

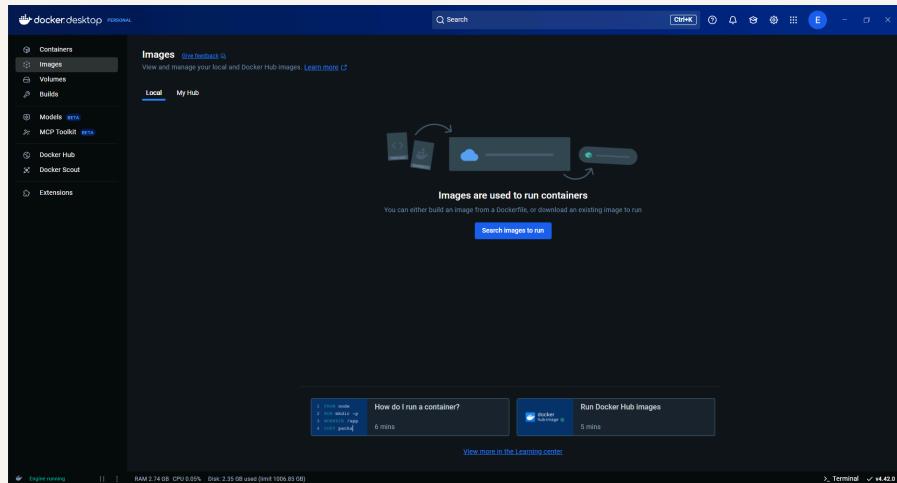


nextwork.org

Deploy an App with Docker



Damien Joseph





Damien Joseph
NextWork Student

nextwork.org

Introducing Today's Project!

What is Docker?

In this Project, we used Docker to create containers images and set up our own container image.

One thing I didn't expect...

One thing i didn't expect was seeing how quickly is to deploy an application using Elastic BeanStalk.

This project took me...

This project took me 3 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 template/blueprint for creating containers. Containers spawned/created from the same image will behave in the same way, which helps with developers in a team having a unified experience when they're running an 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.

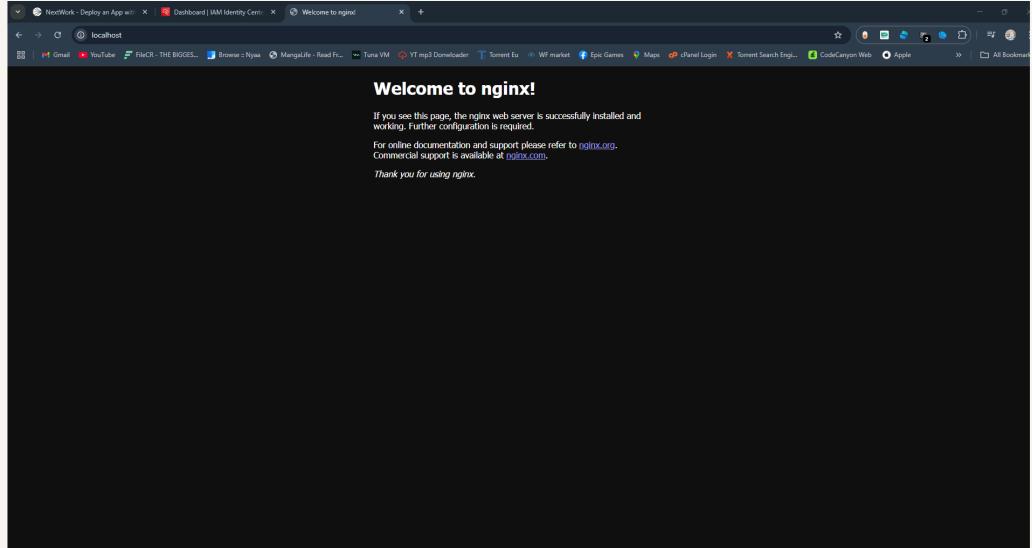
The Docker daemon is like the engine for Docker that receives commands we send through clients e.g. clients in the DockerDesktop or text commands sent in the terminal and actually creates/manages/controls the containers.



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. I also set the flags "-d -p 80:80 nginx", which means we are running the container in the background (-d) and we're matching port 80 in our host computer to the containers port 80 (-p 80:80).



Creating a Custom Image

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

My Dockerfile tells Docker three things. First, our custom container image uses the nginx container image as its base. Then, I made modifying this base by replacing the default Nginx welcome page with my 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 Compute folder in our desktop.

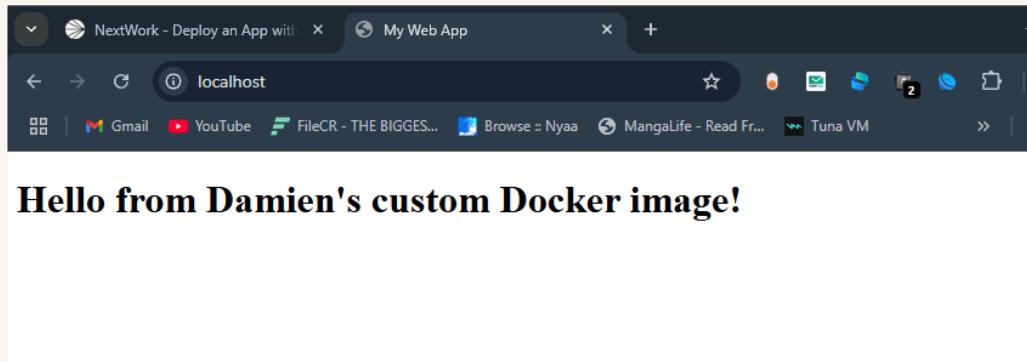
```
FROM nginx:latest
COPY index.html /usr/share/nginx/html/
EXPOSE 80
```



Running My Custom Image

There was an error when I ran my custom image because i tried to map my port 80 to the container's port 80, but a running container was already using it. 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 with those specifications.

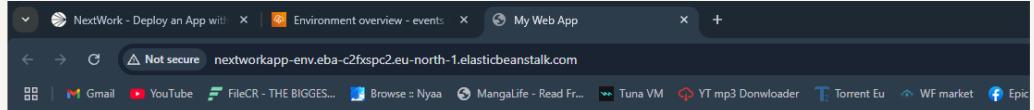




Elastic Beanstalk

Elastic Beanstalk is an AWS service that helps with deploying application in 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.



Hello from Damien's custom Docker image!

Hello from My New custom Docker image!



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

