**How to Deploy a React App to Heroku**

[Joel Olawanle](https://stackabuse.com/author/joel/)

**Introduction**

When a developer creates an application, the next step is to share it with friends or the public so that everyone can access it. That process of transferring code from a development environment to a hosting platform where it is served to end users is called *deployment*.

Hosting used to be pretty inefficient before cloud hosting platforms like Heroku came around. It was mainly done by hosting providers who required uploading all static assets (build files generated by running npm run build) every time we make a change. There was no other way to upload static files other than some sort of FTP interface (either a local one or on the hosting server), which can be pretty stressful and technical.

*In this guide, we'll take a look at how to deploy a React application to Heroku using the CLI (Command Line Interface) via Heroku Git. Also, we will take a look at how to redeploy code when we make some changes to our application.*

**What Is Heroku and Why Use It?**

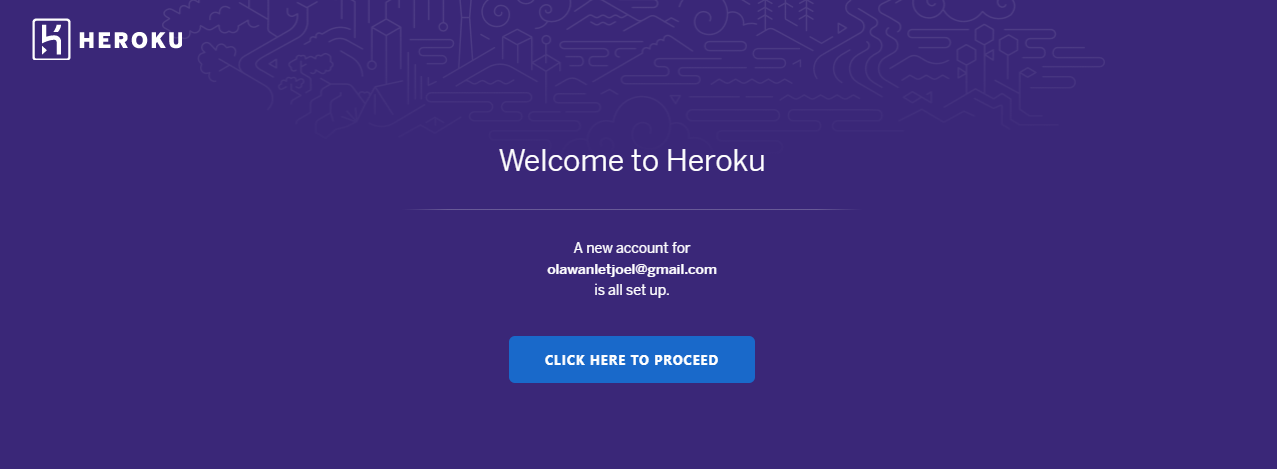
Heroku is a container-based cloud platform that enables developers to easily deploy, manage, and scale modern applications. This allows developers to focus on their core job - creating great apps that delight and engage users. In other words, Heroku increases the developer's productivity by making app deployment, scaling, and management as simple as possible.

There are numerous reasons why we should use Heroku:

* Supports *multiple languages* - from the ground up, the Heroku platform supports more than eight languages, including Node, Java, and Python.
* Supports *several databases* and data stores - Heroku enables developers to select from a variety of databases and data stores based on the specific requirements of individual applications - Postgres SQL, MySQL, MongoDB, and so on.
* *Less expensive* - creating and hosting a static website will save us money in the long run.

**Getting Started**

In this guide, we will deploy a [movies search app](https://github.com/StackAbuse/movies-search-app), which is a simple React app that searches an API for movies. Before we begin, you should sign up for Heroku if you do not already have an account, as this is where we will deploy our React application. We can go to [Heroku.com](https://www.heroku.com/) and sign up by clicking the sign-up button in the upper right corner. The signup pipeline is pretty much the standard one, so you shouldn't have any trouble creating an account on Heroku:



When you've created a Heroku account, we can proceed to the actual deployment of our app.

