# Gatsby, Contentful, and Netlify Walk Into a Bar

Dave Kensell

> Static site generators like Gatsby are incredibly developer friendly and make super fast sites, but they're almost completely inaccessible to non-developers. In this blog post I'll show you how to change that.

CMS exist to give content creators like graphic designers, copywriters, social media managers, and the rest of the marketing team a way to quickly get content online themselves.

But most of them are designed to be all-in-one website solutions, like WordPress. Definitely not static.

So how do you marry a CMS to a static site generator?

Enter Contentful, a headless CMS. A headless CMS gives your non-devs a familiar interface for managing content that is then connected to your site generator by API. Contentful has a nice starter pack for Gatsby which we'll get set up.

And then we'll go even further and show you how content creators can build and deploy the site at the click of a button, using Netlify webhooks.

## Install Gatsby CLI

This post assumes you're already familiar with Gatsby. If you're not, you may want to visit first steps with Gatsby and then return.

First, if you don't already have the gatsby-cli installed, do so:

```
$ yarn global add gatsby-cli
```

## Create New Contentful-Starter

Now it's time to create a new project using the Contentful starter pack.

A starter creates a ready-made Gatsby project with configuration, design elements, and placeholder content for a site.

```
$ gatsby new contentful-starter
```

A successful install delivers instructions on the next setup step for importing a content model and creating a config file `./.contentful.json"`.

### *Setup: connect your space*

Enter your newly created project to follow the instructions:

```
$ cd contentful-starter
$ yarn run setup
```

You're going to be prompted to input your `Space ID` and API access tokens. So let's go get those.

Go to Contentful and create an empty `space`. This is basically a content repository for your website (and wherever else you want that content to go). At the time of this writing, you can have two free spaces on the Basic tier.

Go to `Settings` > `General Settings` to get your Space ID and input it to the terminal.

Go to `Settings` > `API Keys` > `Content Management tokens` to generate your personal access token and input it to the terminal.

Switch over to the `Content delivery / preview tokens` tab within the same window. Press the `Add API key` button. Copy your delivery API token and input it to the terminal.

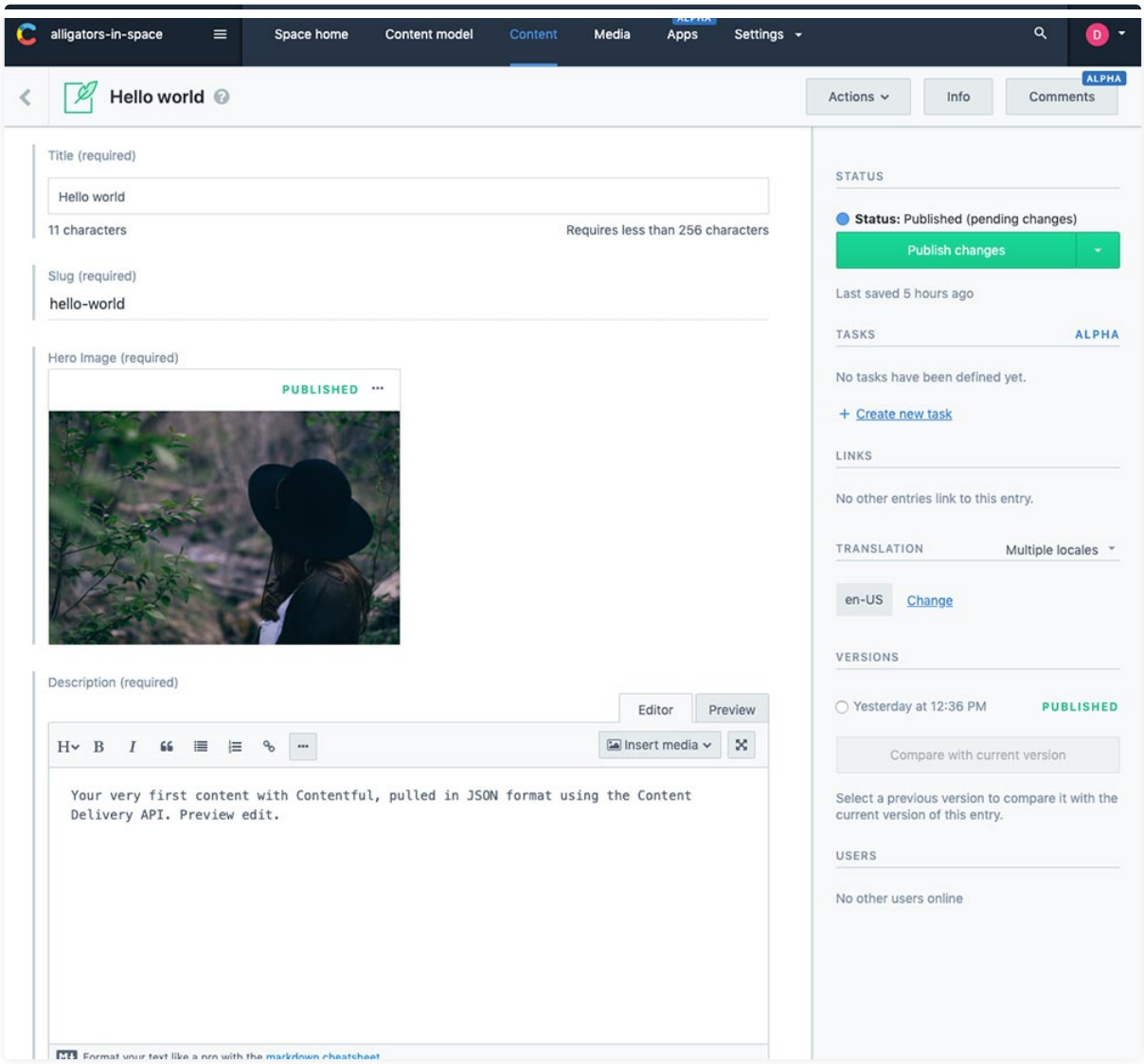
The setup should now run and write dev and prod `.env` config files for you, as well as import starter content and settings into your space.

