

How to Deploy Angular Application to Heroku



Olutunmbi Banto

Oct 28, 2017 · 5 min read



It has always ‘seemed’ easy until you try it. Having deployed other apps to heroku, I encountered series of challenges deploying Angular 4 recently but I pulled through finally. So am writing to explain in details how I did it.

This article will show you guides on deploying your Angular 2/4 apps easily to Heroku, more importantly, avoiding common pitfalls.

This is not a tutorial to help you learn Angular. It will be assumed you have completed development and ready to deploy. However, we’ll setup basic angular project from start and deploy. This tutorial will cover:

- Creating basic angular project
- Setting automatic deployment from GitHub to Heroku
- Deploying Angular app on Heroku server

Setup Your Angular Application

Making use of the Angular CLI, setup a new project by running:

```
ng new demo-deploy
```

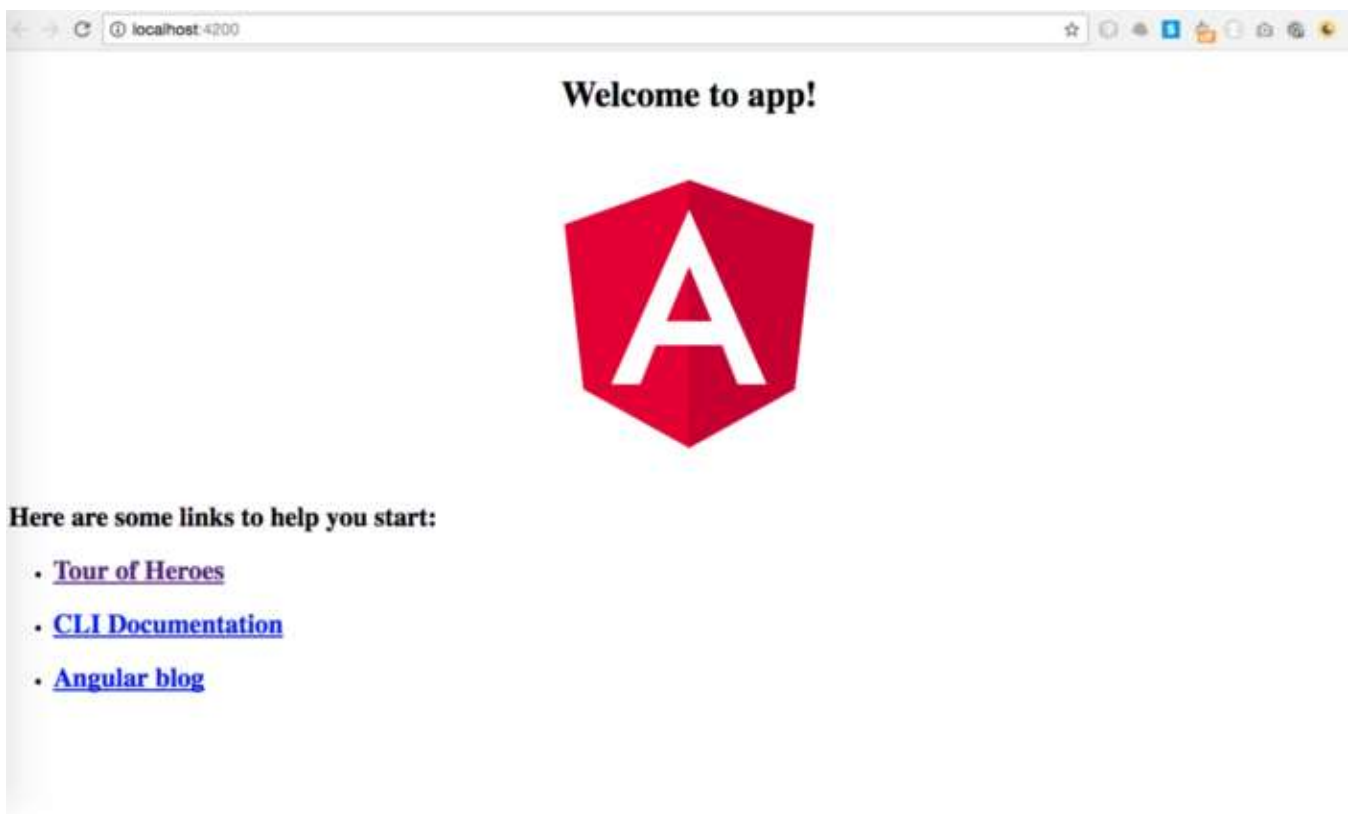
From this, our application will be named `demo-deploy` . Allow for few minutes to setup the new project and install npm packages.

