

爆速サイトを爆速で作れる Netlify + Gatsby

2018.04.24 Roppongi.js #2 - @mottox2



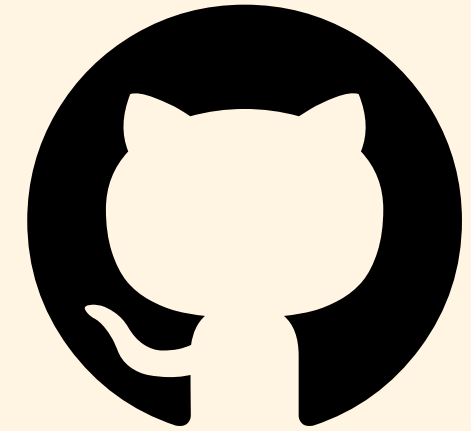
@mottox2

Freelance web engineer in Roppongi

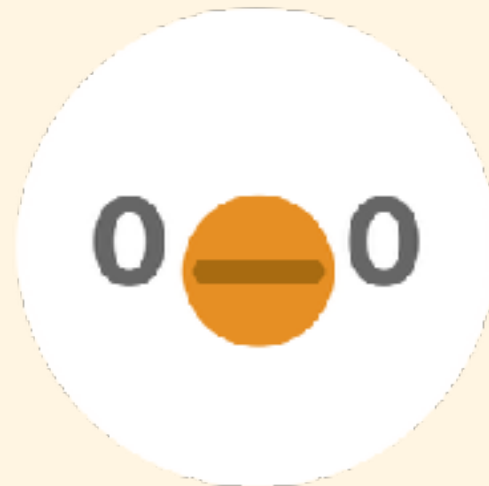
こんなもの作った

<http://gatsby-starter-esa.netlify.com/>

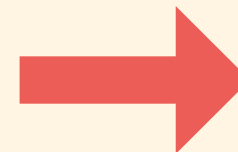
ドキュメントが更新されたら静的サイトが吐き出されるやつ。(所要時間: 4 時間)



