

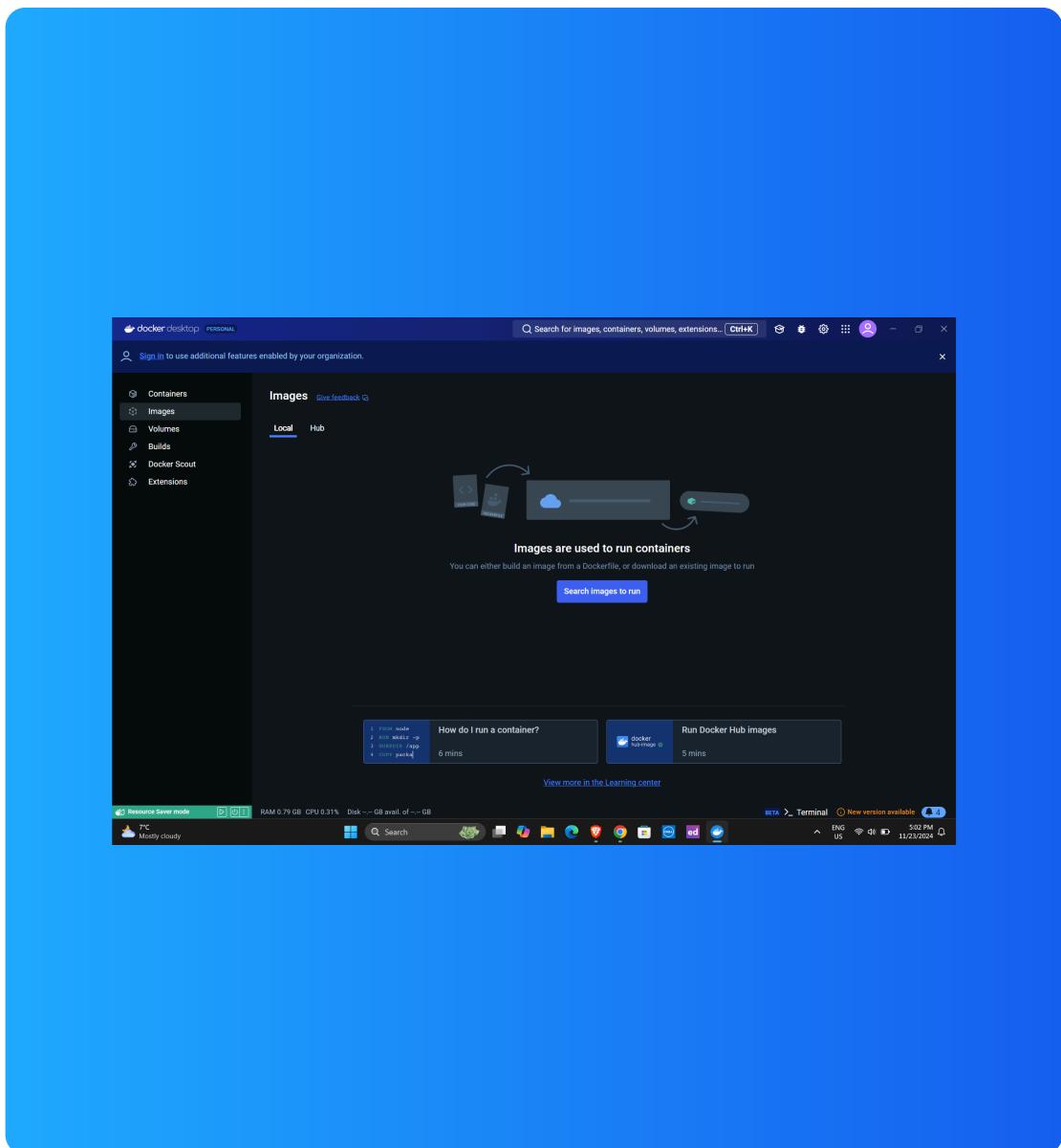


NextWork.org

# Containers on Elastic Beanstalk



Yash Rupani





**Yash Rupani**  
NextWork Student

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

# Introducing Today's Project!

## What is Docker?

In this project, we used Docker to create containers based on container images and set up our own container images.

