

How to Deploy Angular 8 app on Google Cloud Platform (GCP) ?



Karthik Sagar [Follow](#)

Nov 13, 2019 · 3 min read

Deploying your production version of app feels overwhelming when we do not have right resource. The process is quite simple. Once you get hang of it, it feels quite simple. If you are juggling on how to deploy Angular App on GCP, you have come to right place.

There are two ways —

1. **Google Compute Engine:** GCE is Infrastructure-as-a-service i.e. IaaS (We would get a box)
2. **Google App Engine:** GAE is Platform-as-a-service i.e. PaaS. (We need not worry about scaling and use the service)

We would be using Google App Engine to deploy our angular app in 5 steps.

Step 1 : Build production version of your Angular App

> ng build — prod

```
Generating ES5 bundles for differential loading...
ES5 bundle generation complete.

chunk {2} polyfills-es2015.2987770fde9daa1d8a2e.js (polyfills) 36.4 kB [initial] [rendered]
chunk {3} polyfills-es5.ef4b1e1fc703b3ff76e3.js (polyfills-es5) 122 kB [initial] [rendered]
chunk {0} runtime-es2015.edb2fcf2778e7bf1d426.js (runtime) 1.45 kB [entry] [rendered]
chunk {0} runtime-es5.edb2fcf2778e7bf1d426.js (runtime) 1.45 kB [entry] [rendered]
chunk {1} main-es2015.c9495f53970ad187fe64.js (main) 1.56 MB [initial] [rendered]
chunk {1} main-es5.c9495f53970ad187fe64.js (main) 1.63 MB [initial] [rendered]
chunk {4} styles.e8b2ef8f1d891acbc5a8.css (styles) 61 kB [initial] [rendered]
chunk {scripts} scripts.a2656b813c386aba9b52.js (scripts) 168 kB [entry] [rendered]
Date: 2019-11-13T03:09:38.106Z - Hash: 1bf1060ff188a36a85c5 - Time: 73105ms
```

At the end of this step, new folder **dist** will be created containing these minified CSS and JS files. (It is quite interesting as to how our entire code is compressed and optimized to these few files)

Step 2 : Create new configuration file required for hosting

We need to create a new file **app.yaml** to specify the few configurations like that guides GCP to find starting directory, static files and others as you see below.

Copy the below contents to newly created **app.yaml** file

```
runtime: python27

api_version: 1

threadsafe: true

handlers:

- url: /

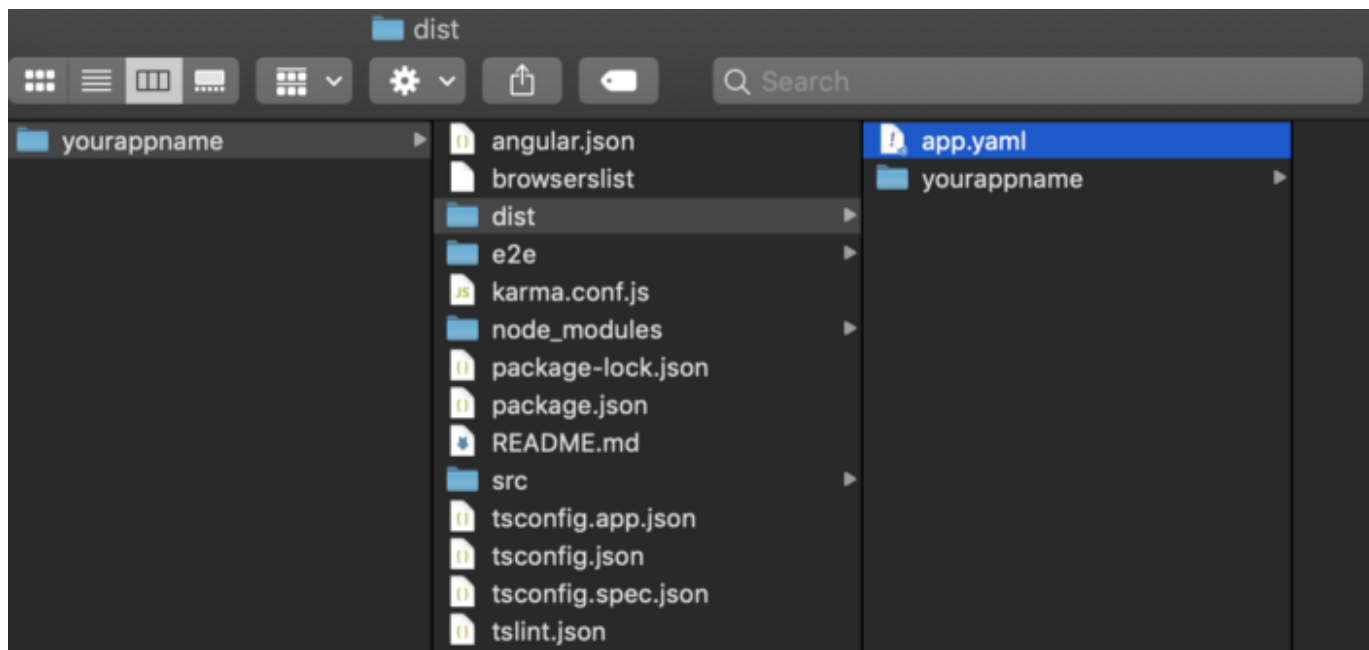
  static_files: <YOUR_APP_NAME>/index.html

  upload: <YOUR_APP_NAME>/index.html

- url: /

  static_dir: <YOUR_APP_NAME>
```

