

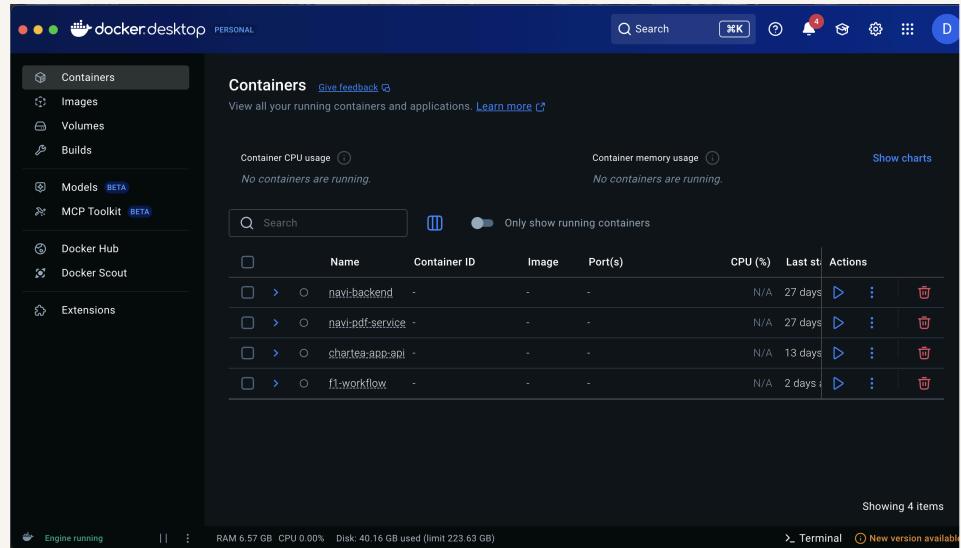


[nextwork.org](https://nextwork.org)

# Deploy an App with Docker



Diego Losada





**Diego Losada**

NextWork Student

[nextwork.org](http://nextwork.org)

# Introducing Today's Project!

## What is Docker?

Docker lets you package and run apps in consistent, portable containers. It's useful because it eliminates setup issues and ensures your app runs the same everywhere. I used it in this project to build, test, and prepare my app for cloud deployment

## One thing I didn't expect...

One thing I didn't expect in this project was that the zip files were stored in S3

## This project took me...

This project took me approximately an hour to complete. The most challenging part was understanding how each Docker instruction translated into the final container behavior. It was most rewarding to see my custom image running successfully



**Diego Losada**

NextWork Student

[nextwork.org](http://nextwork.org)

# Understanding Containers and Docker

## Containers

Containers are lightweight, portable environments that package an application together with everything it needs to run, its code, libraries, dependencies, and runtime. They ensure that an app behaves the same no matter where it's executed

A container image is a packaged blueprint of an application. It includes the code, dependencies, libraries, configuration, and everything needed to create a running container.

## Docker

Docker is a platform that lets you build, run, and manage containers. Docker Desktop is the application that makes Docker easy to use on your computer.

The Docker daemon is the background service that does all the heavy lifting for Docker. It's the engine that builds images, runs containers, manages networks, and handles all the low-level operations behind the scenes.



