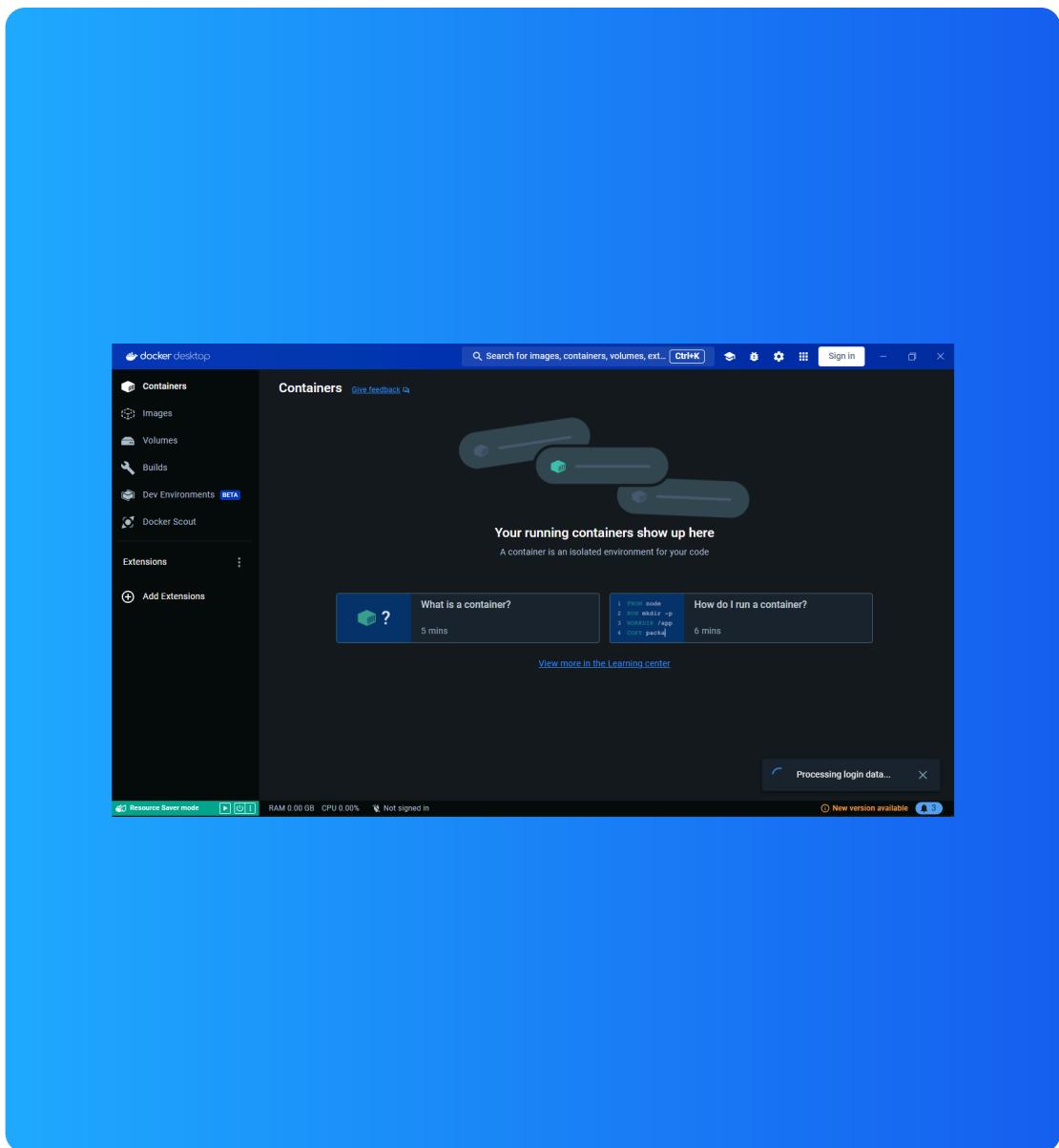




Containers on Elastic Beanstalk



David Arhin





David Arhin
NextWork Student

NextWork.org

Introducing Today's Project!

What is Docker?