**Note:** Previously, there was an option to deploy via GitHub Integration, but that feature has been revoked due to a security breach. The best way to deploy to Heroku as of now is via Heroku Git, which happens in our CLI (*Command Line Interface*).

**Deployment With *Heroku Git***

Heroku uses *the Git* version control system to manage app deployments. It is important to note that we do not need to be Git experts to deploy our React application to Heroku, all we need to know are some fundamentals, which will be covered in this guide.

If you're not confident with Git - don't worry. We'll cover everything you need to know. Otherwise, check out our **free course** on Git: [*Git Essentials: Developer's Guide to Git*](https://stackabuse.com/courses/git-essentials-developers-guide-to-git/)

**Before We Start**

As the name *Heroku Git* implies, we will be using Git, which means we need to have Git installed. The same applies to the Heroku CLI. If you don't have those two installed, you can follow the following instructions to guide you through the installation process:

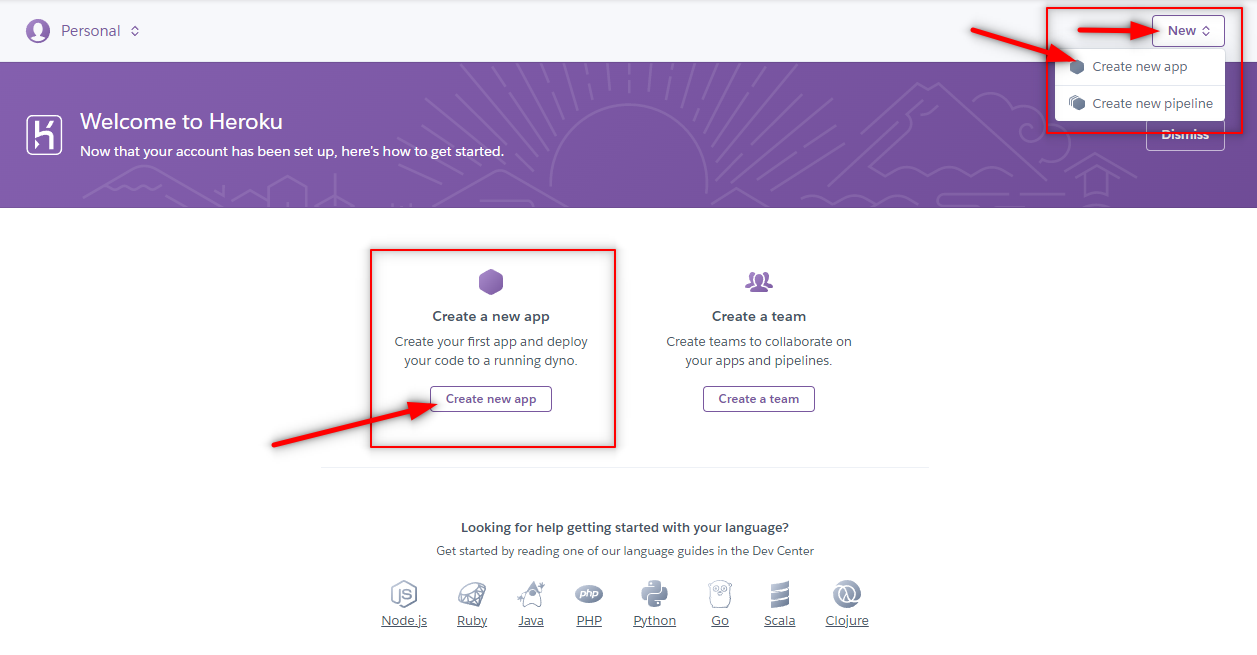
* [Git installation instructions](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git)
* [Heroku CLI installation instructions](https://devcenter.heroku.com/articles/heroku-cli#install-the-heroku-cli)

After successfully installing them, we can proceed to create an app on Heroku, to which our React application will be deployed later. We can create an application on Heroku in two ways - via the *terminal* (CLI command) or manually on our *Heroku dashboard*.