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

One thing I didn't expect was how quick it was to deploy an application using Elastic Beanstalk

## This project took me...

This project took me around 2 hours.



# Understanding Containers and Docker

## Containers

Containers are tools for packaging applications in a way that's easy for developers to run. They are useful because it helps developer/ engineers working in a team together to share their work in a more efficient way.

A container image is a template/ blueprint for creating containers. Containers created from the same container images will behave in same way, which helps team of developers having a unified experience when they are running the same application.

## Docker

Docker is a platform for creating and managing containers. Docker makes working with container easy. Docker Desktop is a software for using/interact with Docker. Docker Desktop makes using Docker itself easy

The Docker daemon is like the 'engine' for Docker that receives command we send through clients e.g. clients click in the Docker Desktop or text command sent in the terminal and actually creates/manages/controls the containers.



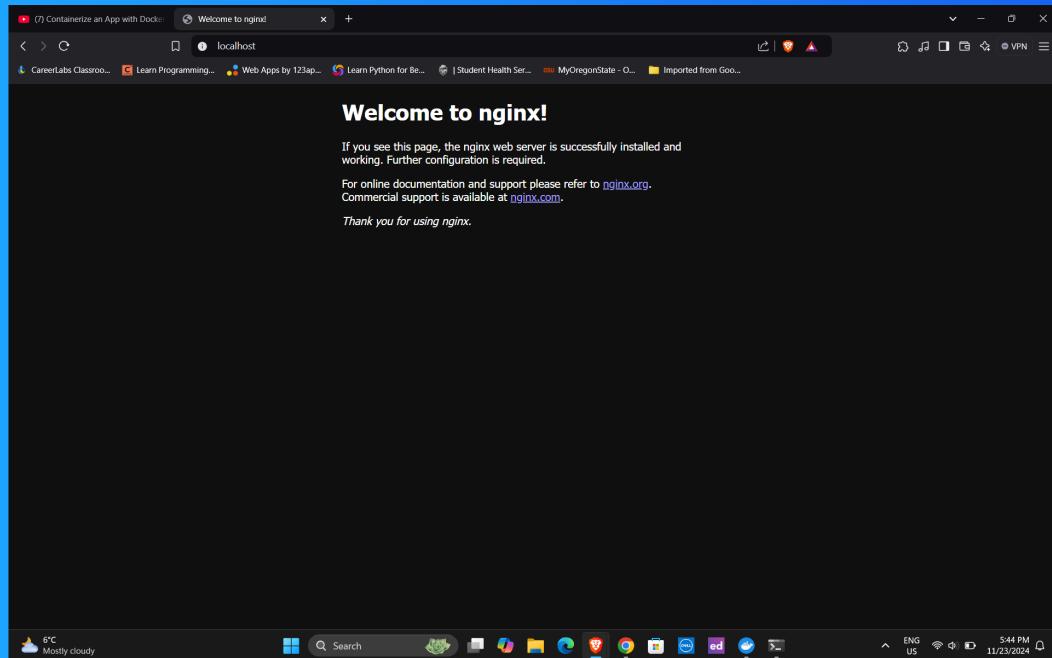
**Yash Rupani**  
NextWork Student

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

# Running an Nginx Image

Nginx is a web server/ a software that helps with serving web content. Nginx is often referred to as a proxy server, which it helps with distributing traffic to your application across the instances running your application.

