshops**

SQL

You are here: [Data](#) > [Schema](#) > [public](#) > [workshops](#) > Browse Rows

workshops





















[Browse Rows \(9\)](#) [Insert Row](#) [Modify](#) [Relationships](#) [Permissions](#)

Filter

-- column -- -- op -- -- value --

Sort

-- column -- --

	◀ id	◀ name	◀ title
   	0516ee9c-1fb7-44b1-b732-25aaed29ff92	Marion Schleifer	Modern and Serverless Web Applications with GraphQL
   	c480c841-1da9-49a3-a489-99ad9da06738	Rolf Haug & Alex Kyriakidis	Scaling applications with Vuex
   	a583ac9b-386a-4a0e-8984-538fea258e15	Swizec Teller	Build and launch a modern JAMstack web app with React, Gra
   	798f9f63-51de-4325-8f40-50f53a26a65c	Evan You	Deep Dive with Vue 3.0
   	16679bef-c684-4c84-b4c6-3a0af720a18b	Harry Roberts	Frontend Performance Building Faster Websites



HASURA

Add new data

My awesome new title 🌟

Save



Deploy

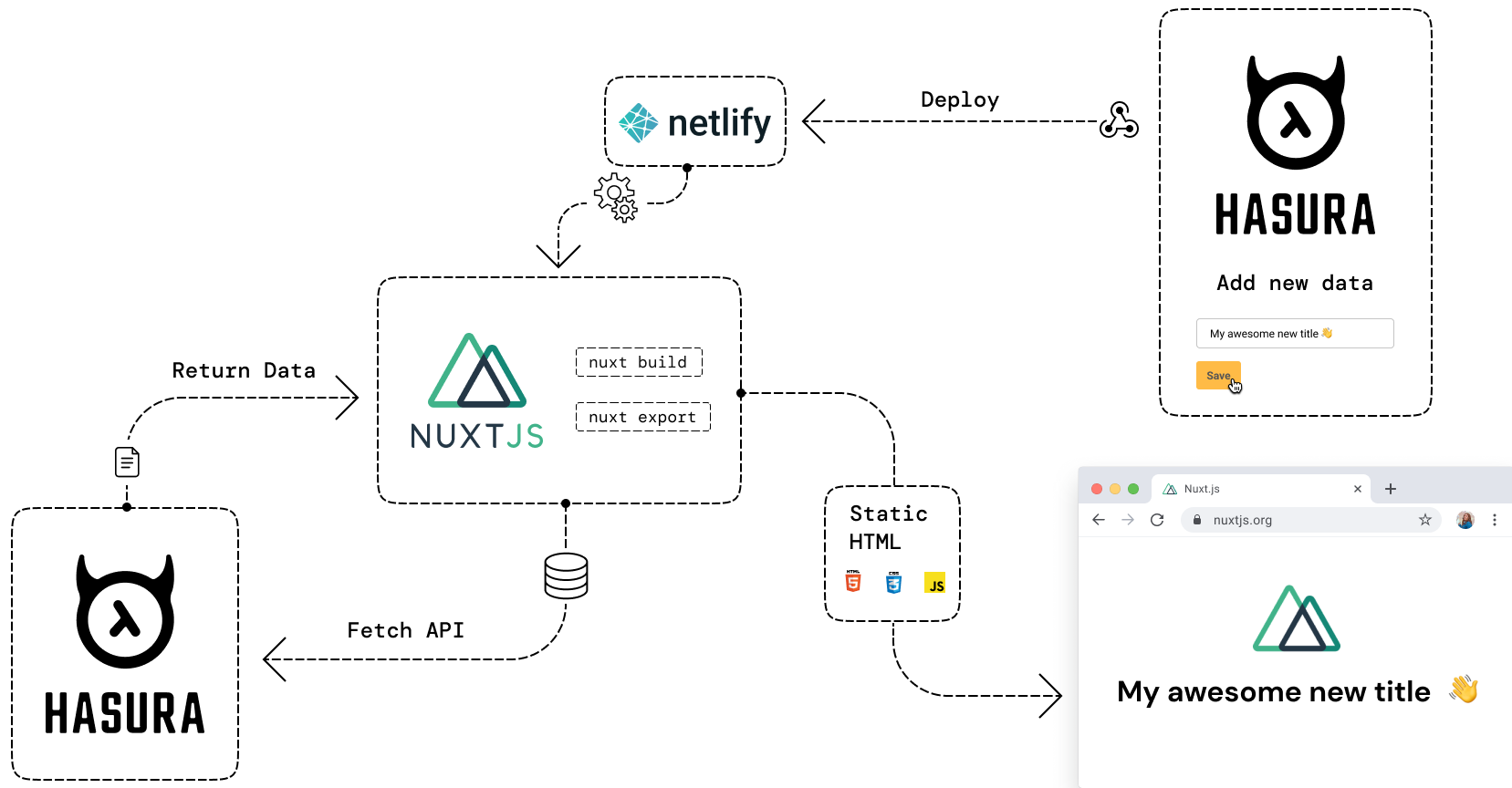


HASURA

Add new data

My awesome new title 🌟

Save



Nuxt Static Module

~~Call the **API** on page change~~

Call the **payload** on page change



Why Static Sites

Why Static?

No server: cheaper, safer

Better offline support

Better performance

Reduces Carbon footprint

Think Static
be Dynamic

Are you
NUXT?

Thank you

Debbie O'Brien



**Head of Learning and
Developer Advocate**

Open Source Contributor