Launch Application

Change directory into new project and launch it using the commands below. This will open in new browser on port 4200 by default. i.e <http://localhost:4200>.

```
cd demo-deploy
```

```
ng serve
```



Angular app greets you welcome

Our basic angular app is ready and running — locally. Lets push to github

Create its GitHub repo and Push

Here, we'll be creating a fresh github repository and pushing our app to it.

- Login to github and create new repository. No need to initialize repository with README
- Open new tab in your terminal/CMD. Or hit Ctrl+C to stop running app. Then run the following commands:

```
git remote add origin <new_github_repository_url>  
git add .  
git commit -m "initial commit"  
git push -u origin master
```

Now our app is on github.

Setup Automatic Deployment from GitHub to Heroku

Advantage of this step is so that, once you push a change to your github repository, it automatically pushes the change to your codebase on heroku, which then takes effect live on the web. This means, you'll only have to push your changes to github and its done.

If you dont have an account yet, create one on [heroku website](https://heroku.com). Its free. Login to your dashboard and create a new app.



create a new app on heroku

Create New App

App name

demo-deploy-app

demo-deploy-app is available

Choose a region

United States

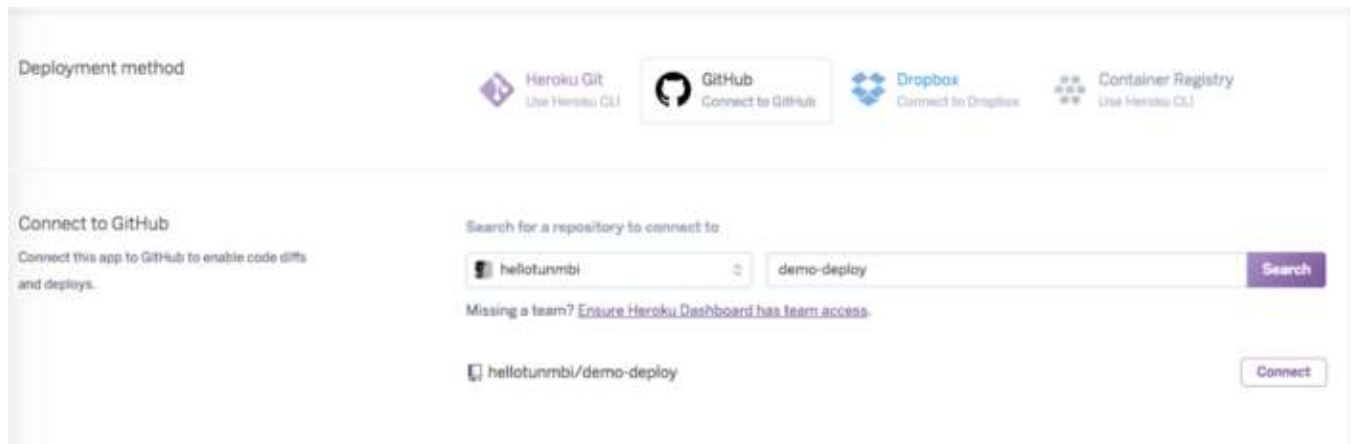
Add to pipeline...

Create app

Click **Create app**

In the **Deploy** menu, under **Deployment method**, select **GitHub**. If you have not done this already, it will ask you to login your github account so it can connect to it.