The command I ran to start a new container was docker run. I also set the flags '-d -p 80:80 nginx', which means running container in background (-d) and we are matching port 80 in our host computer to the container's port 80 (-p 80:80)



**Yash Rupani**  
NextWork Student

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

# Creating a Custom Image

The Dockerfile is a set of instruction that tells Docker how to build your custom container image

My Dockerfile tells Docker three things. First, our custom container uses the latest version of Nginx container image at its base. Then, we are making modifying this base by replacing the default Nginx welcome with our own custom index.html

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

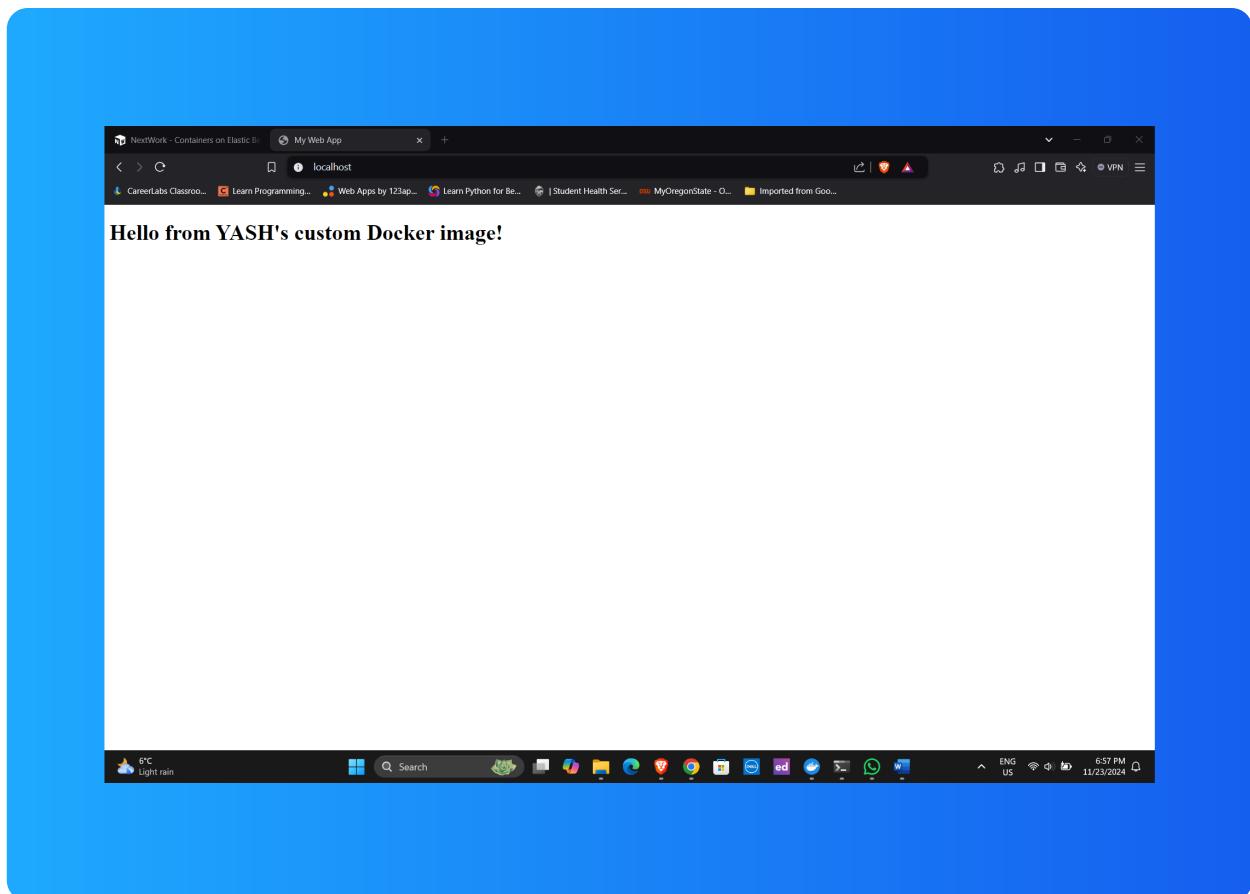
```
Dockerfile.txt
