

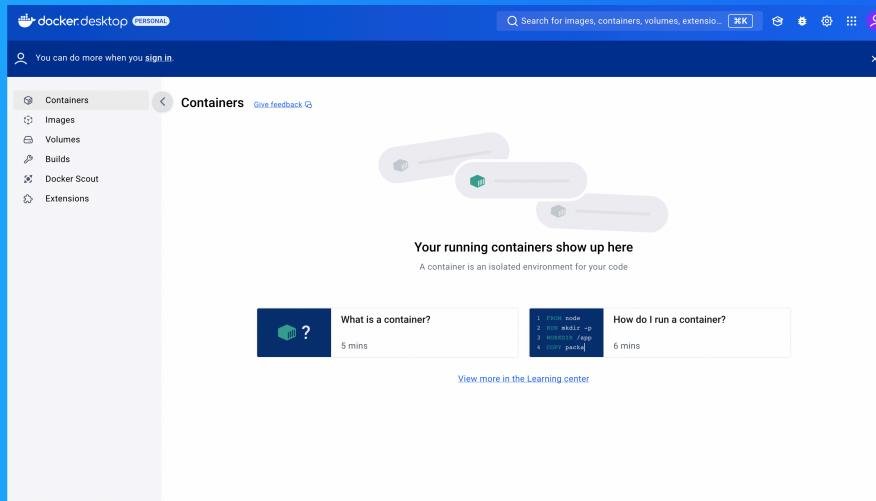


NextWork.org

# Containers on Elastic Beanstalk



sailavanyapudi08@gmail.com



# Introducing Today's Project!

## What is Docker?

Docker is an open-source platform that lets developers package applications and their dependencies into lightweight containers, ensuring consistent behavior across different environments.

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

How easy it is to package an application using docker.

## This project took me...

it took me an hour to learn about docker in depth.

# Understanding Containers and Docker

## Containers

Containers are like lightweight, portable boxes that package everything an application needs to run: the code, libraries, tools, and settings. It ensures that the application will run the same way, no matter where you use it.

A container image is a blueprint or template for containers. It gives Docker instructions on what to include in a container, such as application code, libraries, dependencies, and necessary files.

## Docker

Docker helps you create, manage, and run containers and docker desktop is a user-friendly application that provides a graphical interface (GUI) and the essential Docker tools for working with containers on a local computer.

Docker daemon is a background process that manages the Docker containers on your computer. It takes commands from the Docker client and does the heavy lifting of building, running, and distributing your containers

# Running an Nginx Image

Nginx is a web server, which means it's a program that serves web pages to people on the internet. Sometimes, you might hear Nginx referred to as a 'proxy server', which means it can also be used to forward requests from the internet.

```
docker run -d -p 80:80 nginx
```

## 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](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

# Creating a Custom Image

A Dockerfile is a document with all the instructions for building your Docker image. Docker would read a Dockerfile to understand how to set up your application's environment and which software packages it should install.

The first line, FROM nginx:latest, means our image starts as a copy of the latest Nginx image, but we'll make a few modifications/additions to it to customise it for what we need. Then, the next two lines in the Dockerfile customizes the base Nginx.

The command I used to build a custom image with my Dockerfile was docker build -t my-web-app . The '.' at the end of the command means it tells docker to find the Dockerfile in the current directory i.e. the Compute folder.

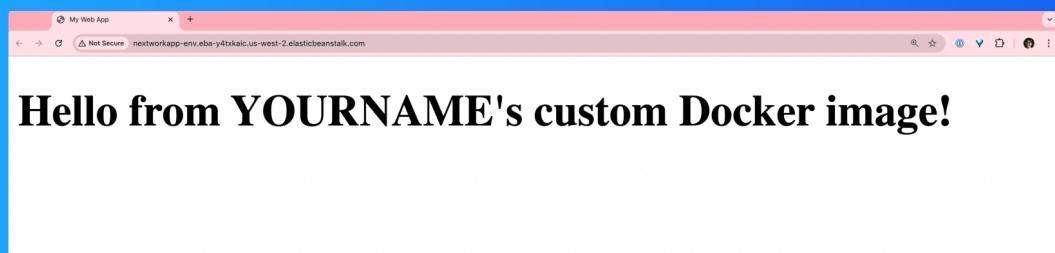


```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```

# Running My Custom Image

There was an error when I ran my custom image because there's already a container using port 80, so the new container that im creating can't access it. I resolved this by finding the container that's running in port 80 and stopping it.

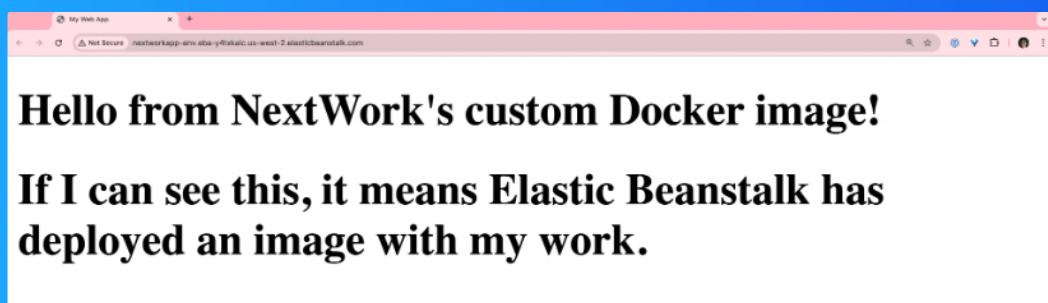
The container image is the blueprint that tells Docker the application code, dependencies, libraries etc that should go into a container. The container is the actual software that's created from this image and running on the web server.



# Elastic Beanstalk

AWS Elastic Beanstalk is a service that makes it easy to deploy cloud applications without worrying about the underlying infrastructure. You simply upload your code and Elastic Beanstalk handles everything needed to get it running.

Deploying my custom image with Elastic Beanstalk took me half an hour to configure.





NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