2. Pull

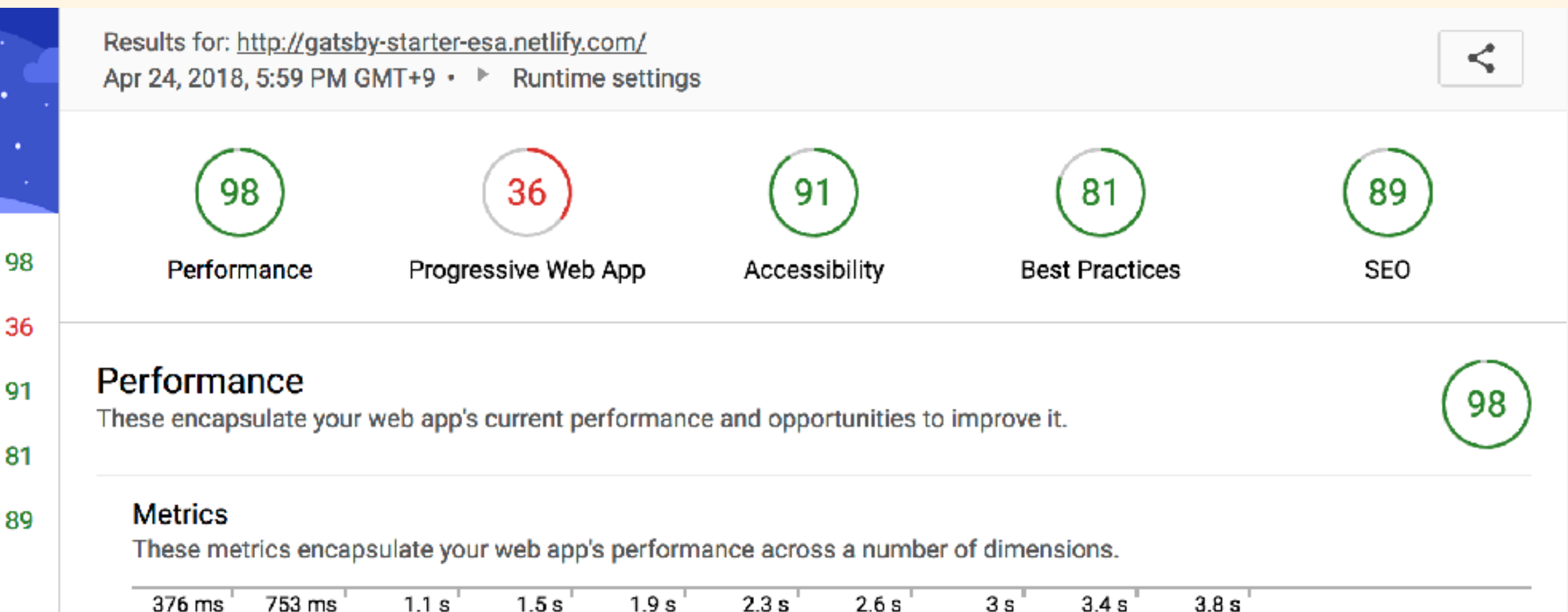


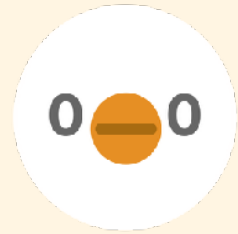
1. Webhook



3. Build & Deploy

それより見てくれ！速いぞ！



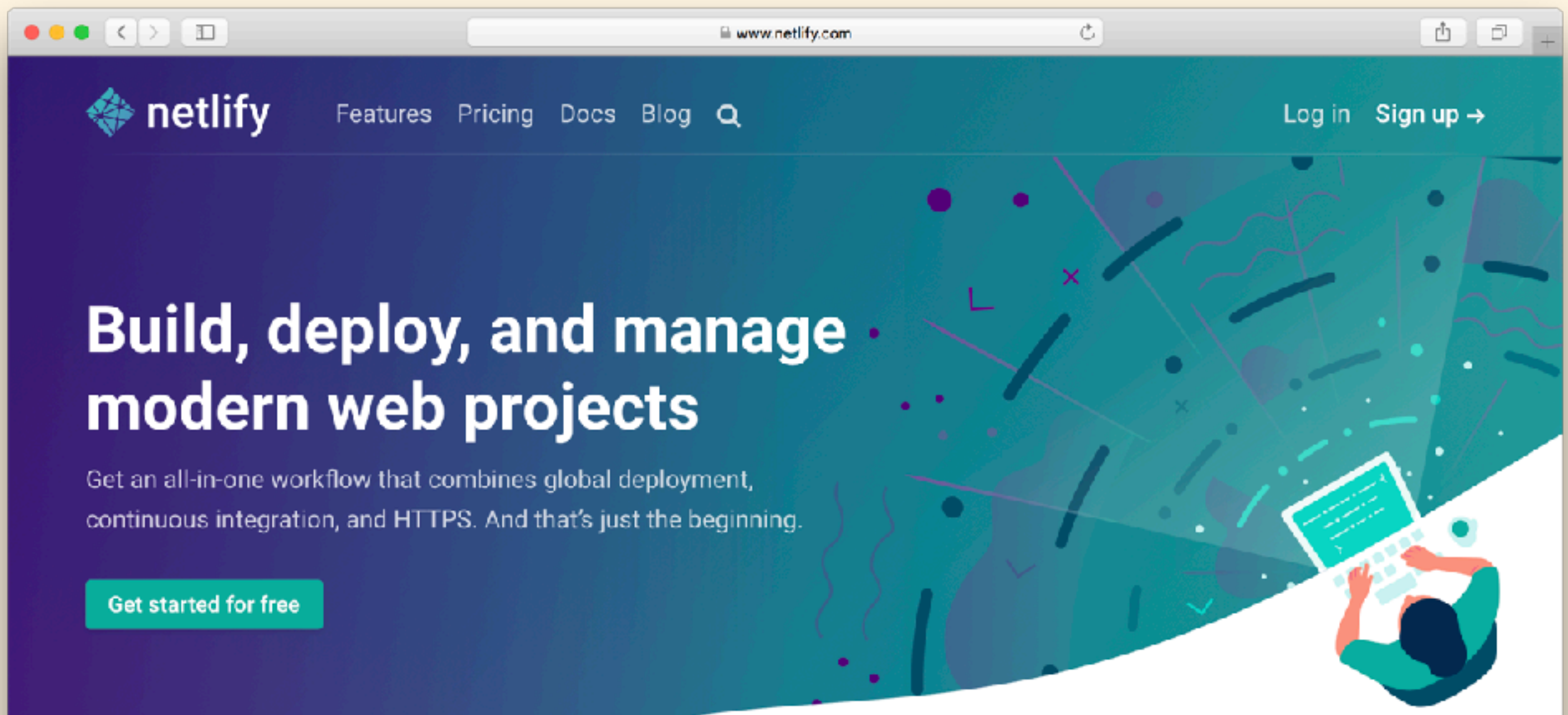


esa.io <https://esa.io/>

- ドキュメント共有サービス