Your local tree should look like this, 2 levels deep:

```
.
├── LICENSE
├── README.md
├── app.json
├── bin
|   ├── hello.js
|   └── setup.js
```

```
├── contentful
│   └── export.json
├── gatsby-config.js
├── gatsby-node.js
├── package.json
├── public
│   ├── avenir-400.woff2
│   ├── favicon.ico
│   ├── index.html
│   ├── page-data
│   ├── robots.txt
│   └── static
├── screenshot.jpg
├── setup.jpg
├── src
│   ├── components
│   ├── pages
│   └── templates
├── static
│   ├── avenir-400.woff2
│   ├── favicon.ico
│   └── robots.txt
├── static.json
└── yarn.lock
```

Your Contentful users can now create and edit content in an uber familiar CMS interface with WYSIWYG.



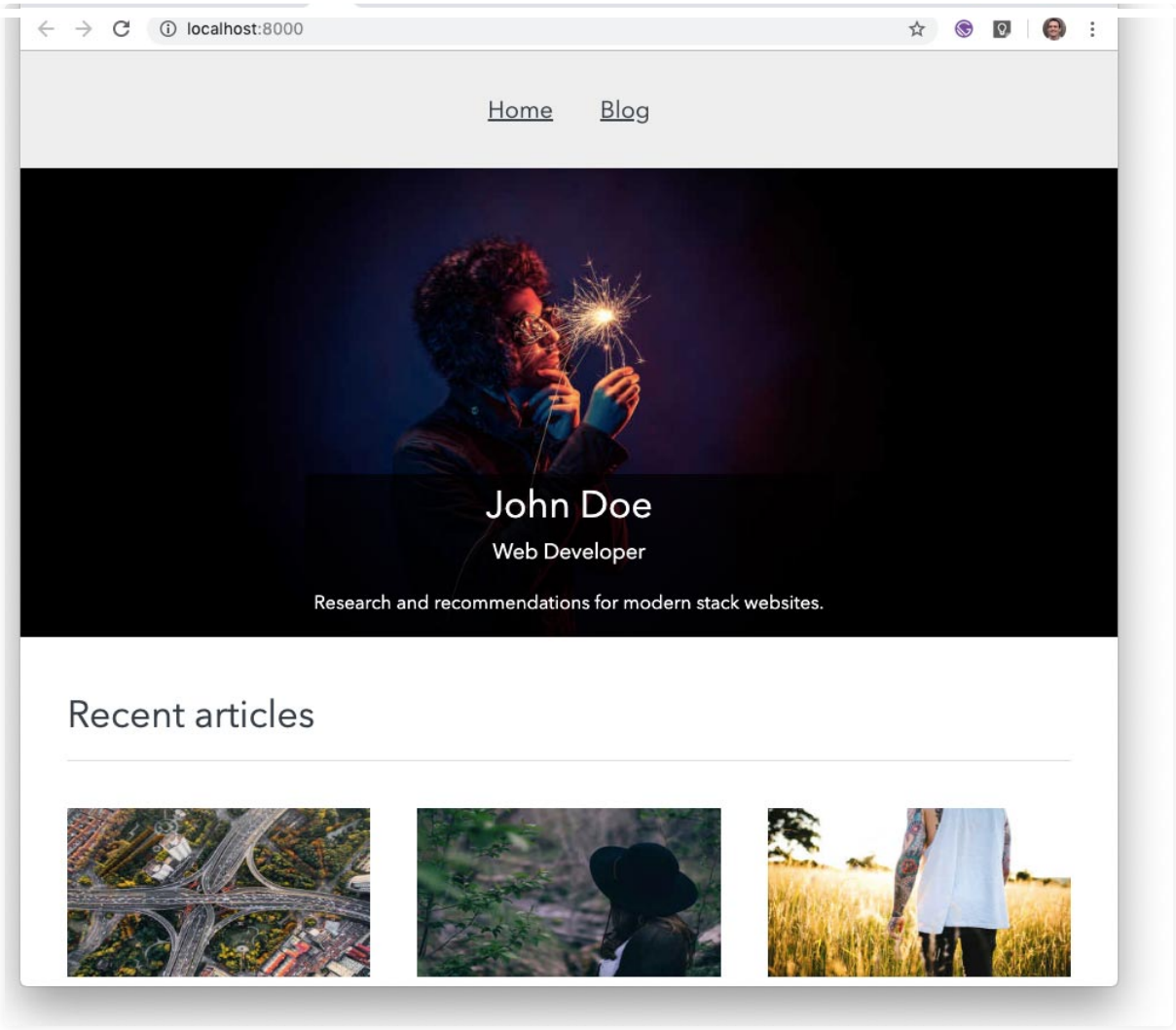

## *Local Dev*

You can run the following command to preview the site on your localhost:

```
$ yarn run dev
```

View it in your browser at http://localhost:8000/.

And view `GraphiQL, an in-browser IDE, to explore your site's data and schema` at http://localhost:8000/___graphql.




Now you can edit and publish content via the Contentful CMS and see it reflected in your local preview.

## *Preview Published or Unpublished Changes Locally*

By default, only content that is `published` via Contentful will be shown on your dev server.

That's probably how you want it, assuming the content creator is the one who's determining whether content is or isn't ready, and you the developer simply want to make sure those changes haven't broken the site.

