Cheat Sheet – Deployment

Deployment Preparations

With the new project structure (Angular CLI as well as custom project structure), deployment is made very easy. In both cases, you have a dist/ folder which will hold all the files you need to deploy.
Therefore, you simply need to deploy this single folder.

Depending on your setup, you should of course make sure that your deployment files are optimized.
Using the Angular CLI, that means that you should run ng build –prod to build your files production-ready (i.e. minified, bundled).
In a custom project you also should have scripts to bundle and minify your files. See the section on creating a custom project structure/ workflow to learn more about that.

Also make sure to execute enableProdMode() in the bootstrap file (don't forget to add the import from @angular/core).

Deployment in General

As explained, no matter to which service/ site you are deploying to, it's the dist/ folder you need to get on your server (as a root folder from which the server should serve).

Example: Deploying to Github Pages

Of course you need a Github account for all that.

Angular CLI (Built-in Way)
Using the Angular CLI, this is very simply, since it has a built-in command for that:

```
ng github-pages:deploy
```

This command will create a new Github repository (if required) and a gh-pages branch in that repository. You'll need to create a token during the process.
Once it is done, you may visit your page at the URL printed by the CLI.
General:
Put your project under version control (git) and stage the dist/ folder (remove it from gitignore + run `git add .`). Then, commit your changes (e.g. `git commit –m 'ready for deployment'`).
Next, push your dist/ folder to Github using this command:

```
git subtree push --prefix dist origin gh-pages
```

You should then be able to view your running app at <your-github-name>.github.io/<project-name>.

Example: Deploying to AWS S3 (+ CloudFront)

Deploying to AWS is pretty easy, too. I'm using S3 since all we have is a static app (no server-side language included).
Of course you should have an AWS account. the first 12 months are free for many AWS services as long as you stay below certain usage amounts. Find out more about the free tier here: https://aws.amazon.com/free/

A very good guide on hosting your static website (optionally also using cloudfront for performance optimizations) can be found here: http://docs.aws.amazon.com/gettingstarted/latest/swh/website-hosting-intro.html

I'm not going to rephrase the whole article here, because it fits perfectly. There are no adjustments required. Just follow this article, upload your dist/ folder contents to your bucket and you should be good to go! ☺