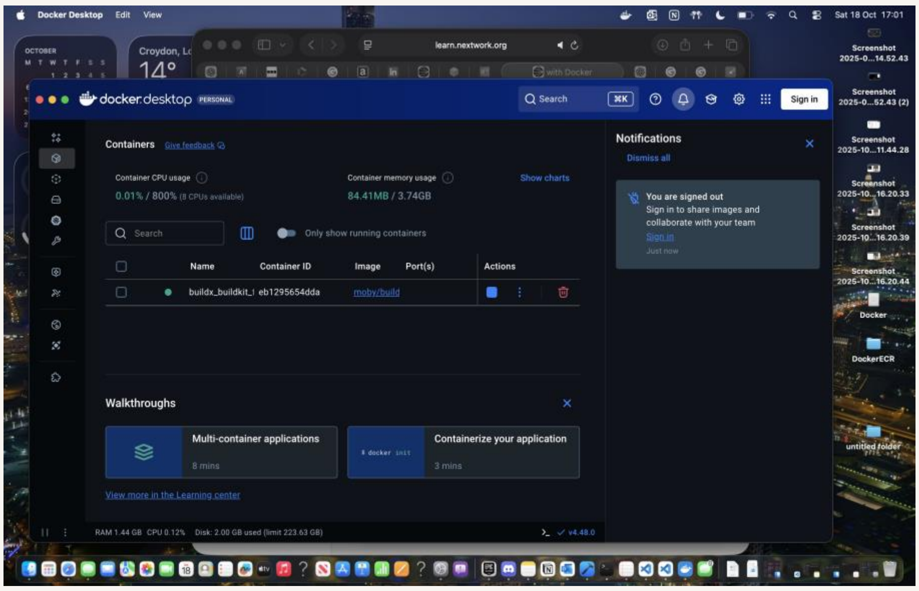
Containerised Web App Deployment with Docker

By Shuayb Mohamed



Container CPU usage 0.01% / 800% (8 CPUs available)

Container memory usage 84.41MB / 3.74GB

Name	Container ID	Image	Port(s)	Actions
buildx_buildkit_	eb1295654dda	moby/build		

Notifications

Dismiss all

You are signed out
Sign in to share images and collaborate with your team
Sign in
Just now

RAM 1.44 GB CPU 0.12% Disk: 2.00 GB used (limit 223.63 GB)

v4.48.0

Introducing Today's Project!

What is Docker?

Docker is a software that contains all of your app’s infrastructure so that when someone else runs it on their computer, it works exactly the same. It’s useful because it makes your app portable, meaning it can run anywhere without worrying about setup or system differences.

One thing I didn't expect...

One thing I didn’t expect in this project was how much knowledge I could grasp within Docker. I got a much deeper understanding of how containers actually work and how they isolate an app from the host system.

This project took me...

This project took me around 2.5 hours to complete. The most challenging part was creating and configuring the environment on AWS Elastic Beanstalk, as it required understanding how application versions and environments work together. Despite that, it was rewarding to see my web app successfully deployed online.

Understanding Containers and Docker

Containers

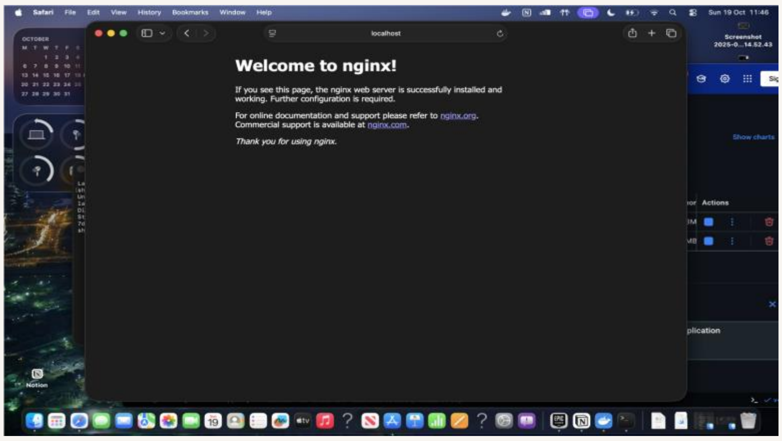
Docker containers package everything an app needs to run including its code, dependencies and settings so it works the same on your computer or anyone else’s. This prevents issues where your code works on one device but not another and helps keep development and deployment consistent.

Docker

Docker is a tool that allows you to package your application and everything it needs so it can run smoothly on another device. Docker Desktop makes it easy to manage and monitor your containers in one place. The Docker daemon is the background process that manages and runs Docker containers on your computer.