e.g. Qiita:Team, Kibela
- WebhookやWeb APIが用意されている
- 一人で使ってる



- 静的サイトのホスティングサービス。

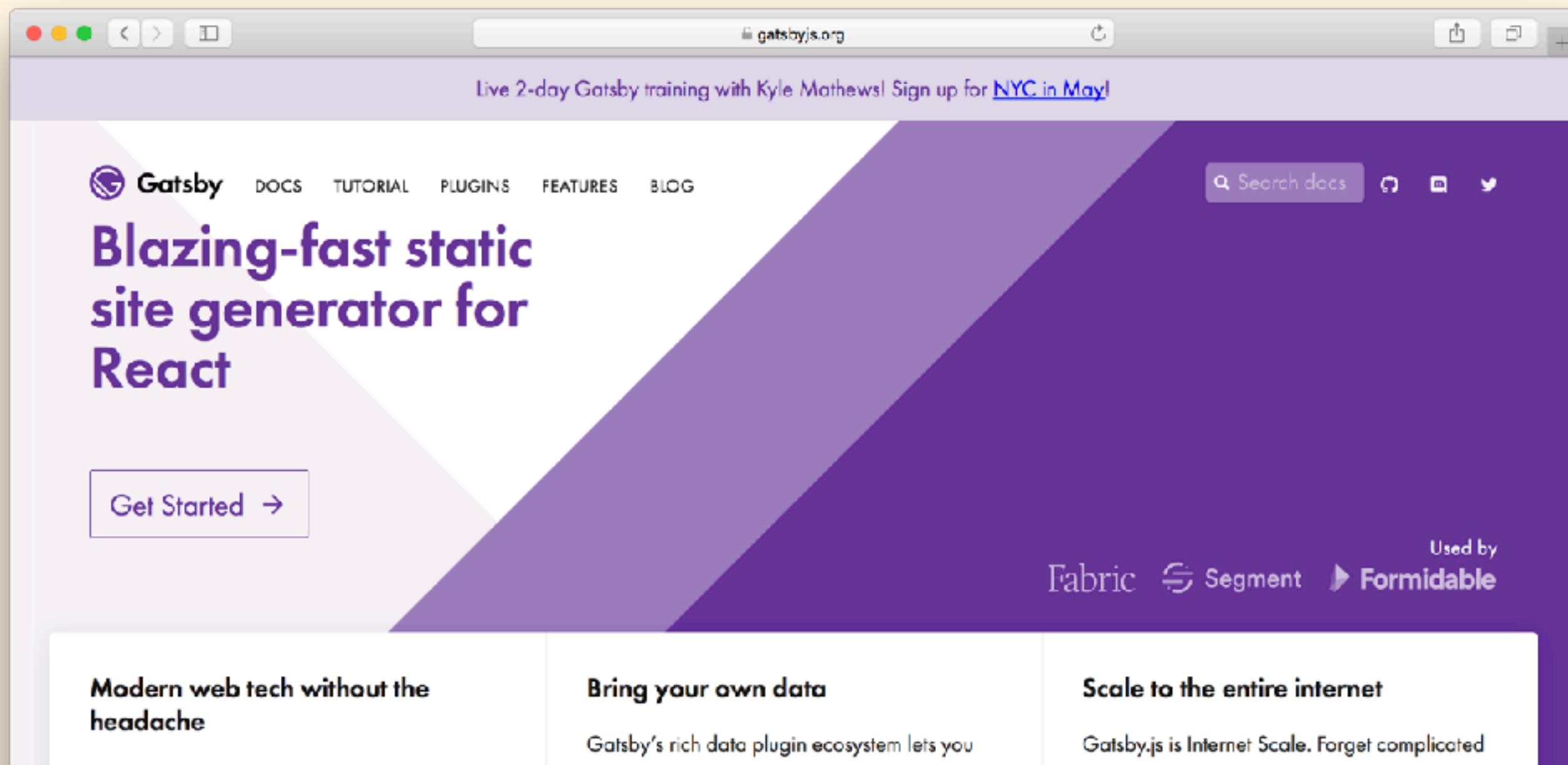




- 静的サイトのホスティングサービス。
- GithubにcommitするだけでDeploy
- POSTリクエストでDeploy(Build hook)
- 自分でサーバーの面倒見なくてよい