I used Docker to create isolated containers for the application components, which helped streamline development and testing. By defining the application environment in a Dockerfile, I was able to easily replicate the setup on the server

One thing I didn't expect...

I didn't expect seeing how quick it was using and deploying on elastic beanstalk

This project took me...

2hours 30 minutes



Understanding Containers and Docker

Containers

Containers are lightweight, portable units that bundle applications with their dependencies, ensuring consistent performance across environments. They are useful because they enable efficient, scalable, and flexible application deployment.

A container image is a lightweight, standalone, and executable package that includes everything needed to run a piece of software, such as code, runtime, libraries, environment variables, and configuration files.

Docker

Docker is a platform for creating and managing containers. Docker makes working with containers easy. Docker Desktop is a software for using/interracting with docker. Docker desktop makes working with docker easy

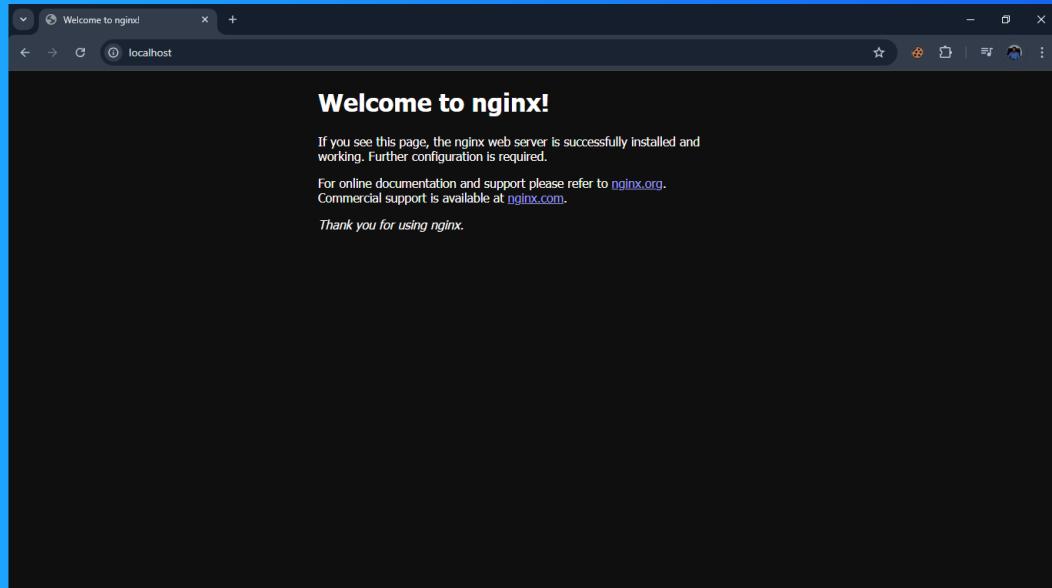
The Docker Daemon is a background service running on a host machine that manages Docker containers, images, networks, and storage.



Running an Nginx Image

Nginx is a high-performance web server and reverse proxy server used for serving web content, load balancing, and caching.

The command I ran to start a new container was `docker run <options> <image>`, `<options>` can specify settings like ports, environment variables, or detached mode (`-d`), `<image>` is the name of the container image to use. E.g: `docker run -d -p 80:80 nginx`



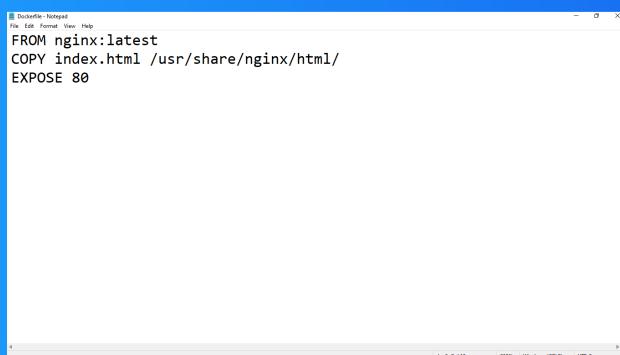


Creating a Custom Image

The Dockerfile is a set of instructions that tell docker how to build your custom container image

My Dockerfile tells Docker three things: First, my custom container uses the latest version of nginx container image Modifications are then made on this base by replacing the Nginx welcome page with my custom index.html It then exposes port 80