Running an Nginx Image

Nginx is a web server that can be used to run websites and web applications. To start a new container, I used the command docker run -d -p 80:80 nginx, which runs Nginx in the background and maps it to port 80 on my computer so it can be accessed through the browser.



Welcome to nginx!

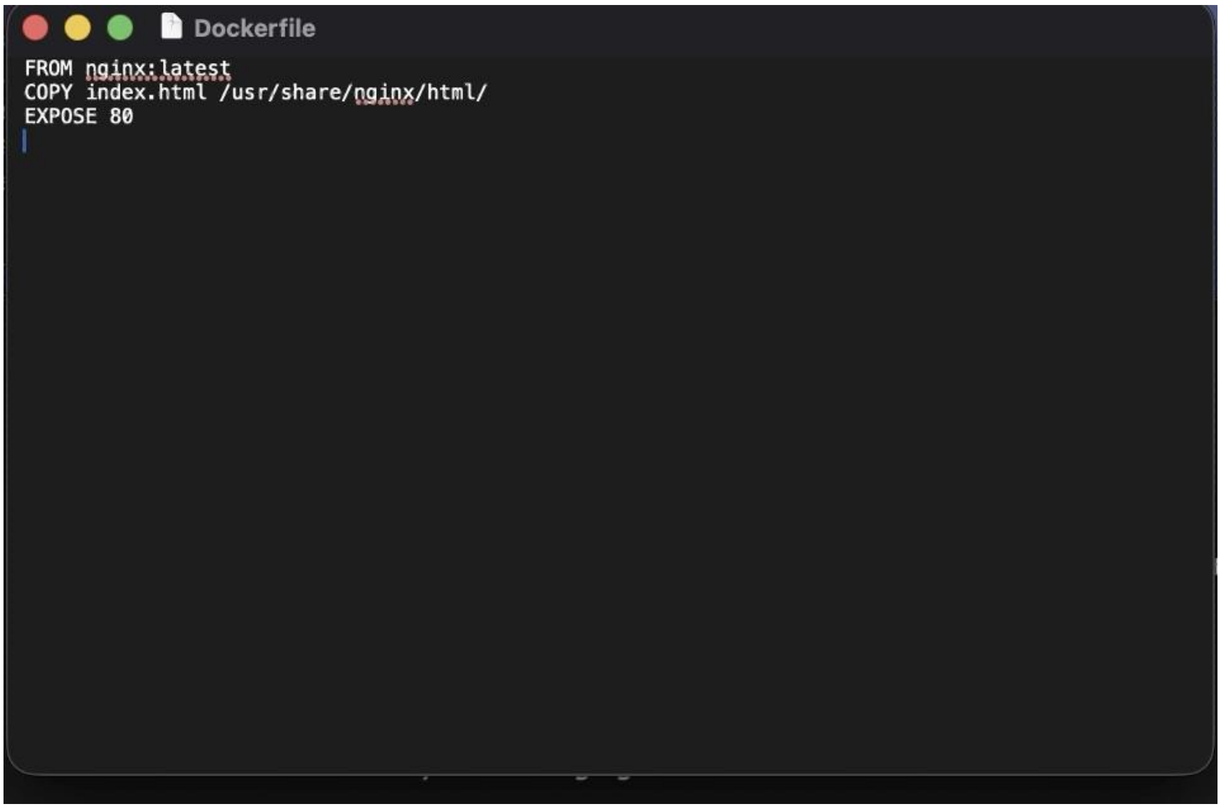
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Creating a Custom Image

