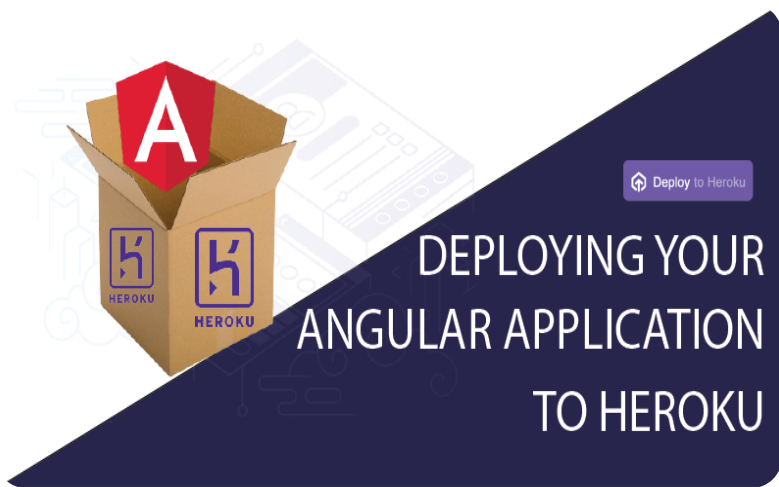


How to Deploy Angular Application to Heroku

December 20, 2020 Sunday (6 minute read) by Girish Godage Deployment



Introduction

It has always 'seemed' easy until you try it. Having deployed other apps to heroku, I encountered series of challenges deploying Angular 11 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 8/9/10/11 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