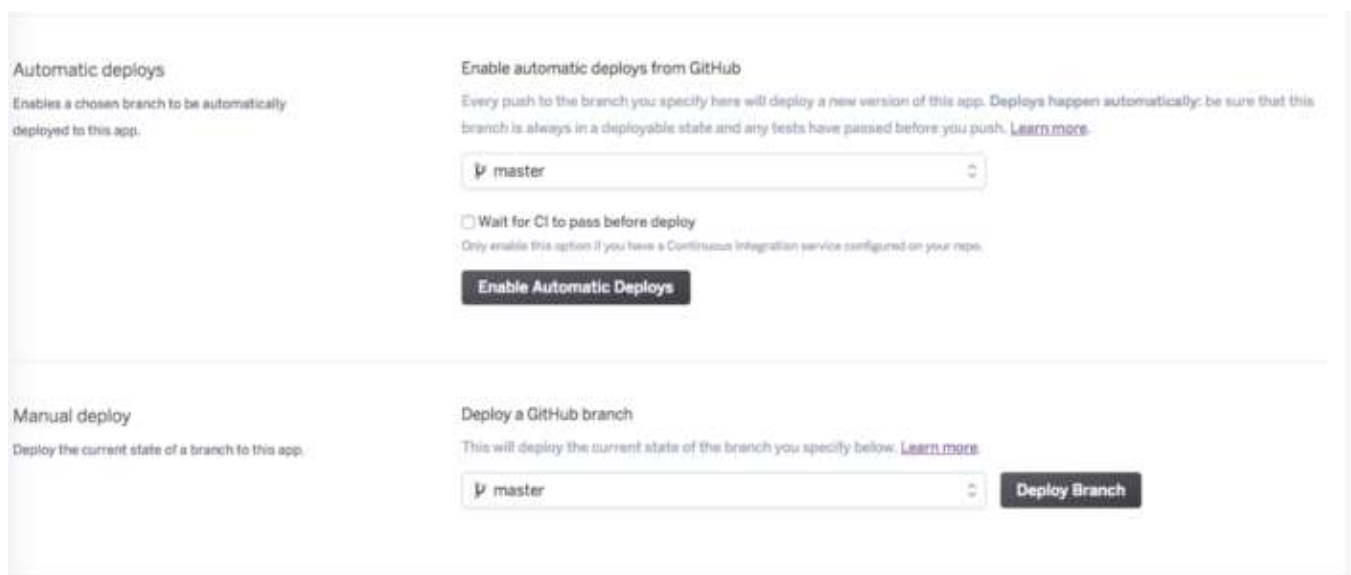
Enter the name of the GitHub repository and click **Search**. Once the repo is shown below, click **Connect**. Viola!



The screenshot shows the Heroku 'Deployment method' selection interface. At the top, there are four options: Heroku Git (Use Heroku CLI), GitHub (Connect to GitHub), Dropbox (Connect to Dropbox), and Container Registry (Use Heroku CLI). The 'GitHub' option is selected. Below this, the 'Connect to GitHub' section is active. It contains a search bar with the text 'Search for a repository to connect to:'. The search results show 'hellotunmbi' as the username and 'demo-deploy' as the repository name. A 'Search' button is next to the repository name. Below the search results, there is a link 'Missing a team? Ensure Heroku Dashboard has team access.' and a 'Connect' button next to the repository name 'hellotunmbi/demo-deploy'.