- めっちゃ高速な静的サイトジェネレーター





- めっちゃ高速な静的サイトジェネレーター
 - reactjs.orgもGatsby製
 - JSだけで完結する
- Preload, Inline CSS, React SPAがデフォルトで動くのでめっちゃ速い（曰く、Blazing-fast）

How Gatsby works

StaticFile生成までのイメージ

Source

Markdown, esa.io, medium

View

React

Build

>

Static Files

Source

- source-pluginを入れGraphQLを使って呼び出す
 - WordPress, Markdown, その他Web API...
 - esa.ioのpluginはなかったので自作した
 - <https://github.com/mottox2/gatsby-source-esa>

View

- Reactで作れる。
- GraphQLでSourceの呼び出し

```
import React from 'react'
import Helmet from 'react-helmet'
import Link from 'gatsby-link'

class PostTemplate extends React.Component {
  render() {
    const post = this.props.data.esaPost

    return (
      <div>
        <Helmet title={` ${post.name} `} />
        <h1>{post.name}</h1>
        <p>{post.updated_at}</p>
        <div dangerouslySetInnerHTML={{ __html: post.
        </div>
      )
    )
  }
}

export default PostTemplate

export const pageQuery = graphql`
  query PostByNumber($number: Int!) {
    esaPost(number: { eq: $number }) {
      number
      name
      wip
      body_html
      updated_at
    }
  }
`
```

Build

- 高速化を狙ったファイル分割
- ビルドは早くない。
- 自前で用意しようとは思えないので最高

```
public
├── 404
├── offline-plugin-app-shell-fallback
├── posts
├── static
├── 404.html
├── app-d7d853cc7da145a4b6b1.js
├── app-d7d853cc7da145a4b6b1.js.map
├── app-ef07caf7463d64a75805.js
├── app-ef07caf7463d64a75805.js.map
├── chunk-manifest.json
├── commons-8d23a67706fd3e385a4d.js
├── commons-8d23a67706fd3e385a4d.js.map
├── component---node-modules-gatsby-plugin-offline-app-shell-js-...
├── component---node-modules-gatsby-plugin-offline-app-shell-js-...
├── component---src-layouts-index-js-20400e78f34381b6ba58.js
├── component---src-layouts-index-js-20400e78f34381b6ba58.js.m...
├── component---src-pages-404-js-4503918ea3a16cfcdb75.js
├── component---src-pages-404-js-4503918ea3a16cfcdb75.js.map
├── component---src-pages-index-js-2e8d5514cb4321ae10e7.js
├── component---src-pages-index-js-2e8d5514cb4321ae10e7.js.map
├── component---src-templates-post-js-23592ea51183a2cd1e2b.js
├── component---src-templates-post-js-23592ea51183a2cd1e2b.js....
├── ★ favicon.ico
├── index.html
├── path----557518bd178906f8d58a.js
├── path----557518bd178906f8d58a.js.map
├── path---404-a0e39f21c11f6a62c5ab.js
├── path---404-a0e39f21c11f6a62c5ab.js.map
├── path---404-html-a0e39f21c11f6a62c5ab.js
├── path---404-html-a0e39f21c11f6a62c5ab.js.map
├── path---index-46ecbc45a03104c36463.js
├── path---index-46ecbc45a03104c36463.js.map
├── path---index-10063ce2c9b40d645d90.js
├── path---index-10063ce2c9b40d645d90.js.map
├── path---offline-plugin-app-shell-fallback-a0e39f21c11f6a62c5ab.js
├── path---offline-plugin-app-shell-fallback-a0e39f21c11f6a62c5ab....
├── path---posts-107-54578e66a7d68e9d6d81.js
```

Publish

- Github PageやNetlifyなどにデプロイしている例が多い
- それほど難しいことをしていないのに高速なサイトが作れる。

In Real World

Wordpressのフロントで使いたい

- WordPressのフロントエンドを分離したい気持ち強い。
- ただ更新の度にビルドが必要になるので更新頻度の高いサイトには向かない
- また、DataSourceが巨大になるほどビルドが遅くなっていく。

個人で使うのはGood Choice

- 「更新頻度が高くない、記事数もそれほど」というサイトには向いている。
- ポートフォリオ・個人ブログ・ドキュメンテーションなどに相性がよい。
- 作るだけならテンプレートが転がってるので30分あれば公開まで出来ます。

Thanks