But if you're the content creator or for some other reason you'd like to preview unpublished content, i.e. `pending changes` or `drafts`, you need to make a couple changes to your dotenv settings.

Take a look at your gatsby-config.js, and you'll see this:

```
require('dotenv').config({
  path: `.env.${process.env.NODE_ENV}`
})
```

```
const contentfulConfig = {
  spaceId: process.env.CONTENTFUL_SPACE_ID,
  accessToken: process.env.CONTENTFUL_ACCESS_TOKEN,
  host: process.env.CONTENTFUL_HOST
}
```

If you actually check your `.env.development` and `.env.production` files, you'll see that `CONTENTFUL_HOST` isn't defined. By default, it will pull from published content.

To change that, add:

```
CONTENTFUL_HOST='preview.contentful.com'
```

To your .env.development file. You also need to change your access token to use the Content Preview API - access token. Get that from within Contentful > Settings > API Keys.

Your .env.development file should now look something like:

```
CONTENTFUL_SPACE_ID='asdf1234'
CONTENTFUL_ACCESS_TOKEN='fasdfdafsfdsafdsa253523523f'
CONTENTFUL_HOST='preview.contentful.com'
```

## *Contentful & Gatsby Preview*

Where it starts to get exciting is with the ability for Contentful users to see live preview changes. If you're storing your source on GitHub, using Contentful and Gatsby allows you to take advantage of Gatsby Preview. At the time of this writing, it's $50/mo for 1 site with a 14 day free trial.

