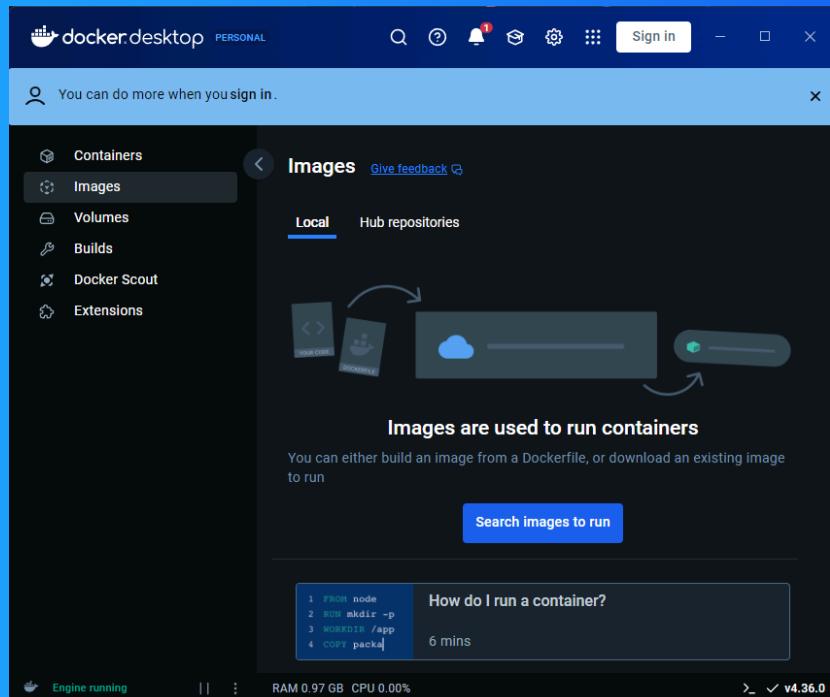




Containers on Elastic Beanstalk



Tony Winterz





Tony Winterz
NextWork Student

NextWork.org

Introducing Today's Project!

What is Docker?

Docker is an open-source platform that enables developers to build, ship, and run applications in a standardized unit called a container. We use docker to create our container and Elastic Beanstalk to deploy our containerized application into cloud.

One thing I didn't expect...

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

This project took me...

This Project took two and half hours, including all the demo time.



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 developers/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. Container spawn/created from the same image will behave in the same way, which helps teams of developers to have a unified experience when they're running the same application

Docker

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

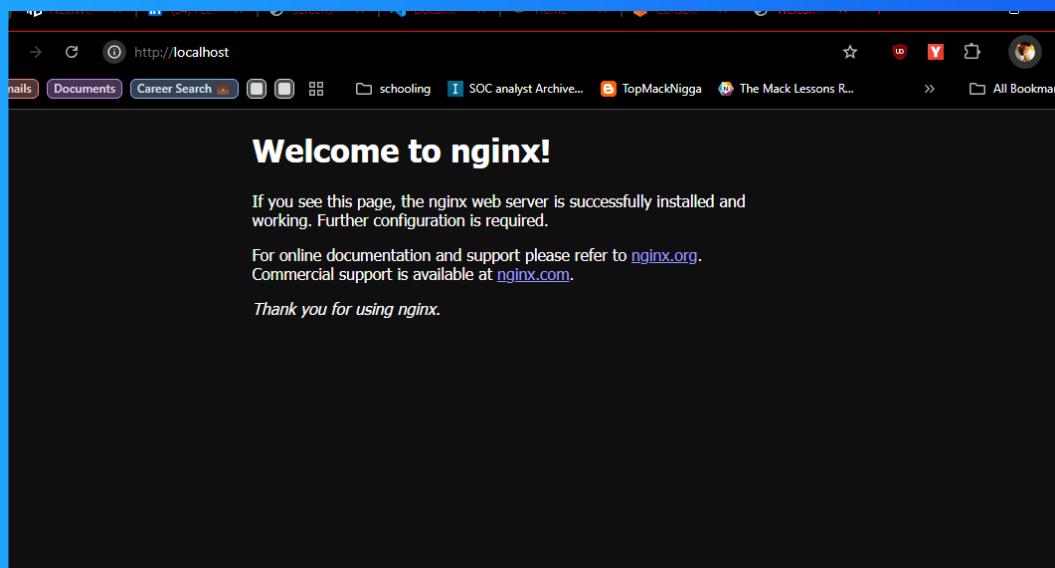
The Docker daemon is like the 'engine' for Docker that receives commands we send through clients e.g. clients in the Docker Desktop or text commands sent in terminal, and actually creates/manages/control the containers.



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 means 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. We also set the flags -d -p 80:80 nginx; which means we're running the container in the background(-d) and we're matching port 80 in our host computer to the container's port 80(-p 80:80)





Tony Winterz
NextWork Student

NextWork.org

Creating a Custom Image

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

Our Dockerfile tells Docker three things. First, our custom container image uses the (latest verison of the Nginx container image at its base. Then, we're modifiying this base by replacing the default Nginx welcome page 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 that Docker can find the Dockerfile in the current directory i.e. the computer folder on our desktop.

