

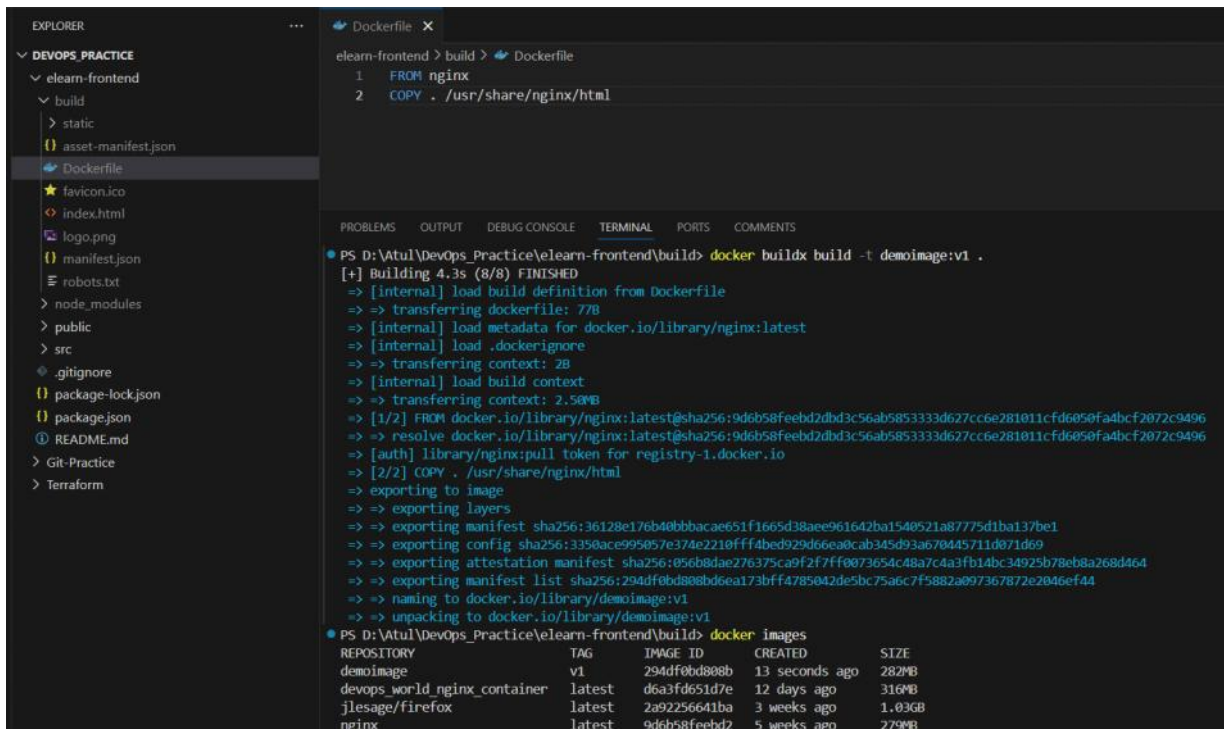
Deployment Of Frontend Application by making custom image using nginx as base image and running on Container

15 March 2025 23:50

Follow all the process mention in the README.md and run npm run build and this will create build folder.

Go inside the build folder and create Dockerfile

Using this command `docker buildx build -t demoimage:v1 .` This will create a custom image of application using nginx base image on Debian based operating system.



The screenshot shows a VS Code editor with a file explorer on the left and a terminal on the right. The file explorer shows a project structure with a 'build' folder containing 'static', 'asset-manifest.json', 'Dockerfile', 'favicon.ico', 'index.html', 'logo.png', 'manifest.json', 'robots.txt', 'node_modules', 'public', 'src', '.gitignore', 'package-lock.json', 'package.json', 'README.md', 'Git-Practice', and 'Terraform'. The 'Dockerfile' is selected and its content is shown in the editor: `FROM nginx` and `COPY . /usr/share/nginx/html`. The terminal shows the output of the `docker buildx build -t demoimage:v1 .` command, which successfully builds the image. Below the terminal output, a table lists the images available in the repository.

```
1 FROM nginx
2 COPY . /usr/share/nginx/html
```

```
PS D:\Atul\DevOps_Practice\eleam-frontend\build> docker buildx build -t demoimage:v1 .
[+] Building 4.3s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 77B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 2.50MB
=> [1/2] FROM docker.io/library/nginx:latest@sha256:9d6b58febd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
=> => resolve docker.io/library/nginx:latest@sha256:9d6b58febd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [2/2] COPY . /usr/share/nginx/html
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:36128e176b40bbacae651f1665d38aee961642ba1540521a87775d1ba137be1
=> => exporting config sha256:3350ace995057e374e2210fff4bed929d66ea0cab345d93a670445711d071d69
=> => exporting attestation manifest sha256:056b8dae276375ca9f2f7ff0073654c48a7c4a3fb14bc34925b78eb8a268d464
=> => exporting manifest list sha256:294df0bd808bd0ea173bff4785042de5bc75a6c7f5882a097367872e2046ef44
=> => naming to docker.io/library/demoimage:v1
=> => unpacking to docker.io/library/demoimage:v1

PS D:\Atul\DevOps_Practice\eleam-frontend\build> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
demoimage	v1	294df0bd808b	13 seconds ago	282MB
devops_world/nginx_container	latest	d6a3fd651d7e	12 days ago	316MB
jlesage/firefox	latest	2a92256641ba	3 weeks ago	1.03GB
nginx	latest	9d6b58febd2	5 weeks ago	279MB

Now it is ready to run this image on the Newly created container.

```
PS D:\Atul\DevOps_Practice\eleam-frontend\build> docker run --detach --name demoContainer -p 8000:80 demoimage:v1 d69113715625fcd34689354438b02b73e01e9ea1f8cd5f98519e81cc82a8d47
```

Error fetching courses

Made with ❤️ at DevOps Insiders