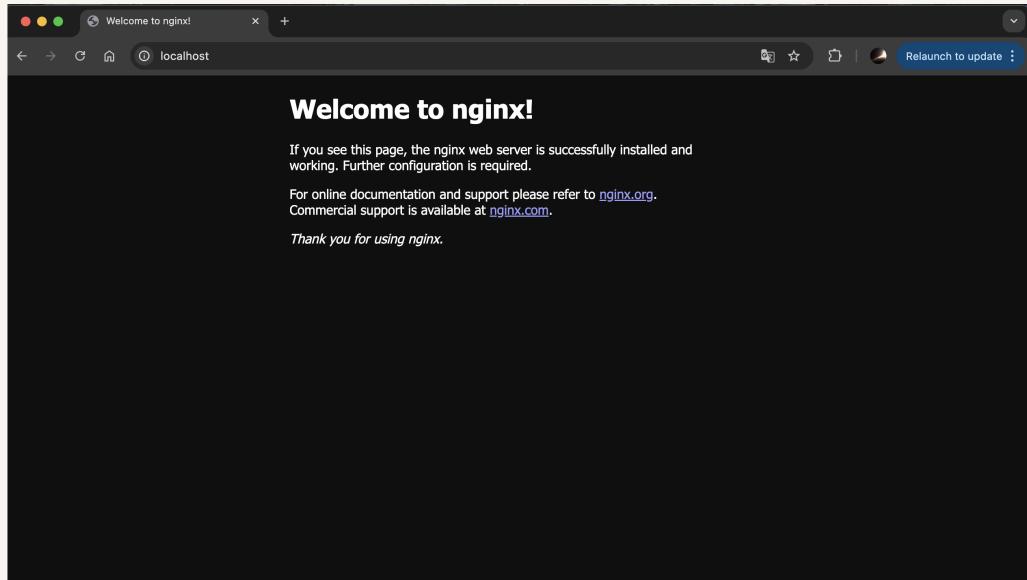
**Diego Losada**  
NextWork Student

[nextwork.org](http://nextwork.org)

# Running an Nginx Image

Nginx is a web server, which is a program you use to run websites and web apps.

The command I ran to start a new container was this: docker run -d -p 80:80 nginx





**Diego Losada**

NextWork Student

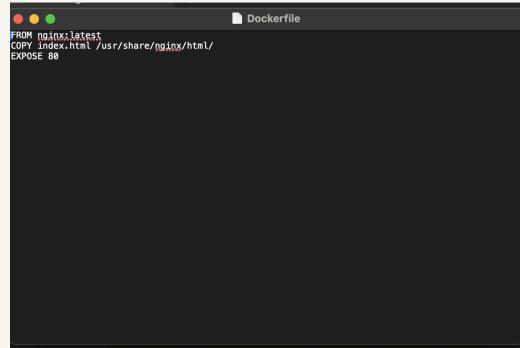
[nextwork.org](http://nextwork.org)

# Creating a Custom Image

The Dockerfile is a small text file that contains the exact instructions Docker should follow to build a container image. It acts like a recipe.

My Dockerfile tells Docker three things: 1- Start with an official Nginx image as the base. 2- Replace the default Nginx webpage with my own 3- Let Docker know that this container will serve content on port 80

The command I used to build a custom image with my Dockerfile was: docker build -t my-custom-image . The '.' at the end of the command means “build using the Dockerfile in the current directory.”



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



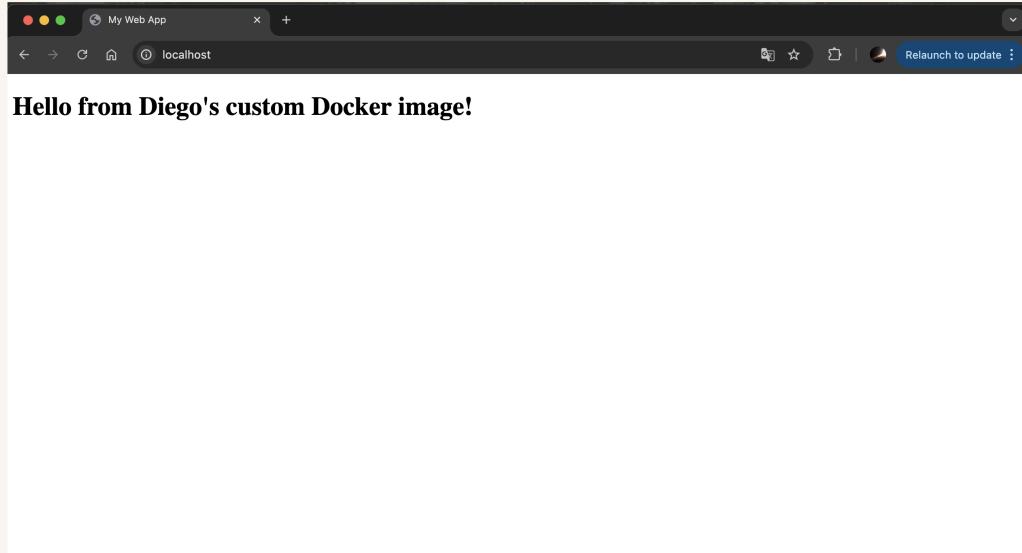
**Diego Losada**  
NextWork Student

[nextwork.org](http://nextwork.org)

## Running My Custom Image

There was an error when I ran my custom image because I didn't stop my old Nginx container. I resolved this by stopping and deleting that container, then running the new one again.