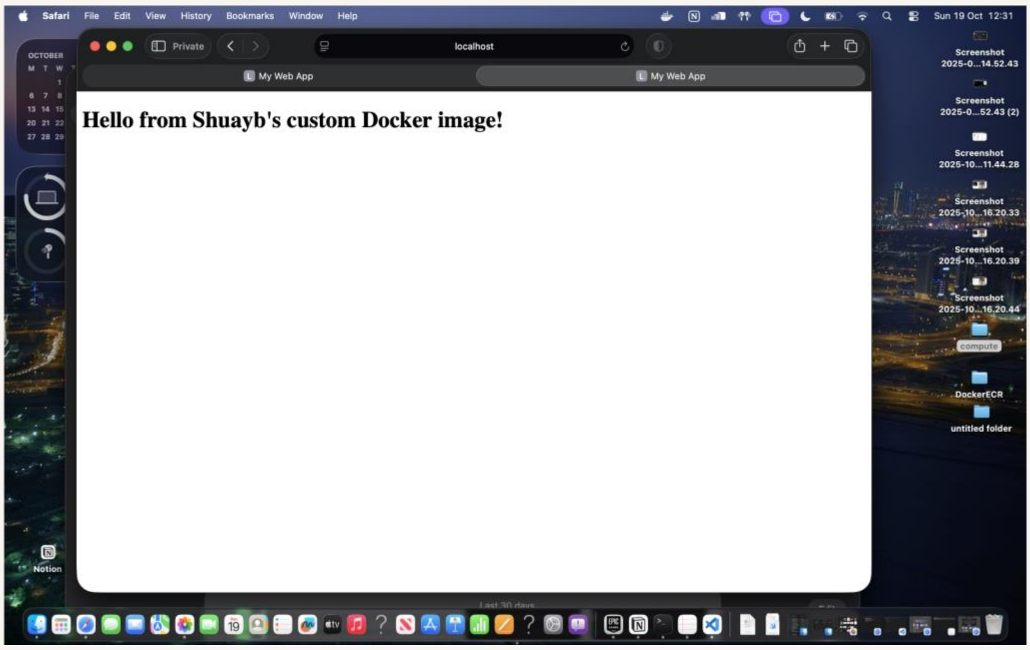
The Dockerfile is a file that contains all the instructions needed to build a Docker image. My Dockerfile tells Docker three things. In the first line, it defines the base image that my project starts from. The next two lines customise that base image for my web app. The command I used to build a custom image from my Dockerfile was “docker build -t my-web-app .”. The dot at the end tells Docker to look for the Dockerfile in the current directory.



```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```

Running My Custom Image

The error I ran into was caused by another container already using port 80, which meant my new custom container couldn't start on the same port. I resolved this by opening Docker Desktop and stopping the container that was using port 80. In this example, the container image acts as the template that defines the application's code and setup, while the container itself is the actual software instance created and run from that image.

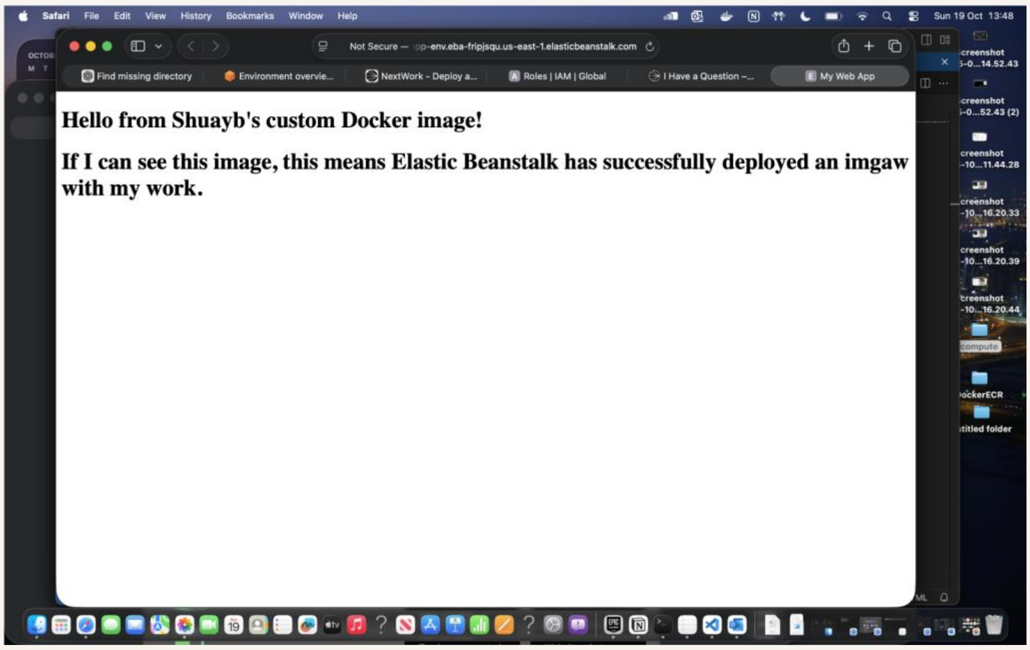


Hello from Shuayb's custom Docker image!

Elastic Beanstalk

Elastic Beanstalk is a service that makes it easy to deploy applications without having to worry about the underlying infrastructure. Deploying my custom Docker image with Elastic Beanstalk took me around two hours, as I learned how application versions and environments work together to host my web app online.

I accidentally wrote “imgaw” instead of “image,” which gave me a laugh later; but it still proved that my deployment worked perfectly.



Hello from Shuayb's custom Docker image!

If I can see this image, this means Elastic Beanstalk has successfully deployed an imgaw with my work.

The place to learn & showcase your skills

Check out nextwork.org for more projects