The command I used to build a custom image with my Dockerfile was 'docker build'. The '.' at the end of the command means that Docker can find the dockerfile in the current directory i.e Compute folder on the desktop



A screenshot of a text editor window titled "Dockerfile - Notepad". The content of the file is as follows:

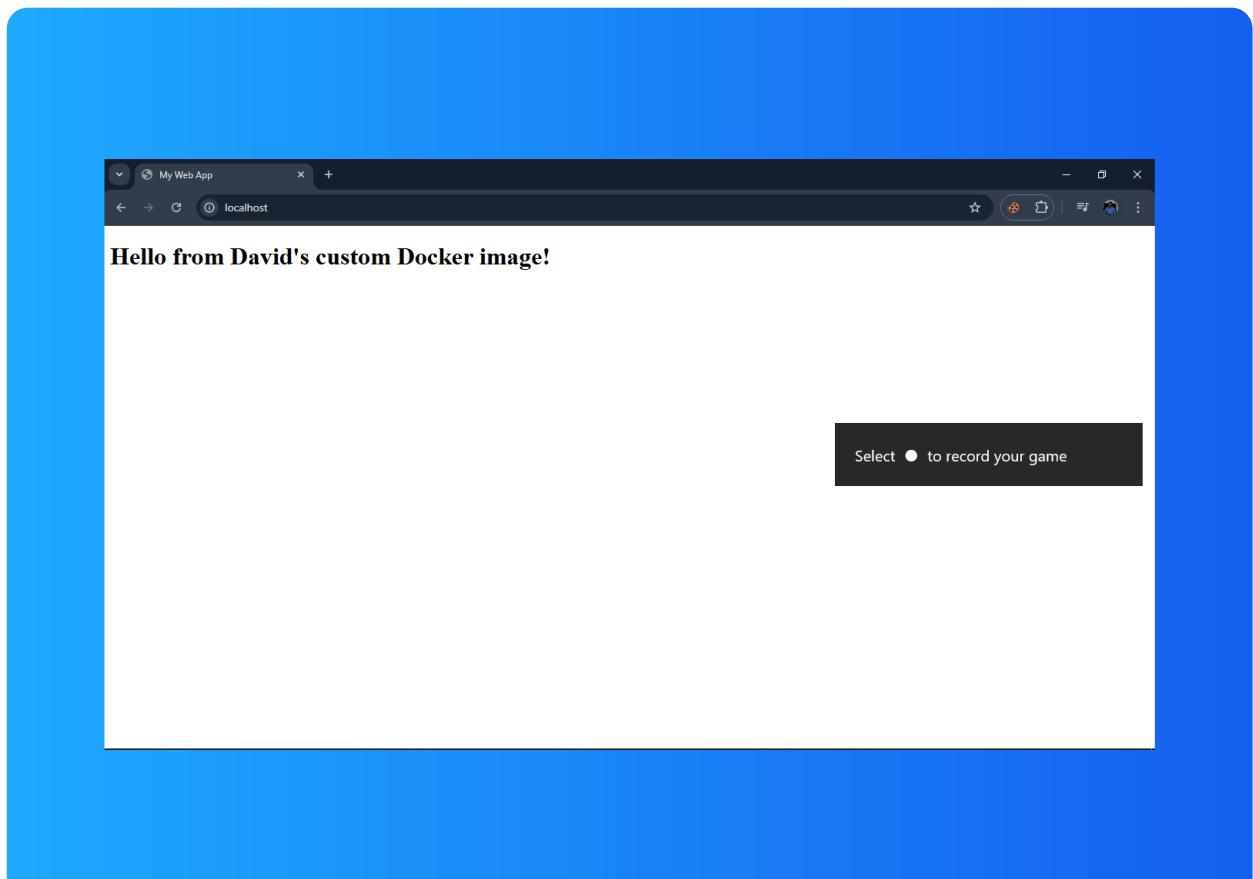
```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```



Running My Custom Image

There was an error when I ran my custom image because I tried to map my port 80 to the new container's port 80 but a running container was already using port 80. I resolved this by stopping the running container so the we can start the new one.