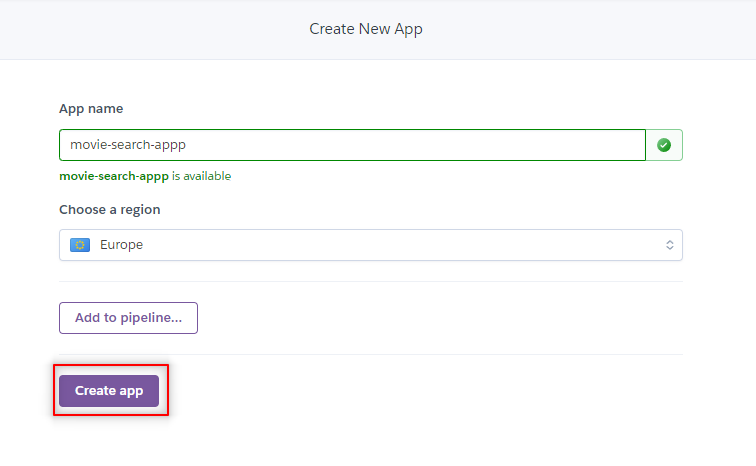
**Note:** A common misconception is that Git and GitHub are the same things, but they are not! Git is a version control system used by many apps and services, including but not limited to GitHub. Therefore you don’t need to push your code to GitHub, nor have a GitHub account to use Heroku.

**How to Create Heroku App Manually**

Let’s first see how we can create an app using the [Heroku dashboard](https://dashboard.heroku.com/). The first step is to click the **create new app** button:



This would redirect us to a page where we need to fill up the information about the app we want to create:



**Note:** Make sure you *remember the name* of the app you created on Heroku because we will be connecting our local repository to this remote repository soon.

Once this process is completed, we can start deploying our app from a local environment to Heroku. But, before we take a look at how to deploy an app, let's consider an alternative approach to creating a Heroku app - using the Heroku CLI.

**How to Create Heroku App via CLI**

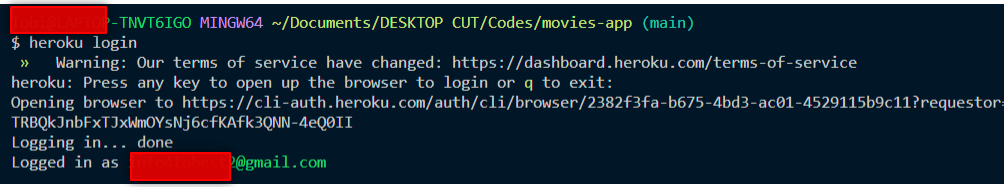
Alternatively, you can create an app on Heroku using the CLI. Heroku made sure this is as straightforward as possible. The only thing you need to do is to run the following command in your terminal of choice (just make sure to replace <app-name> with the actual name of your app):

$ heroku create -a <app-name>

**Note:** If you run this command from the app’s root directory, the empty Heroku Git repository is automatically set as a remote for our local repository.

**How to Push Code to Heroku**

The first step before pushing the code to Heroku will be to position yourself in the root directory of your app (in the terminal). Then use the heroku login command to log into the Heroku dashboard. After that, you need to accept Heroku's terms and conditions and, finally, log in to Heroku using your login credentials:



Text

Description automatically generated

**Free eBook: Git Essentials**

Check out our hands-on, practical guide to learning Git, with best-practices, industry-accepted standards, and included cheat sheet. Stop Googling Git commands and actually *learn* it!

Download the eBook

You will be returned to the terminal afterward, so you can continue the process of deploying to Heroku. Now, you should initialize the repository:

$ git init

And then register the app we created earlier on Heroku as the remote repository for the local one we initialized in the previous step:

$ heroku git:remote -a <app-name>

**Note:** Make sure to replace <app-name> with the name of the app we've created on Heroku earlier (e.g. movies-search-app).

