

# Run an Angular app locally with Docker and Nginx



**NGINX**®  
Part of F5

Drew Shirts  
@drewshirts



Drew Shirts

@drewshirts

Software Engineer

Church of Jesus Christ of Latter-Day Saints

# Tools

- Node.js
- NPM
- Visual Studio Code (Or your preferred editor)
- Docker Desktop
- Angular CLI



<https://nodejs.org/en/download/>



# Visual Studio Code

<https://code.visualstudio.com/download>

# Docker Desktop

<https://www.docker.com/products/docker-desktop>

---

Validate install:



```
$ docker --version
```

# Angular CLI



```
$ npm install -g @angular/cli
```

# Angular App



```
$ ng new angular-nginx-docker --defaults
```



```
drewshirts ng-conf-2021 %ng new angular-nginx-docker --defaults
CREATE angular-nginx-docker/README.md (1027 bytes)
CREATE angular-nginx-docker/.editorconfig (274 bytes)
CREATE angular-nginx-docker/.gitignore (631 bytes)
CREATE angular-nginx-docker/angular.json (3647 bytes)
CREATE angular-nginx-docker/package.json (1210 bytes)
CREATE angular-nginx-docker/tsconfig.json (538 bytes)
CREATE angular-nginx-docker/tslint.json (3185 bytes)
CREATE angular-nginx-docker/.browserslistrc (703 bytes)
CREATE angular-nginx-docker/karma.conf.js (1437 bytes)
CREATE angular-nginx-docker/tsconfig.app.json (287 bytes)
CREATE angular-nginx-docker/tsconfig.spec.json (333 bytes)
CREATE angular-nginx-docker/src/favicon.ico (948 bytes)
CREATE angular-nginx-docker/src/index.html (304 bytes)
CREATE angular-nginx-docker/src/main.ts (372 bytes)
CREATE angular-nginx-docker/src/polyfills.ts (2830 bytes)
CREATE angular-nginx-docker/src/styles.css (80 bytes)
CREATE angular-nginx-docker/src/test.ts (753 bytes)
CREATE angular-nginx-docker/src/assets/.gitkeep (0 bytes)
CREATE angular-nginx-docker/src/environments/environment.prod.ts (51 bytes)
CREATE angular-nginx-docker/src/environments/environment.ts (662 bytes)
CREATE angular-nginx-docker/src/app/app.module.ts (314 bytes)
CREATE angular-nginx-docker/src/app/app.component.css (0 bytes)
CREATE angular-nginx-docker/src/app/app.component.html (25725 bytes)
CREATE angular-nginx-docker/src/app/app.component.spec.ts (982 bytes)
CREATE angular-nginx-docker/src/app/app.component.ts (224 bytes)
CREATE angular-nginx-docker/e2e/protractor.conf.js (904 bytes)
CREATE angular-nginx-docker/e2e/tsconfig.json (274 bytes)
CREATE angular-nginx-docker/e2e/src/app.e2e-spec.ts (671 bytes)
CREATE angular-nginx-docker/e2e/src/app.po.ts (274 bytes)
✓ Packages installed successfully.
  Successfully initialized git.
drewshirts ng-conf-2021 %
```

# Angular App Validation



```
$ cd angular-nginx-docker
```



```
$ ng serve
```



drewshirts ng-conf-2021 %cd angular-nginx-docker

drewshirts angular-nginx-docker %ng serve

✓ Browser application bundle generation complete.

Initial Chunk Files	Names	Size
vendor.js	vendor	2.43 MB
polyfills.js	polyfills	128.80 kB
main.js	main	56.03 kB
runtime.js	runtime	6.15 kB
styles.css	styles	119 bytes
Initial Total		2.62 MB

Build at: 2021-04-22T06:07:32.210Z - Hash: b217119e71be0af757fa - Time: 8060ms

\*\* Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ \*\*

✓ Compiled successfully.

# Docker setup: Dockerfile



```
$ touch Dockerfile
```

# Docker setup: .dockerignore



```
$ echo "node_modules" > .dockerignore
```

# Docker setup: Dockerfile Node Stage



Dockerfile > ...

```
1  # Create the node stage with the name "builder"
2  FROM node:latest as builder
3  # Set the working directory
4  WORKDIR /app
5  # Copy files from current directory to working directory
6  COPY . .
7  # Run npm install and build all assets
8  RUN npm i && npm run ng build
9
```



# Docker setup: Dockerfile Nginx Stage

 Dockerfile > ...

```
10  # Create the nginx stage for serving content
11  FROM nginx:alpine
12  # Set the working directory to nginx asset directory
13  WORKDIR /usr/share/nginx/html
14  # Remove default nginx static assets
15  RUN rm -rf ./*
16  # Copy static assets from builder stage
17  COPY --from=builder /app/dist/angular-nginx-docker .
18  # Containers run nginx with global directives and daemon off
19  ENTRYPOINT ["nginx", "-g", "daemon off;"]
```

# Build Docker Image



```
$ docker build -t angular-nginx-docker .
```



```
drewshirts angular-nginx-docker %docker build -t angular-nginx-docker .
```

```
[+] Building 1.5s (14/14) FINISHED
```

```
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 37B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 34B 0.0s
=> [internal] load metadata for docker.io/library/node:latest 1.4s
=> [internal] load metadata for docker.io/library/nginx:alpine 0.0s
=> [builder 1/4] FROM docker.io/library/node:latest@sha256:6cbc150709d59d2667f5d34cbf03fb4594dc 0.0s
=> [stage-1 1/4] FROM docker.io/library/nginx:alpine 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 6.61kB 0.0s
=> CACHED [stage-1 2/4] WORKDIR /usr/share/nginx/html 0.0s
=> CACHED [stage-1 3/4] RUN rm -rf ./ * 0.0s
=> CACHED [builder 2/4] WORKDIR /app 0.0s
=> CACHED [builder 3/4] COPY . . 0.0s
=> CACHED [builder 4/4] RUN npm i && npm run ng build 0.0s
=> CACHED [stage-1 4/4] COPY --from=builder /app/dist/angular-nginx-docker . 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:af7561fc75d33832d291e07f8e1023614d1fbf895b05c984c63512b22270cb3f 0.0s
=> => naming to docker.io/library/angular-nginx-docker 0.0s
```

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

```
drewshirts angular-nginx-docker %
```

# Docker Validation



```
$ docker image ls
```

# Docker Startup



```
$ docker run --rm -it -p 8080:80 angular-nginx
```

# Docker Startup Validation



```
$ docker container ls
```

# Docker Management



```
$ docker container stop af756  
$ docker container rm af756
```





# Thank You!!

Drew Shirts

@drewshirts

Software Engineer

Church of Jesus Christ of Latter-Day Saints