Docker Static Site Tutorial (Explained)

Goal

Serve your own `index.html` using NGINX inside Docker, with:
- a custom folder
- isolation
- no overwriting defaults
- no touching global crap

STEP 1: Make Your Project Folder

Create a new folder to hold everything for this one project -- Dockerfile, HTML, maybe config later.

Why:
It keeps this self-contained. You can zip it, nuke it, move it -- nothing leaks out.

STEP 2: Add Your HTML

Inside your folder, make another folder called `html/`, and put your file in there:
html/index.html

Why:
You want to serve your content. This is your site's root directory. NGINX serves files from here.

STEP 3: Write the Dockerfile

Create a Dockerfile at the root of your project folder:

```
FROM nginx:alpine
COPY html /usr/share/nginx/html
```

Explanation:
- FROM nginx:alpine: pulls the lightweight NGINX image.
- COPY html /usr/share/nginx/html: puts your html/ folder in the container's web root.

STEP 4: Build Your Docker Image

Run:
```
docker build -t my-static-site .
```

Explanation:
- -t my-static-site: tags the image with a name.
- . : look in this folder for the Dockerfile and context.

Docker builds the image using NGINX and adds your HTML.

STEP 5: Run the Container

Run:
```
docker run -d -p 8080:80 my-static-site
```

Explanation:
- -d: detached mode
- -p 8080:80: maps your computer's port 8080 to container's port 80
- my-static-site: the image name

Go to http://localhost:8080 to see your site.

STEP 6: Cleanup

List containers:
docker ps

Stop and remove:
docker stop <container_id>
docker rm <container_id>

Optional: delete image:
docker rmi my-static-site

Final Structure Recap

my-project/
■■■ Dockerfile
■■■ html/
  ■■■ index.html