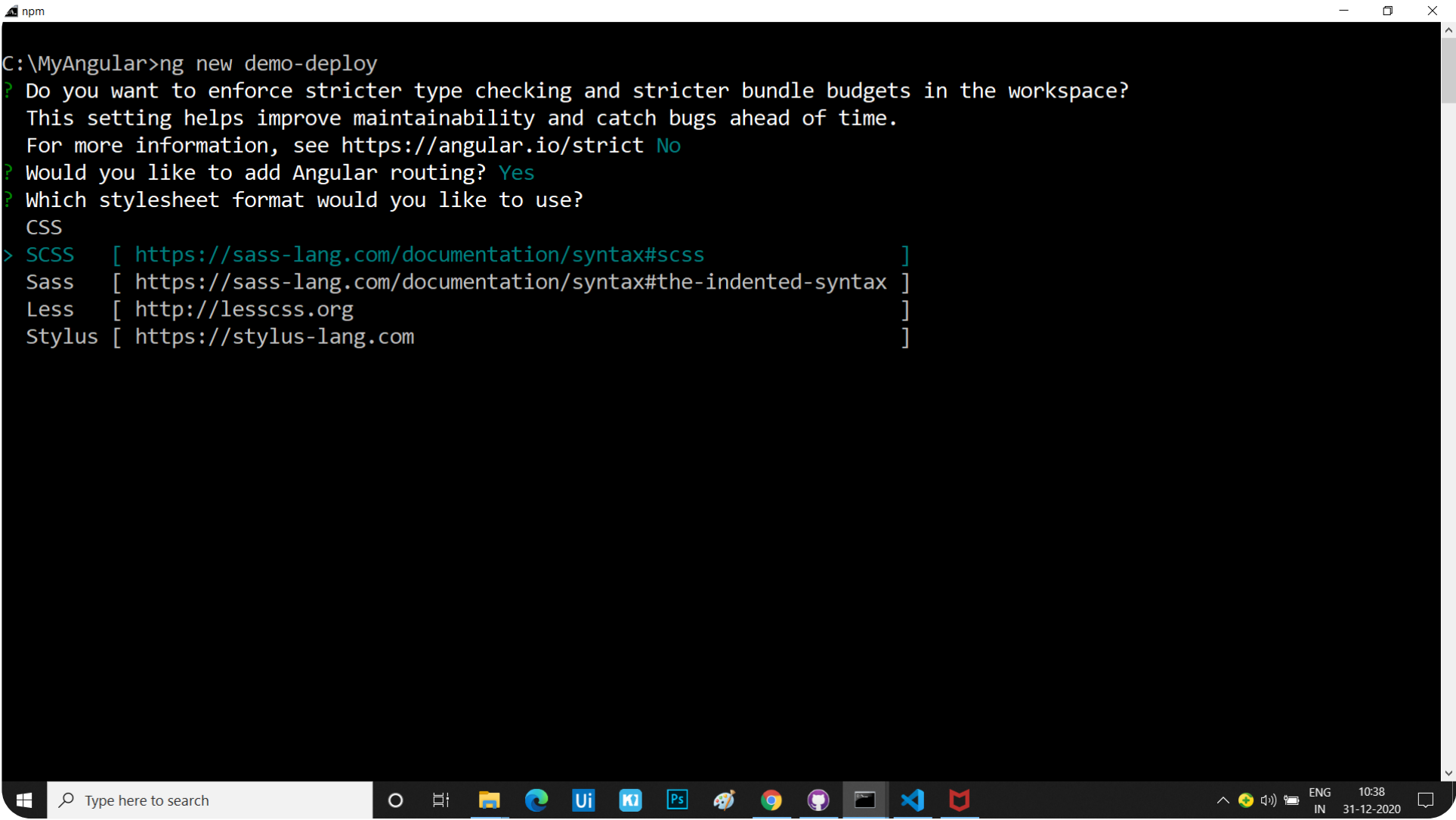
Setup Your Angular Application

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

```
ng new demo-deploy
```

COPY





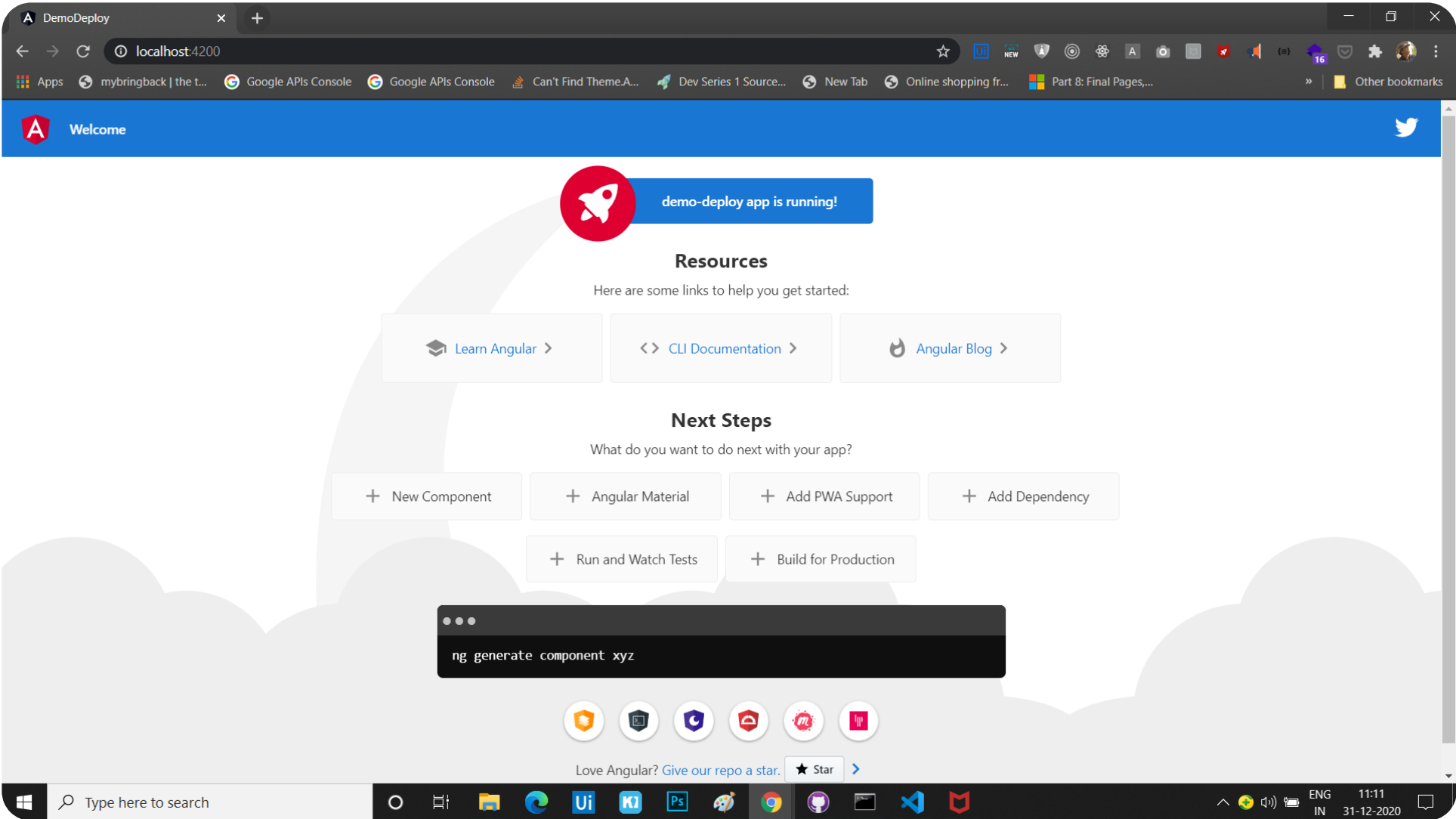
From this, our application will be named demo-deploy . Allow for few minutes to setup the new project and install npm packages.