In Gatsby's words, Gatsby Preview provides a (shareable!) temporary URL for viewing changes immediately and in context — so you can make sure that new header plays nicely with the rest of the page before hitting "publish."

Contentful has developed a sidebar preview tool using Gatsby Preview.

But we'll chew on Gatsby Preview elsewhere.

Because we can go a step further and let content creators actually build and deploy.

## Continuous Deployment with Netlify

Once you've got your site how you want it, you can build and deploy to whatever host you'd like. The starter pack is set up to use GitHub Pages, but we're going to use Netlify instead. You'll see why in a minute.

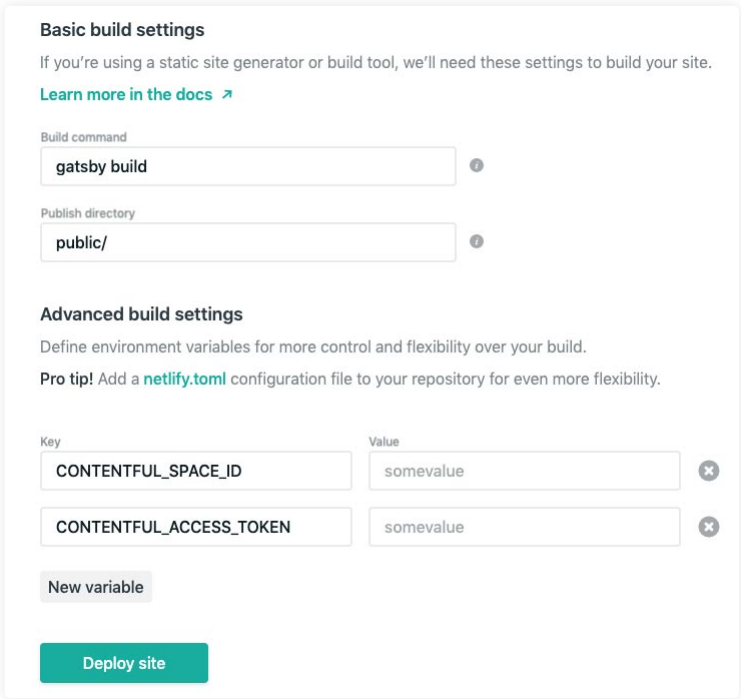
If you haven't already, commit your changes and add your remote at GitHub:

```
$ git remote add origin <url>
```

### Connect GitHub and Netlify

Go to Netlify and create a free account. If you already have one, select `New site from Git`. Choose `GitHub` as your source code host.

You'll be prompted to connect the Netlify app at GitHub, or to adjust authorization settings if you've connected sites in the past. Do so, and then select your contentful-starter repository to link your site.




Follow the instructions. Adjust `gatsby build` to `yarn build` in `Basic build settings.` It may not be necessary for you, but it solved a problem I ran into.

- `Build command` : `yarn build`
- `Publish directory` : `public/`

Press the `Show Advanced` button in order to define env variables. We need to put in our .env.production contentful info here.

```
CONTENTFUL_SPACE_ID='your-space-id'
CONTENTFUL_ACCESS_TOKEN='your-access-token'
```

Now click the `Deploy Site` button.

That should work! When it's done, you'll get a URL you can use as-is or set up a custom domain.