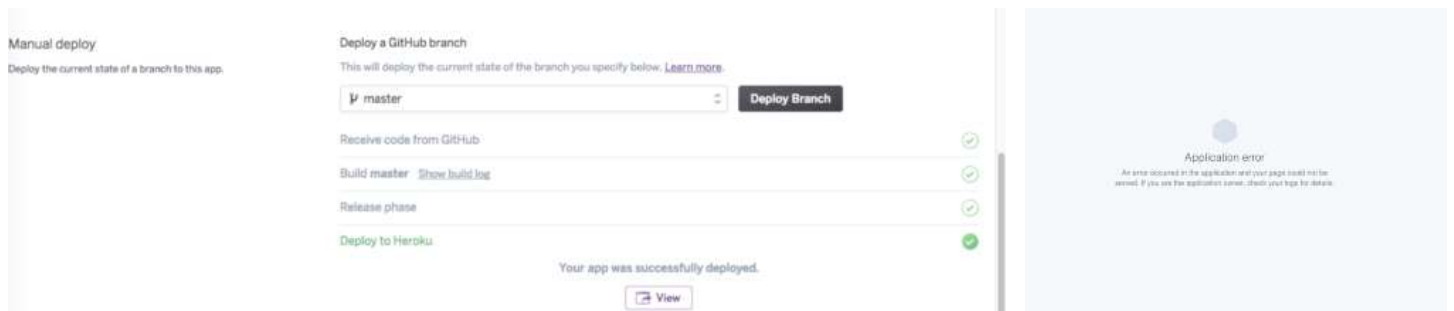
Uh, wait. Two more simple steps.

1. Under Automatic Deploys, select the master branch and click Enable Automatic Deploys.
2. Under Manual Deploys, click Deploy Branch. This is to push our fresh code to heroku.



The screenshot shows two sections of the Heroku deployment configuration interface. The top section is 'Automatic deploys', which includes a description 'Enables a chosen branch to be automatically deployed to this app.' and a sub-section 'Enable automatic deploys from GitHub'. This sub-section has a description 'Every push to the branch you specify here will deploy a new version of this app. Deploys happen automatically: be sure that this branch is always in a deployable state and any tests have passed before you push. [Learn more](#).' Below this is a dropdown menu with 'master' selected. There is also an unchecked checkbox 'Wait for CI to pass before deploy' with a note 'Only enable this option if you have a Continuous Integration service configured on your repo.' and an 'Enable Automatic Deploys' button. The bottom section is 'Manual deploy', which includes a description 'Deploy the current state of a branch to this app.' and a sub-section 'Deploy a GitHub branch'. This sub-section has a description 'This will deploy the current state of the branch you specify below. [Learn more](#).' Below this is a dropdown menu with 'master' selected and a 'Deploy Branch' button.

Okay, we'r done with this stage really. It might take a little while but will show you successfully deployed message once done, like so:



If you click View, a new tab will be opened but your app will not display. Next series of steps will guide you on configuring and spinning up your angular app.

Configure Your Angular App to Deploy Properly on Heroku

The following are production-ready steps to easily and properly deploy your app without hitches.

Ensure you have the latest version of angular cli and angular compiler cli.

Install them into your application by running this commands in your terminal:

```
npm install @angular/cli@latest @angular/compiler-cli --save-dev
```

In your package.json, copy

```
"@angular/cli": "1.4.9",  
"@angular/compiler-cli": "^4.4.6",
```

from devDependencies to dependencies

Create postinstall script in package.json

Under “scripts”, add a “heroku-postinstall” command like so:

```
"heroku-postbuild": "ng build --prod"
```

This tells Heroku to build the application using Ahead Of Time (AOT) compiler and make it production-ready. This will create a `dist` folder where all html and javascript converted version of our app will be launched from.

Add Node and NPM engines

You will need to add the Node and NPM engines that Heroku will use to run your application. Preferably, it should be same version you have on your machine. So, run `node -v` and `npm -v` to get the correct version and include it in your package.json file like so:

```
1  "engines": {  
2    "node": "6.11.0",  
3    "npm": "3.10.10"  
4  }
```

gistfile1.txt hosted with ❤ by GitHub

[view raw](#)

Copy typescript to dependencies.

Copy `"typescript": "~2.3.3"` from devDependencies to dependencies to also inform Heroku what typescript version to use.