In this example, the container image is the template for creating a new container running an Nginx server that serves my custom index.html. The container is the actual software that's running an Nginx web server with those customization





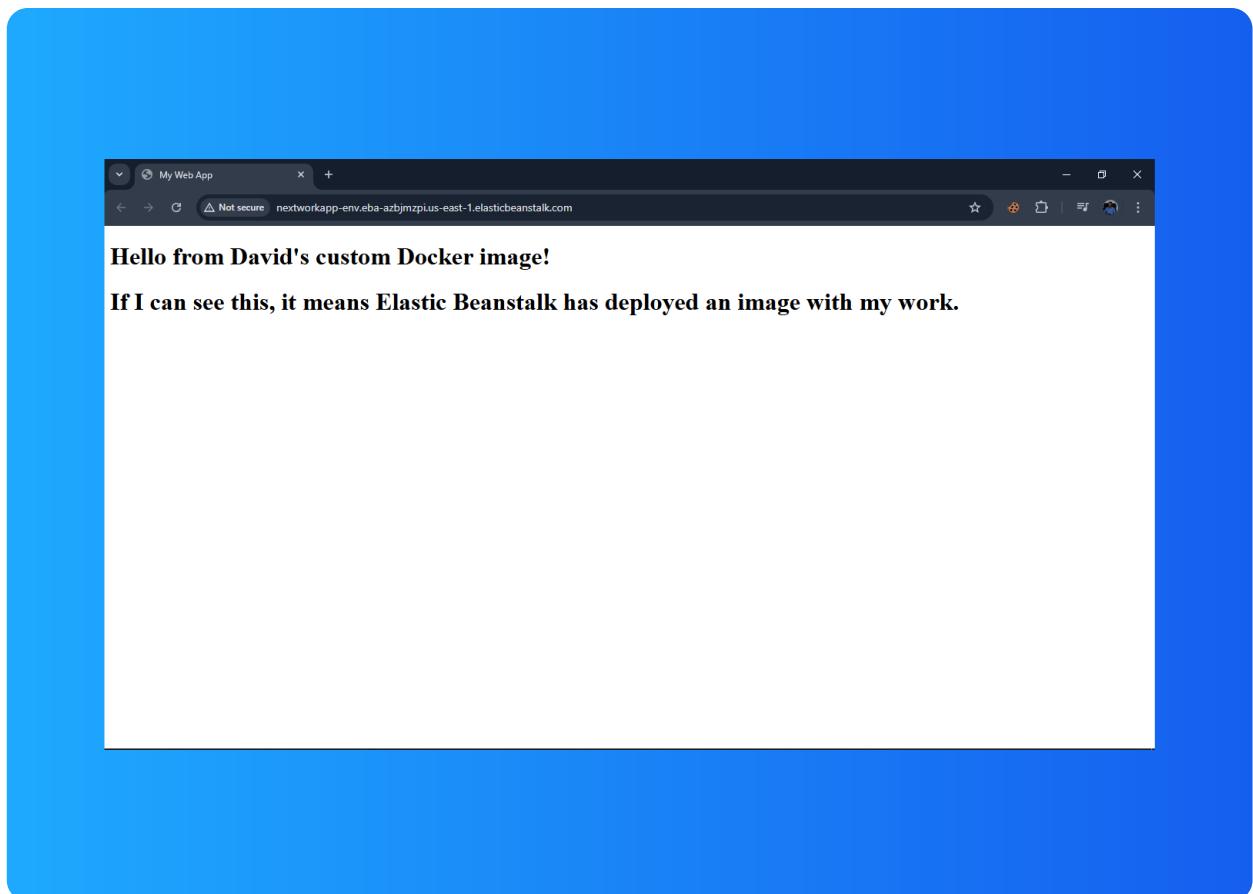
David Arhin
NextWork Student

NextWork.org

Elastic Beanstalk

Elastic Beanstalk is a Platform as a Service (PaaS) offered by AWS that simplifies the deployment and management of web applications and services, allowing developers to focus on writing code without worrying about the underlying infrastructure.

Deploying my custom image with Elastic Beanstalk took me 10 minutes. This includes the time it took to launch the Elastic Beanstalk application





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