Now we can proceed to deploy our application. But, since we need to deploy a React application, we first need to add the React buildpack:

$ heroku buildpacks:set mars/create-react-app

Once that is completed, the next step is to actually push our code to the remote repository we've created on Heroku. The first step is to stage our files, commit them, and finally push them to the remote repository:

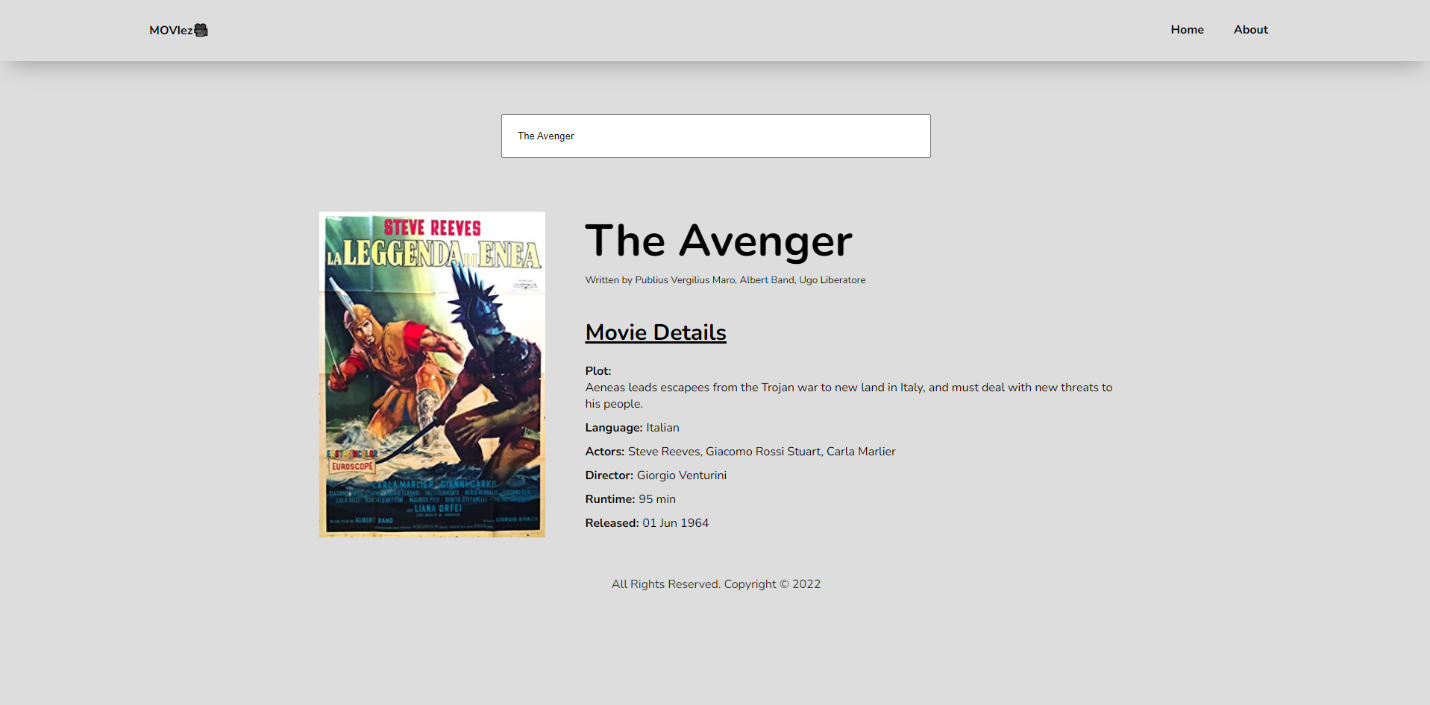
$ git commit -am "my commit"

$ git push heroku main

**Note:** Suppose we want to switch our branch from main to development. We can run the following command: git checkout -b development.

Once we have successfully pushed to Heroku, we can open our newly deployed app in our browser:

$ heroku open



**How to Update Our Deployment**

The next question you'd probably have is how to redeploy the app after we make changes to it. This works similarly to how it does in any Git-based platform - all we have to do is stage the files, commit, and then push the code to Heroku:

$ git commit -am "added changes"

$ git push heroku main

Heroku automatically picks this change up and applies it to the live application.

**Conclusion**

Heroku can be a fairly useful tool for deploying your React app. In this article, we've taken a look at how to deploy a React application to Heroku using Heroku Git. Additionally, we've gone over some basic Git commands