Install Enhanced Resolve 3.3.0

Run the command `npm install enhanced-resolve@3.3.0 --save-dev`

Install Server to run your app

Locally we run `ng serve` from terminal to run our app on local browser. But we will need to setup an Express server that will run our production ready app (from dist folder created) only to ensure light-weight and fast loading.

Install Express server by running:

```
npm install express path --save
```

Create a server.js file in the root of the application and paste the following code.

```
1  //Install express server  
2  const express = require('express');  
3  const path = require('path');  
4  
5  const app = express();  
6  
7  // Serve only the static files form the dist directory  
8  app.use(express.static(__dirname + '/dist/<name-of-app>'));  
9  
10 app.get('/*', function(req,res) {  
11
```

```
12 res.sendFile(path.join(__dirname+'/dist/<name-of-app>/index.html'));
13 });
14
15 // Start the app by listening on the default Heroku port
16 app.listen(process.env.PORT || 8080);
```

gistfile1.txt hosted with ❤ by GitHub

[view raw](#)

Change start command

In package.json, change the “start” command to `node server.js` so it becomes:

```
"start": "node server.js"
```

Here’s what the complete package.json looks like. Yours may contain more depending on your application-specific packages.

```
1  {
2    "name": "demo-deploy",
3    "version": "0.0.0",
4    "license": "MIT",
5    "scripts": {
6      "ng": "ng",
7      "start": "node server.js",
8      "build": "ng build",
9      "test": "ng test",
10     "lint": "ng lint",
11     "e2e": "ng e2e",
12     "postinstall": "ng build --aot -prod"
13   },
14   "private": true,
15   "dependencies": {
16     "@angular/animations": "^4.2.4",
17     "@angular/cli": "^1.4.9",
18     "@angular/common": "^4.2.4",
19     "@angular/compiler": "^4.2.4",
20     "@angular/compiler-cli": "^4.4.6",
21     "@angular/core": "^4.2.4",
22     "@angular/forms": "^4.2.4",
23     "@angular/http": "^4.2.4",
24     "@angular/platform-browser": "^4.2.4",
25     "@angular/platform-browser-dynamic": "^4.2.4",
26     "@angular/router": "^4.2.4",
27     "core-js": "^2.4.1",
28     "express": "^4.16.2",
29     "rxjs": "^5.4.2",
30     "typescript": "~2.3.3",
```

```
31     "zone.js": "^0.8.14"
32   },
33   "devDependencies": {
34     "@angular/cli": "^1.4.9",
35     "@angular/compiler-cli": "^4.4.6",
36     "@angular/language-service": "^4.2.4",
37     "@types/jasmine": "~2.5.53",
38     "@types/jasminewd2": "~2.0.2",
39     "@types/node": "~6.0.60",
40     "codelyzer": "~3.2.0",
41     "enhanced-resolve": "^3.3.0",
42     "jasmine-core": "~2.6.2",
43     "jasmine-spec-reporter": "~4.1.0",
44     "karma": "~1.7.0",
45     "karma-chrome-launcher": "~2.1.1",
46     "karma-cli": "~1.0.1",
47     "karma-coverage-istanbul-reporter": "^1.2.1",
48     "karma-jasmine": "~1.1.0",
49     "karma-jasmine-html-reporter": "^0.2.2",
50     "protractor": "~5.1.2",
51     "ts-node": "~3.2.0",
52     "tslint": "~5.7.0",
53     "typescript": "~2.3.3"
54   },
55   "engines": {
56     "node": "6.11.0",
57     "npm": "3.10.10"
58   }
59 }
```

Push changes to GitHub:

```
git add .
git commit -m "updates to deploy to heroku"
git push
```

At this point, your application on Heroku will automatically take the changes from GitHub and update itself.