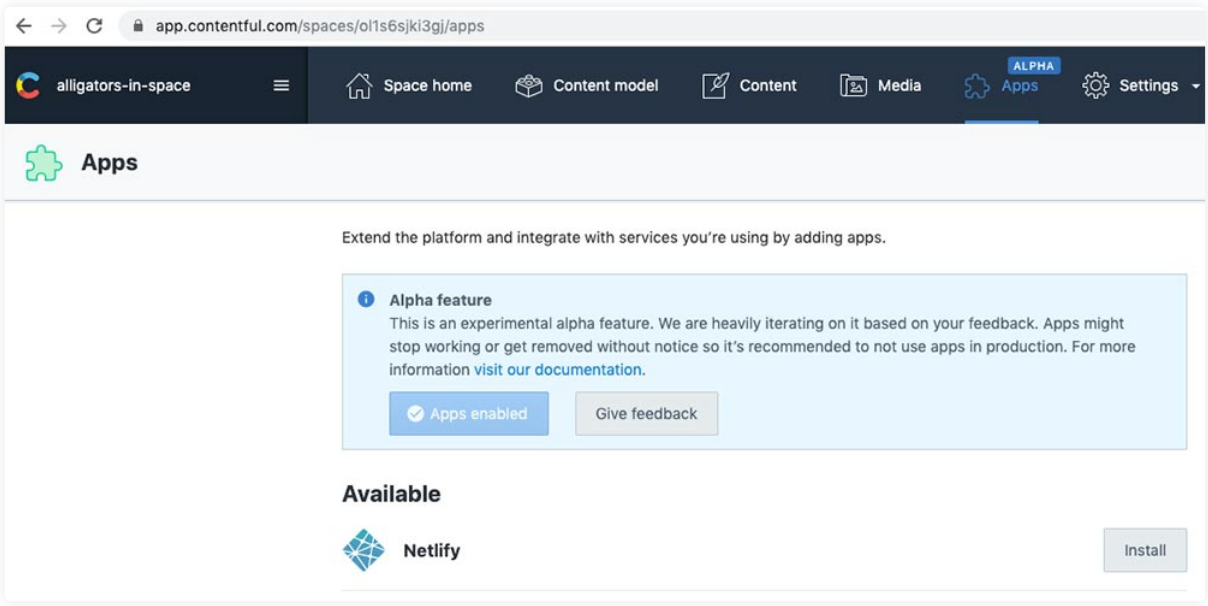
From now on, any time you push code to `master` at your GitHub repo, Netlify will build and deploy your website.

But we're not done yet.

## *Connect Contentful and Netlify*

We can go a step further and cut you, the dev, out of the content team's workflow altogether.

Because Netlify has a hook for Contentful to trigger updates whenever a change is published there.

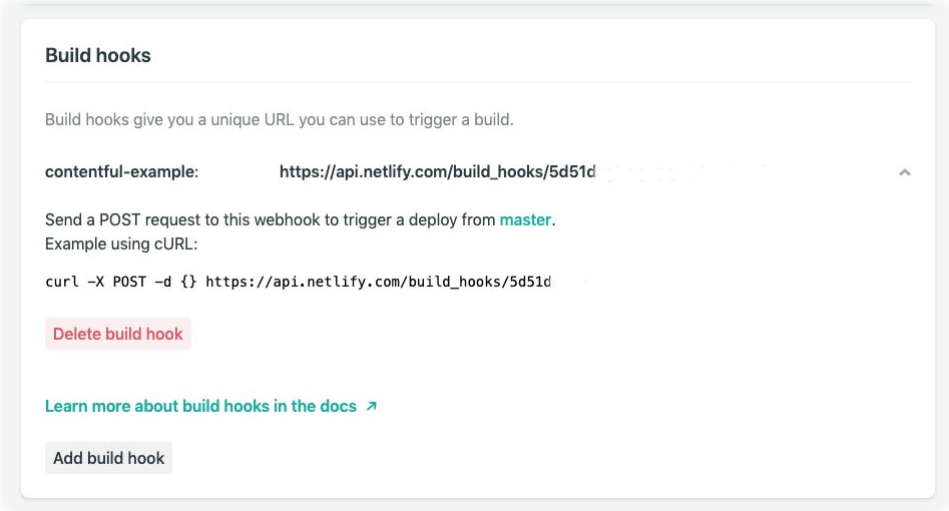



Go to your Contentful account. Click `Apps` from the top nav, `enable apps`, then install the Netlify app and follow the instructions to connect your accounts.

To see how the connection has been made at Netlify, from the Netlify dashboard select `Settings` from the top menu and then `Build & Deploy` from the left nav. Scroll down to `Build hooks` and you'll see the build hook URL. Something like this:

```
https://api.netlify.com/build_hooks/randomabc123
```