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

COPY



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 --all
git commit -m "initial commit"
git push -u origin master
```

COPY

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. Its free. Login to your dashboard and create a new app.

Create New App

App name



demo-deploy-app is available

Choose a region



 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!

Deployment method

 Heroku Git
Use Heroku CLI

 GitHub
Connect to GitHub


 Dropbox
Connect to Dropbox

 Container Registry
Use Heroku CLI

Connect to GitHub


Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to



Search

Missing a team? [Ensure Heroku Dashboard has team access.](#)




Connect

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.



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:



Application error

An error occurred in the application and your page could not be served. If you are the application owner, check your logs for details.

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
```

COPY

In your `package.json`, copy



```
"@angular/cli": "~11.0.5",  
"@angular/compiler-cli": "~11.0.5",
```

COPY

from devDependencies to dependencies

Create postinstall script in package.json

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

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

COPY

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:

```
"engines": {  
  "node": "15.2.0",  
  "npm": "7.0.12"  
}
```

COPY

Copy typescript to dependencies.

Copy “typescript”: “~4.0.2” 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
```

COPY

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



```
//Install express server
const express = require('express');
const path = require('path');

const app = express();

// Serve only the static files form the dist directory
app.use(express.static(__dirname + '/dist/<name-of-app>'));

app.get('/*', function(req,res) {

  res.sendFile(path.join(__dirname+'/dist/<name-of-app>/index.html'));
});

// Start the app by listening on the default Heroku port
app.listen(process.env.PORT || 8080);
```

COPY

Change start command

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

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

COPY

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



COPY