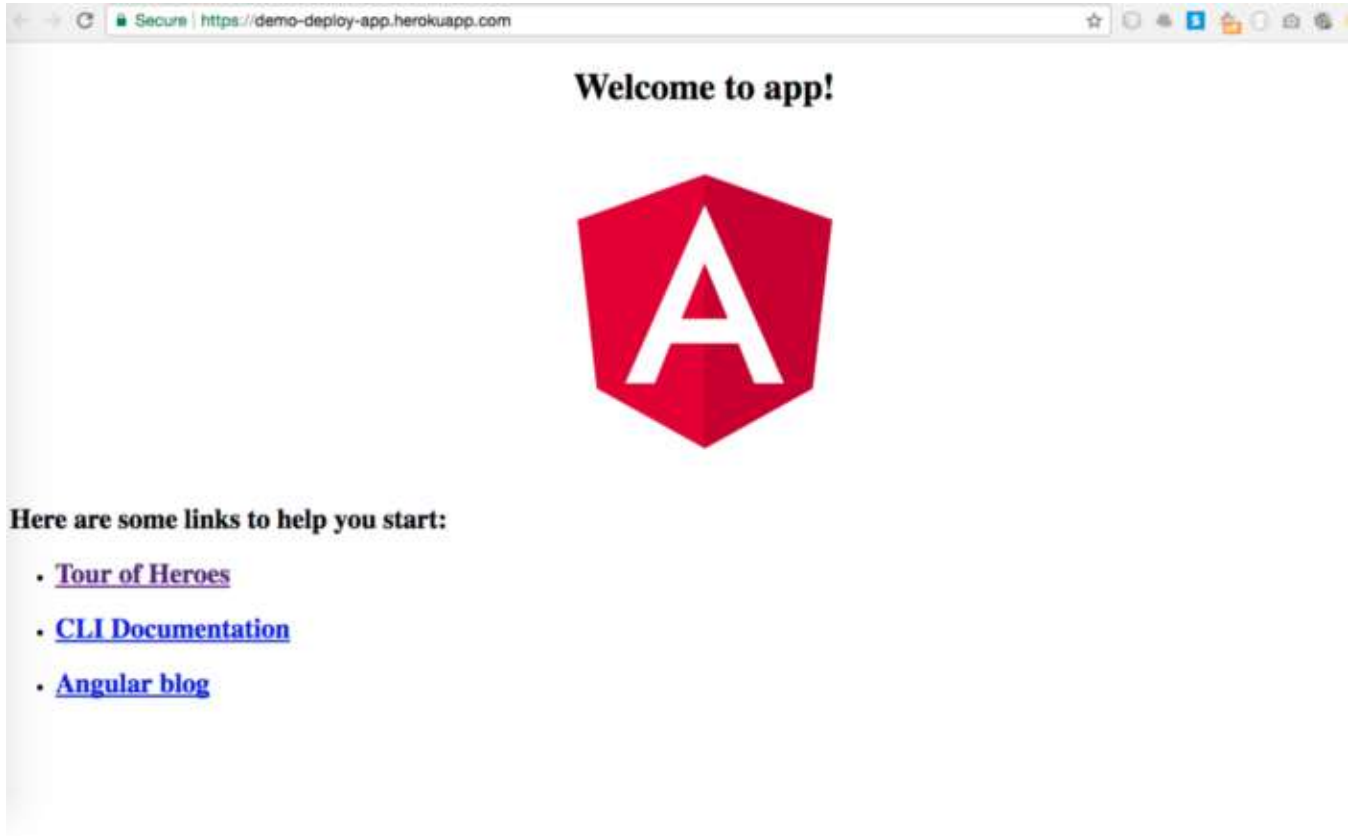
Also, it'll look into your package.json and install packages.

It will run the postinstall and then, `node server.js` to spin up your application.

You can check Activity tab and open Build log to see how it actually runs.

You should not run into any issue. I followed through while writing this post also and.

Viola!! Our Angular app is Ready and LIVE!



For following through till this stage, Thank you.

You can follow me on [Twitter](#) or here on Medium.

Say cheese! ... comment below, also if you encountered any issue or want to suggest better ways.

If you found this article helpful and would like to read more and know when I publish related articles, do [subscribe to my newsletter](#). I promise not to spam you.

Get the Medium app