Important — Place this file inside dist folder. Please see the directory structure in the image below.




Step 3 : Configuration in GCP UI

a) Login to Google console to create new project (e.g. angular-project)

Google Cloud Platform

New Project



You have 13 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *


YOURPROJECTNAME

?

Project ID: yourprojectname-258903. It cannot be changed later. [EDIT](#)

Any charges for this project will be billed to the account you select here.

Location *

 No organization

[BROWSE](#)

Parent organization or folder

CREATE

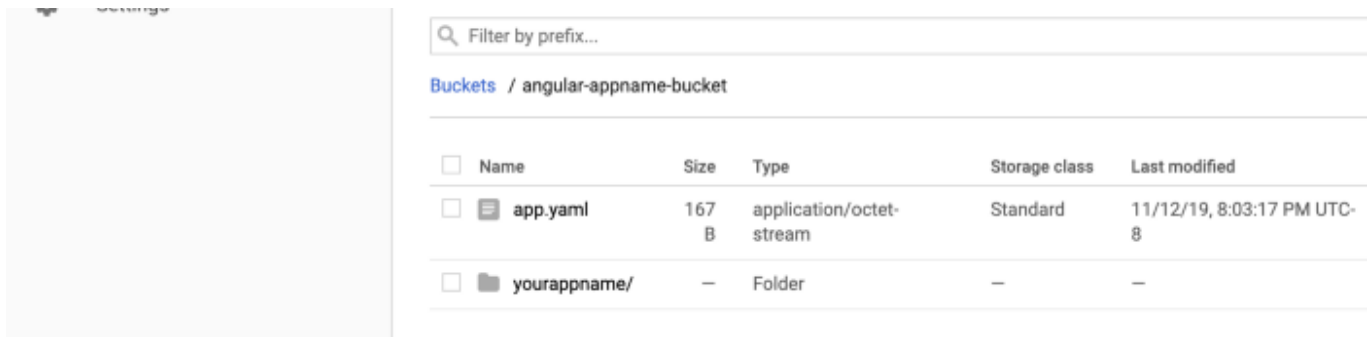
CANCEL

b) Create new Google Storage Bucket in the same project. (Just name your bucket and keep rest of details default)

The screenshot shows the 'Create a bucket' wizard in the Google Cloud Platform console. The left sidebar has a 'Storage' section with options: Browser, Transfer, Transfer Appliance, and Settings. The main area is titled 'Create a bucket' and contains a list of steps: 'Name your bucket' (current step), 'Choose where to store your data', 'Choose a default storage class for your data', 'Choose how to control access to objects', and 'Advanced settings (optional)'. In the 'Name your bucket' step, a text input field contains 'angular-appname-bucket' and a tip says 'Don't include any sensitive information'. A 'CONTINUE' button is below the input. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

c) Upload your **project folder** and **app.yaml** file to the newly created bucket.

The screenshot shows the 'Bucket details' page for the 'angular-appname-bucket'. The left sidebar is the same as in the previous screenshot. The main area has a title 'Bucket details' with 'EDIT BUCKET' and 'REFRESH BUCKET' links. Below the title is the bucket name 'angular-appname-bucket' and tabs for 'Objects', 'Overview', 'Permissions', and 'Bucket Lock'. The 'Objects' tab is active. At the bottom, there are buttons for 'Upload files', 'Upload folder', 'Create folder', 'Manage holds', and 'Delete'.



Step 4 : Sync the data in your bucket to your project

a) Login to Google cloud console by clicking console icon from top right of the console.



b) create a new directory

```
> mkdir angular-gcp-app
```

c) Sync data from bucket to this directory

```
> gsutil rsync -r gs://angular-appname-bucket ./angular-gcp-app
```

d) Go inside the directory and deploy the gcp

```
> cd angular-gcp-app
> gcloud app deploy
```

Woohoo Done !

Deployment Url will be displayed in the console. E.g. [https://angular-gcp-app.appspot.com]

[Google Cloud Platform](#) [Angular](#) [Cloud](#)

[About](#) [Help](#) [Legal](#)

Get the Medium app