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



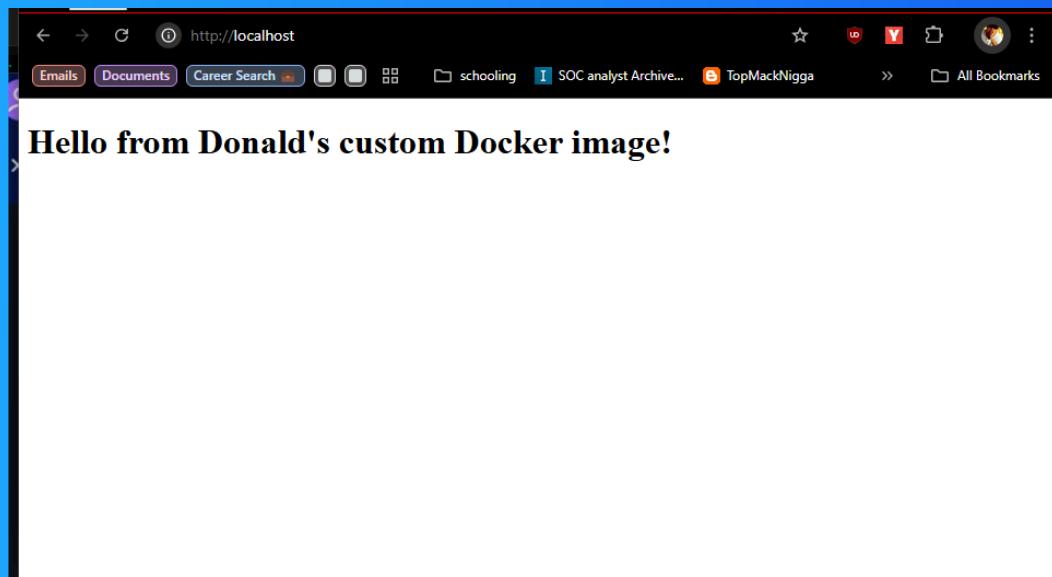
Tony Winterz
NextWork Student

NextWork.org

Running My Custom Image

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

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





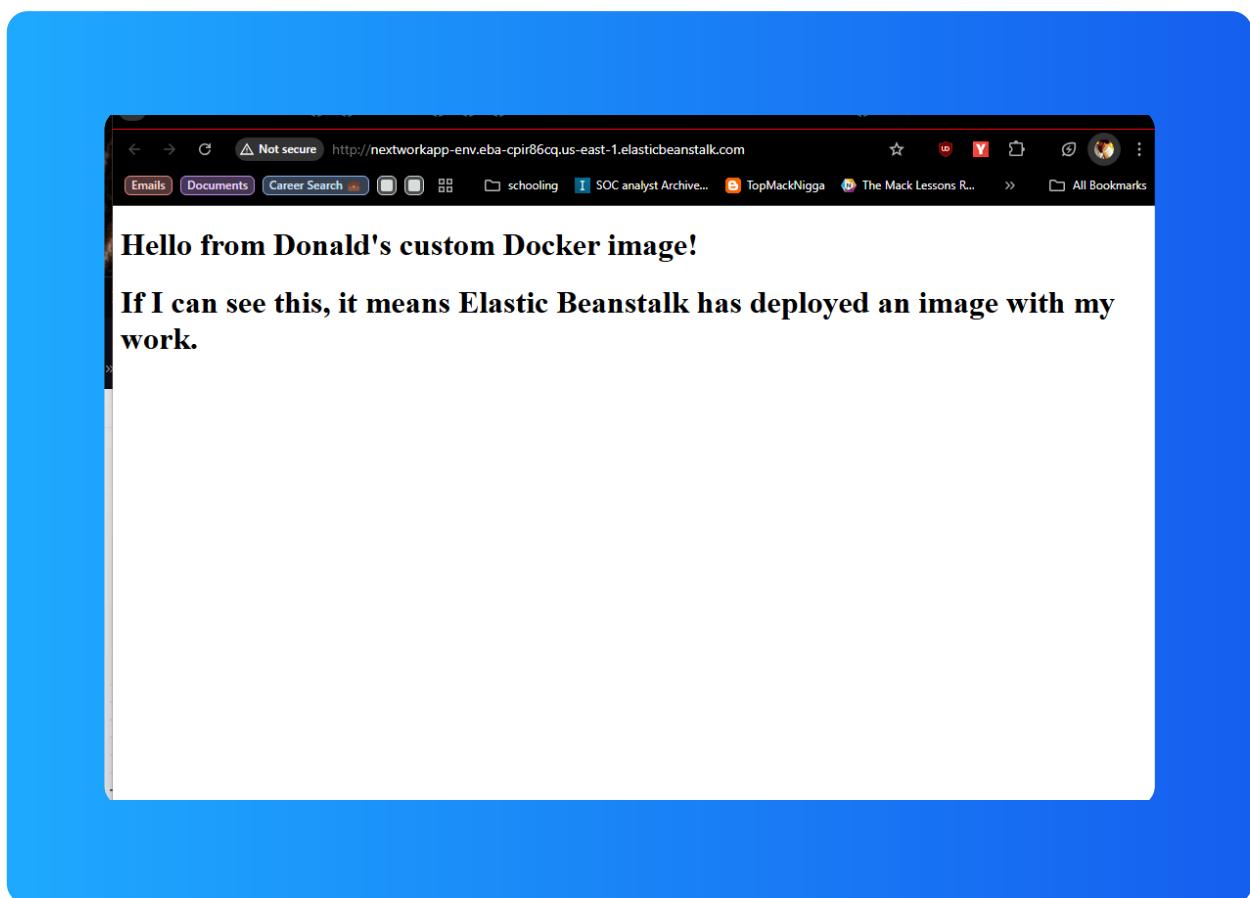
Tony Winterz
NextWork Student

NextWork.org

Elastic Beanstalk

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

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