In this example, the container image is the blueprint, the packaged index.html and instructions that describe how the application should run. The container is the running instance created from that image.





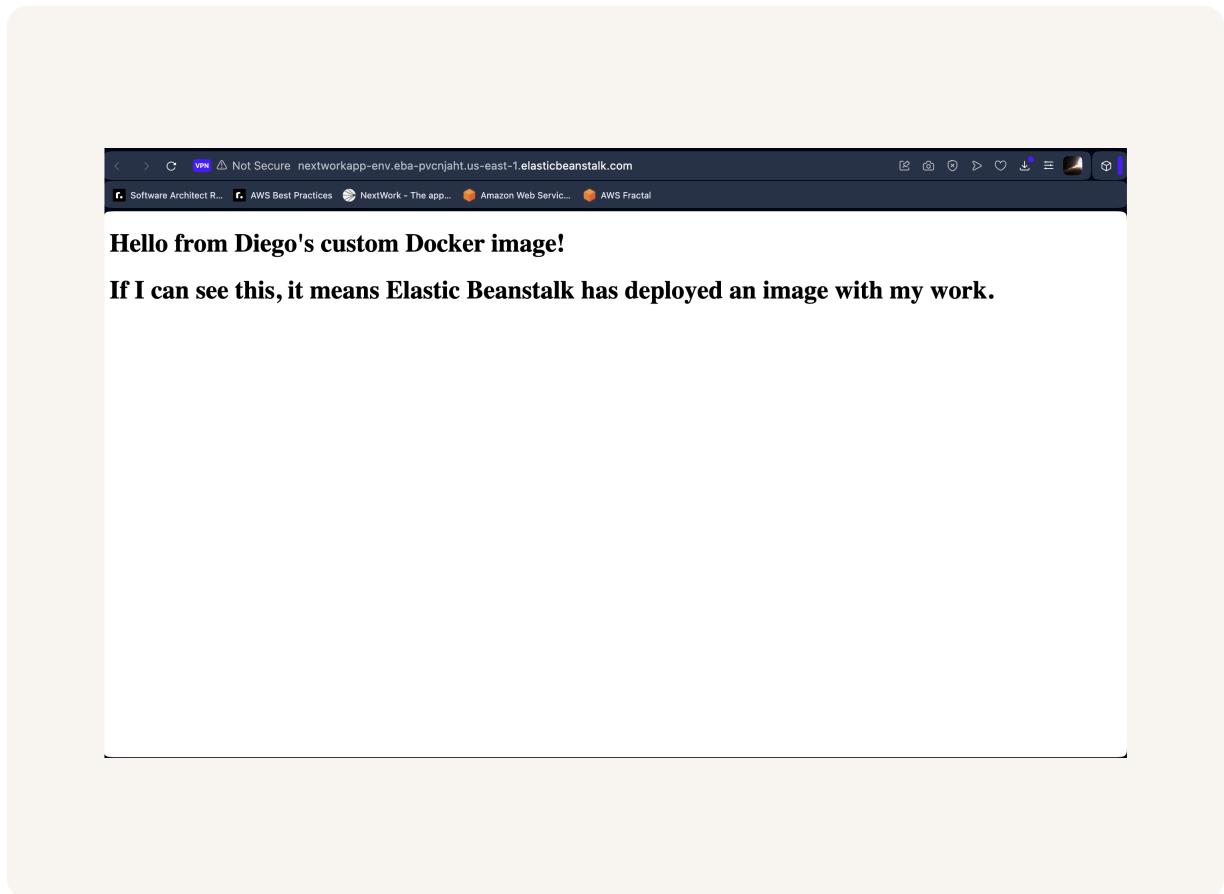
**Diego Losada**  
NextWork Student

[nextwork.org](http://nextwork.org)

# Elastic Beanstalk

Elastic Beanstalk is AWS's way of running your Docker containers without the setup headaches

Deploying my custom image with Elastic Beanstalk took me around 7 minutes





[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