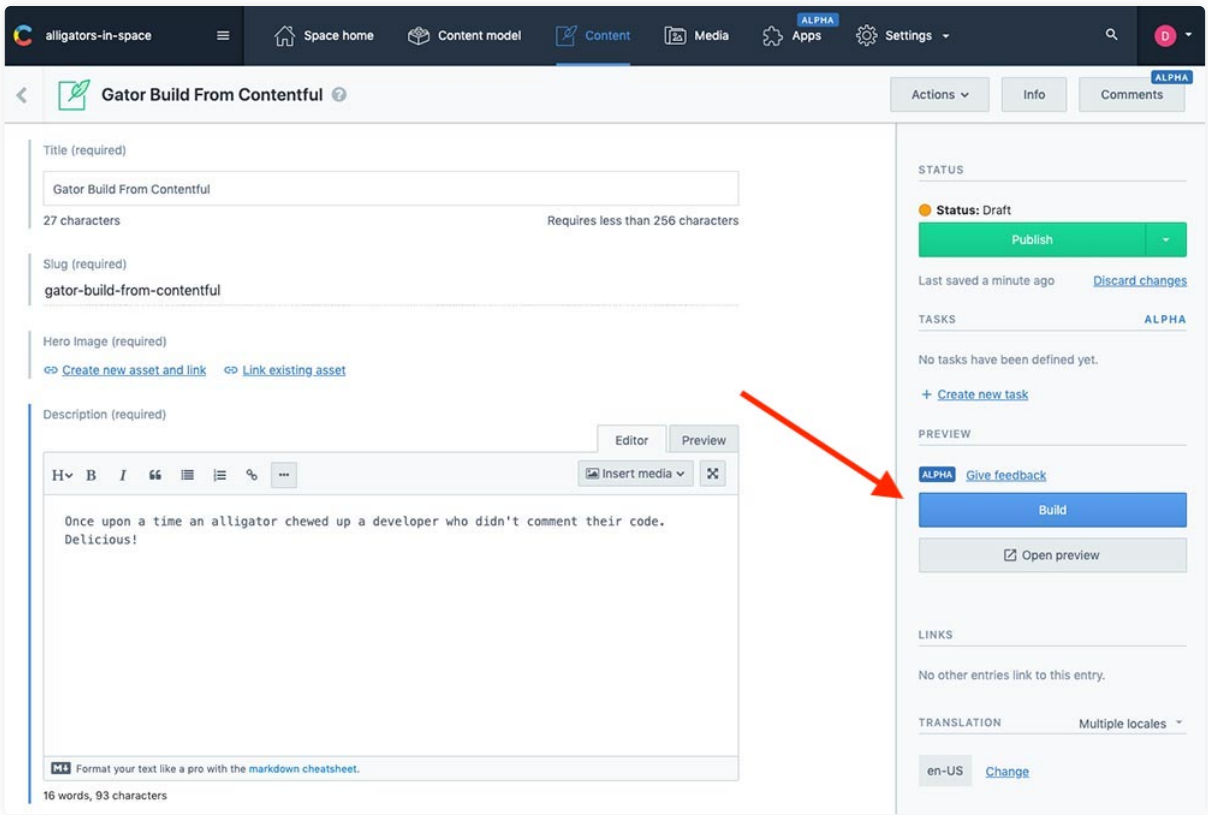
You can delete it at any time if you want to cut the connection.




Now back to Contentful.

Their idea is that you will have two different URLs, your production site and a preview website. But we'll just stick with our single site.

The connection you created is an alpha feature that will add two new buttons next time someone is creating content: Build and Open Preview.



`Open Preview` just opens whatever URL has been set for it, which doesn't do much for us. But the `Build` button will actually **trigger a build and deploy at Netlify**!

That's right! Your content team can create and deploy content from a CMS WYSIWYG system to a static website without having to touch code or enter the terminal!

Note: Anyone who has access to your Contentful space will be able to trigger a build, so be judicious with access.

You now have several ways to deploy your site:

### Contentful

- `Build` button

### Netlify

- GitHub updates to trigger build
- `$ yarn build` and upload the `/public` folder
- Web interface inside your Netlify account
- POST to your build_hook, like

```
$ curl -X POST -d {} https://api.netlify.com/build_hooks/abcdefgh123456789
```

## *Feels Good*

So who's going to party harder? You, freed from constantly having to apply tedious updates, or the content creators who can finally update their site again?

## latest gatsbyjs posts

Using React Hooks in Gatsby

Pagination in Gatsby Using gatsby-awesome-pagination

Exploring TinaCMS with Gatsby

State Management in Gatsby using the wrapRootElement Hook

all gatsbyjs posts

## follow us @alligatorio

Published: August 13, 2019