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```



# Running My Custom Image

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

In this example, the container image is the template for creating new container running on Nginx server that serves our index.html file. The container is the actual software that's running an Nginx web server with those customisation.





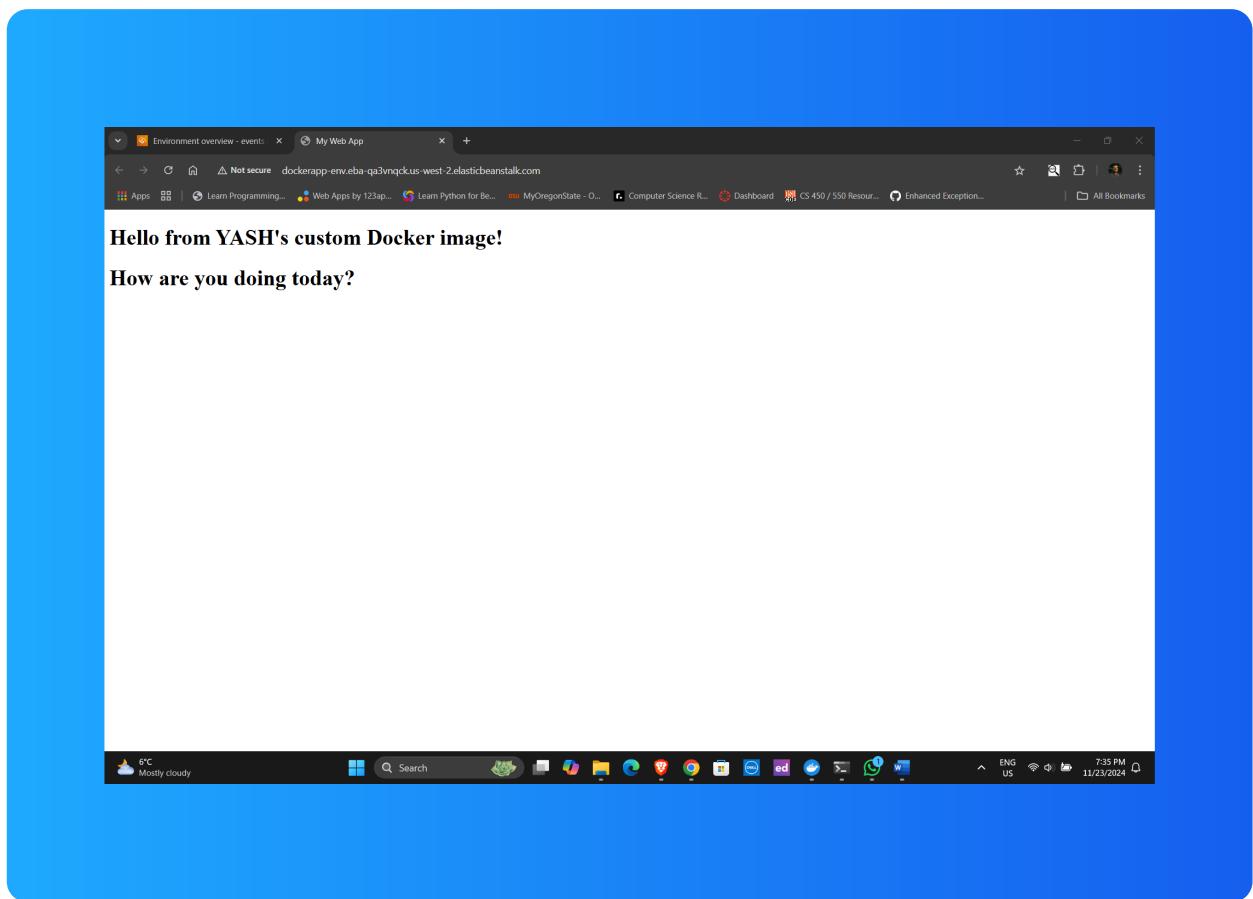
**Yash Rupani**  
NextWork Student

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

# Elastic Beanstalk

It is an AWS service that helps in deploying application to cloud. It abstracts away a lot of work with setting up cloud infrastructure when deploying application.

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





NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