```
{
  "name": "demo-deploy",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "node server.js",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e",
    "heroku-postbuild": "ng build --aot --prod "
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "~11.0.5",
    "@angular/cdk": "^11.0.3",
    "@angular/cli": "~11.0.5",
    "@angular/common": "~11.0.5",
    "@angular/compiler": "~11.0.5",
    "@angular/compiler-cli": "~11.0.5",
    "@angular/core": "~11.0.5",
    "@angular/forms": "~11.0.5",
    "@angular/platform-browser": "~11.0.5",
    "@angular/platform-browser-dynamic": "~11.0.5",
    "@angular/router": "~11.0.5",
    "express": "^4.17.1",
    "path": "^0.12.7",
    "rxjs": "~6.6.0",
    "tslib": "^2.0.0",
    "typescript": "~4.0.2",
    "zone.js": "~0.10.2"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.1100.5",
    "@angular/cli": "~11.0.5",
    "@angular/compiler-cli": "~11.0.5",
    "@types/googlemaps": "3.39.14",
    "@types/jasmine": "~3.6.0",
    "@types/node": "^12.11.1",
    "codelyzer": "^6.0.0",
    "enhanced-resolve": "^3.3.0",
    "jasmine-core": "~3.6.0",
    "jasmine-spec-reporter": "~5.0.0",
    "karma": "~5.1.0",
    "karma-chrome-launcher": "~3.1.0",
    "karma-coverage": "~2.0.3",
    "karma-jasmine": "~4.0.0",
    "karma-jasmine-html-reporter": "^1.5.0",
    "protractor": "~7.0.0",
    "ts-node": "~8.3.0",
    "tslint": "~6.1.0",
    "typescript": "~4.0.2"
  },
  "engines": {
    "node": "15.2.0",
    "npm": "7.0.12"
  }
}
```

Push changes to GitHub:

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

COPY

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

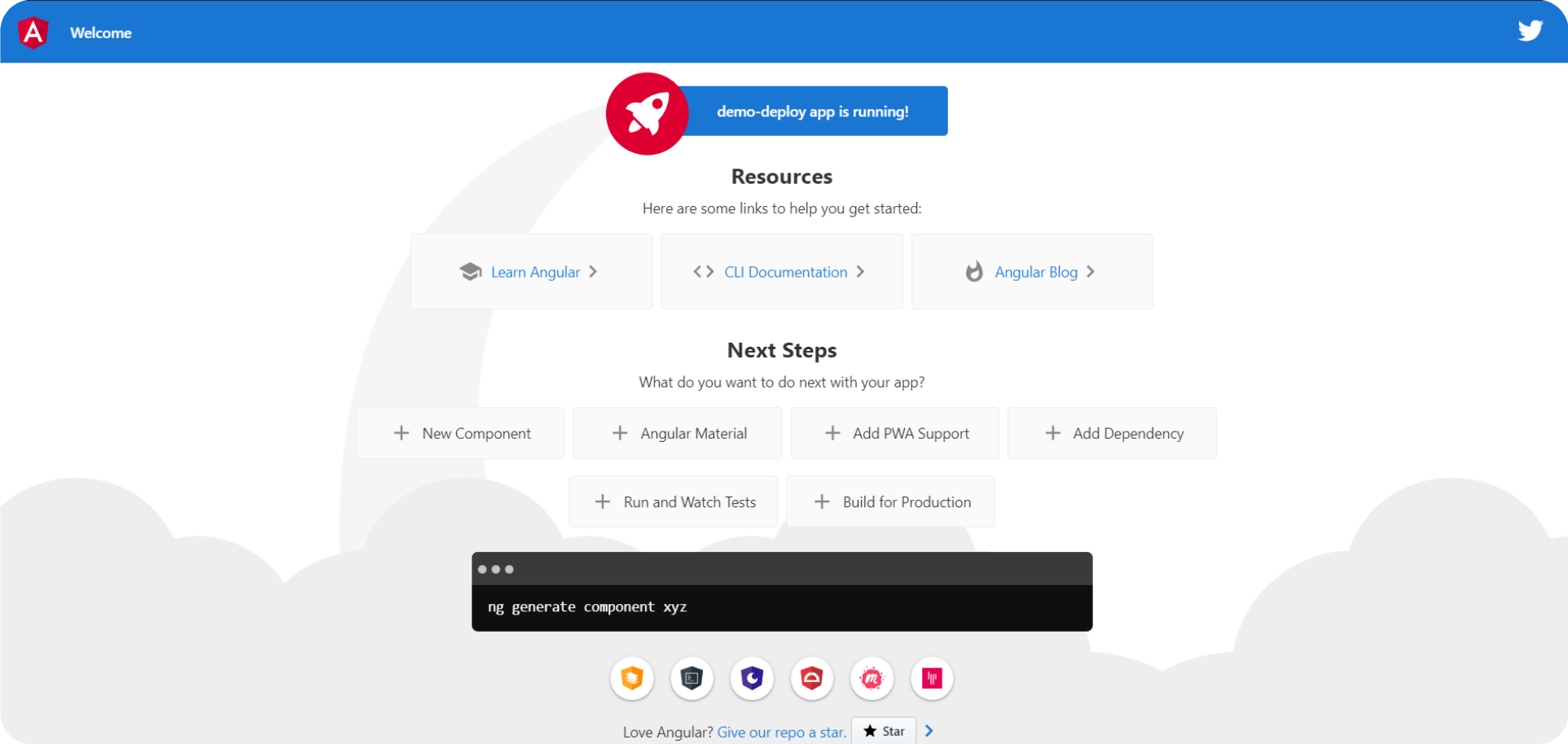
Also, it'll look into your package.json and install packages.

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.

Provide your valuable comment below, also if you encountered any issue or want to suggest better ways.

What do you think?

0 Responses

👍 Upvote

😂 Funny

❤️ Love

😮 Surprised

😡 Angry

😞 Sad

0 Comments

girishgodage.github.io

🔒 Disqus' Privacy Policy

1 Login

❤️ Recommend

🐦 Tweet

📱 Share

Sort by Best

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Be the first to comment.

📧 Subscribe

🔗 Add Disqus to your siteAdd DisqusAdd

⚠️ Do Not Sell My Data

- Sitemap

